

IMPORTANT NOTE

This document is supplied in its current, incomplete state to provide you with as much information as possible as early as possible. Although every attempt has been made to verify the accuracy of the information, this document may contain errors and is subject to change. Documents such as this one are sometimes released in an unfinished form to enable you to use and understand a product, prior to its release, as part of a test program.

Although the product referred to in this document has been announced, the information in this document is CONFIDENTIAL. Do not reproduce any part of this document and do not share the information it contains with others. The software described in this document is still independent and is still undergoing design chape.

Paula Metz Network and Communications Publications Preliminary Draft

Apple Confidential

May 26, 1989

APPLE COMPUTER, INC.

This manual and the software described in it are copyrighted, with all rights reserved. Under the copyright laws, this manual or the software may not be copied, in whole or part, without written consent of Apple, except in the normal use of the software or to make a backup copy of the software. The same proprietary and copyright notices must be affixed to any permitted copies as were affixed to the original. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased (with all backup copies) may be sold, given, or loaned to another person. Under the law, copying includes translating into another language or format.

You may use the software on any computer owned by you, but extra copies cannot be made for this purpose.

The Apple logo is a registered trademark of Apple Computer, Inc. Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

© Apple Computer, Inc., 1989 20525 Mariani Avenue Cupertino, CA 95014 (408) 996-1010

Apple, the Apple logo, AppleLink, AppleTalk, A/UX, LaserWriter, and Macintosh are registered trademarks of Apple Computer, Inc.

Comm Toolbox, Disk First Aid, MacTCP, MacX, and MultiFinder are trademarks of Apple Computer, Inc.

Ethernet is a registered trademark of Xerox Corporation.

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

Linotronic is a registered trademark of Linotype Company.

Microsoft is a registered trademarks of Microsoft Corporation.

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

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

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

VMS is a trademark of Digital Equipment Corporation.

Simultaneously published in the United States and Canada.

9/5/88

Contents

Symbols and conventions in this guide...iv Other publications and resources v 1 Malcolm Overview 1 What is the X Window System? 2 X Window System structure...3 X protocol 4 X server 5 Xlib 5 Window manager 5 X toolkit 6 A brief look at how X works...6 Windows and their hierarchy...7 Introducing Malcolm 8

What you get with Malcolm...8

Welcome to Malcolm

Who should read this guide ii

How to use this guide ii

For the beginner iii

For the expert...iii

Summary of contents...iii

Apple Confidential Contents

X with a Macintosh flair...9

What you see on the screen...9
Invisible root window 10
Visible root window 10
Using multiple screens...12

Overview of Malcolm features...14
Remote commands 14
Window management 15
Cut, copy, paste, and other editing features 17
Preferences 17
Fonts 19
Color Namer 20
Save, Kill, Quit, and other File menu commands 21

2 Getting Started 23

Opening Malcolm 24

Making a fast exit...25

Mouse and keyboard differences...25

Mouse differences 25

Keyboard differences 26

Substituting keys for mouse clicks...27

Starting a Terminal Emulator 28

Starting xterm...29

Handling errors...33

No xterm window opens 33

Other problems 34

Quitting Malcolm 34

Logging off versus killing clients...36

Closing a settings document without...36

3 Using Remote Commands

The Remote menu 38
Creating new commands 38
Saving passwords 43
Viewing, editing, and executing commands 45
The Output window 45

37

4 Handling Windows

47

The basics of using windows 48

The Window menu 48

Styling and manipulating windows...48

Window styles...49

Moving and sizing windows...51

Converting windows to icons...52

Killing clients and closing windows...53

Root window preferences 54

5 Managing Fonts and Color

57

Using fonts in Malcolm 58 The MacX Fonts folder...58 The Font Director 58 The X format for font names...59 Assigning aliases to font names...61 Copying and pasting font names...61 Removing fonts and font aliases...62 Sorting fonts...63 Compiling fonts from BDF files...64 Using Color in Malcolm 66 The Color Namer...67 Sorting, scrolling, saving, and cancelling...67 Adding new colors...68 Removing colors...69 Changing colors and color names...70 Copying and pasting color names...70

6 Troubleshooting 73

About errors 74

Problems opening Malcolm 75

Could not open the default font...75

Could not open the default cursor font...76

Unable to open MacX Temp...76

No connection tools are working at the moment...79

Error opening or reading a settings document...79

Auto Execute commands could not be started...79

Problems with remote commands 80

Host name not translated into a network address...80

Network communications error occurred...80

Internal error...81

Not enough memory...81

Problems with fonts 82

Error updating Font Directory file...82

A font property could not be loaded...84

Problems saving a settings file 84

Error occurred while saving document...84

7 Installation

87

What you need to run MacX 88 Hardware...88 Software...88 Communications...88 Your MacX package...89 Sample configurations 89 Installing the Communications Toolbox 91 Which tool should I install?...92 Installing ADSP 92 Installing MacTCP 92 LocalTalk or EtherTalk Cabling Method 94 Ethernet Cabling Method 97 RARP and BootP setting 98 RARP and RIP settings 98 RARP settings 99 BootP settings 99 RIP settings 100 Settings when no protocols apply 100 Entering configuration settings 101 Entering configuration settings for networks using RIP or none of the protocols 103 Installing MacX 106 Installing MacX on a hard disk...106 Preparing MacX to run on two floppy drives...107

Appendix A

Malcolm Versus Other X Servers

111

Font path searching 112
Typing international characters 112
Client incompatibilities 112
Reproducing uwm animation requests 113
Printing images of invisible root windows 113
Listing fonts...113

Appendix B

The Malcolm Window Manager 115

Transferring text between clients 116

Appendix C

Bitmap Distribution Format 2.1

117

Appendix D

X Logical Font Description Conventions

127

Appendix E

Character Conversion Charts 159

. .

Figures and tables

CHAPTER 1	MacX Overview 1		
	Figure 1-1	Some typical X windows 2	
	Figure 1-2	The client/server model 4	
	Figure 1-3	Malcolm on the desktop with the root window invisible 10	
	Figure 1-4	Malcolm on the desktop with the root window visible 11	
	Figure 1-5	A Macintosh with multiple screens attached 12	
	Figure 1-6	A control panel diagram of multiple screens 13	
	Figure 1-7	The Remote Command window 15	
	Figure 1-8	Malcolm Window styles 16	
	Figure 1-9	The Miscellaneous Preferences 18	
	Figure 1-10	The Font Director Window 19	
	Figure 1-11	The Color Namer Window 20	
	Table 1-1	Screen numbers 14	
CHAPTER 2	Getting Start	re d	
	Figure 2-1	Malcolm Desktop 24	
	Figure 2-2	Substitute mouse buttons 25	
	Figure 2-3	The Arrow keys preference option 26	
	Figure 2-4	The Remote Command window 29	
	Figure 2-5	The host name dialog box 31	
	Figure 2-6	Sample xterm window 32	
	Figure 2-7	Sample output window 34	

Apple Confidential Contents vii

	Figure 2-8	"The Save changes before closing?" dialog box 35	
	Figure 2-9	The "Save as" dialog box 35	
	Figure 2-10	The Kill Client dialog box 36	
	Table 2-1	Keyboard substitutes for clicking buttons 27	
CHAPTER 3	Using Remote	Commands	
	Figure 3-1	The Remote menu 38	
	Figure 3-2	The Remote Command window 39	
	Figure 3-3	The Output window 41	
	Figure 3-4	Selecting Miscellaneous Preferences 43	
	Figure 3-5	Miscellaneous Preferences window 44	
	Figure 3-6	An example of the Edit Command submenu 45	
	Figure 3-7	Top portion of the Output window 46	
	Figure 3-8	The Change Settings box 46	
CHAPTER 4	Handling Windows		
	Figure 4-1	The Window menu 48	
	Figure 4-2	Malcolm window styles 49	
	Figure 4-3	Location of the Windows preference 51	
	Figure 4-4	The Kill Client alert box 53	
	Figure 4-5	The Monochrome Rooted Screen Setup window 54	
•	Figure 4-6	Root window warning message 55	
CHAPTER 5	Handling Fon	ts and Color	
	Figure 5-1	The Font Director Window 59	
	Figure 5-2	Entering a font alias 61	
	Figure 5-3	Removing a font 62	
	Figure 5-4	Compiling an X font 64	
	Figure 5-5	The Color Namer window 67	
	Figure 5-6	Entering a new color name 68	
	Figure 5-7	The Color Wheel dialog box 69	
	Table 5-1	X logical font name format 60	
CHAPTER 6	Troubleshoot	ing	
	Figure 6-1	Sample internal error alert box 74	
	Figure 6-2	Sample alert box for an unexpected error 74	

	Figure 6-3	Sample warning message 75
	Figure 6-4	About the Finder display 77
	Figure 6-5	Get Info Display for Malcolm 78
CHAPTER 7	Installation	
	Figure 7-1	What you should have 89
	Figure 7-2	A small Local Area Network 90
	Figure 7-3	A small internet 90
	Figure 7-4	A large internet 91
	Figure 7-5	Contents of the connection tool disk 93
	Figure 7-6	The MacTCP Control Panel 94
	Figure 7-7	The MacTCP Configuration Panel 96, 102, 103
	Figure 7-8	Entering the node range endpoints 105
	Figure 7-9	Entering the IP address 105
	Figure 7-10	Contents of the MacX Program disk 107
	Figure 7-11	Contents of the MacX Program disk 108
	Figure 7-12	Contents of the MacX Disk 108
	Figure 7-13	Sample System Folder 109

Apple Confidential

Welcome to Malcolm

HE MALCOLM USER'S GUIDE DESCRIBES HOW TO INSTALL AND USE MALCOLM, 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.

Apple Confidential About This Guide

Who should read this guide

Anyone who plans to install and use Malcolm should consult this guide. Malcolm 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 Malcolm 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, knowlege of X, and profession, this user's guide tries to accommodate its readers' divergent needs. The next section, "How to use this guide," suggests to approaches to reading this manual, based on your technical knowledge and experience with X.

Although this guide teaches you how to install and use Malcolm, 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 VMSTM, 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 and other operating systems.

How to use this guide

Your experience with X, computers, and networks will determine the way you should use the Malcolm User's Guide. Choose one of the two suggested approaches that best suits you, depending on your level of experience in these areas.

For the beginner

If you haven't installed Malcolm, go to the Malcolm Installation Booklet (currently Chapter 7) 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 Malcolm differs from traditional implementations of X. It also contains a summary of Malcolm features, which you can read or scan, at your option. 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, below, for synopses of the remaining chapters and appendices.

For the expert

People with technical backgrounds or extensive X experience can bypass much of the introductory material. Thus, after installing Malcolm, read "Introducing Malcolm" in Chapter 1 to learn how Malcolm differs from standard X implementations. Chapter 2 is tutorial. Except for the "Opening Malcolm" and "Quitting Malcolm" sections, 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 appendices, where specific features are documented in detail.

Summary of contents

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

- Chapter 1, "Malcolm Overview," briefly describes the X Window System and then introduces Malcolm. It defines the major components of Malcolm and explains how Malcolm works with a visible or invisible root window and with multiple screens. Finally, it presents an overview of Malcolm features, including illustrations of menus and the dialog boxes that appear.
- Chapter 2, "Getting Started," covers the basic things you need to know to function in an X environment using Malcolm. It describes how to open Malcolm, log on to a host and start a terminal emulator, log off the host and exit from Malcolm.

- Chapter 3, "Using Remote Commands," describes how to create, execute, and edit remote commands using the Remote menu and the Remote Command window.
- 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 and set preferences for the root window.
- 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.
- Chapter 7, "Installation," states system requirements and illustrates sample network configurations for Malcolm. It describes what comes in the Malcolm product package, how to install Malcolm and two communications protocols: MacTCP and ADSP. (This chapter could become a separate document. Final format of installation still to be determined.)
- Appendix A, "Malcolm Versus Other X Servers," states the differences between Malcolm and conventional X servers.
- Appendix B, "The Malcolm Window Manager," identifies notable features of the Malcolm 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 matrices of ISO Latin 1 characters and Macintosh Extended ASCII characters and describes how Malcolm converts characters from one code set to the other.

Symbols and conventions in this guide

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

By the way: Text in this format are notes that present 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. \triangle

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.

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

An entire suite of 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 user's guide, available from Apple, is called the X Window System User's Guide for A/UX. For a complete description of A/UX manuals as well as part numbers and ordering information, see the Road Map to A/UX Documentation.

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.

Apple Confidential About This Guide

Malcolm Overview

N EARLY 1984, TWO GROUPS AT MIT EMBARKED ON PROJECTS THAT REQUIRED people to work with computers in an entirely different way. In one case, 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 two 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 Malcolm, Apple Computer's version of the X server, the specialized software that enables a Macintosh® computer to run X applications.

Apple Confidential

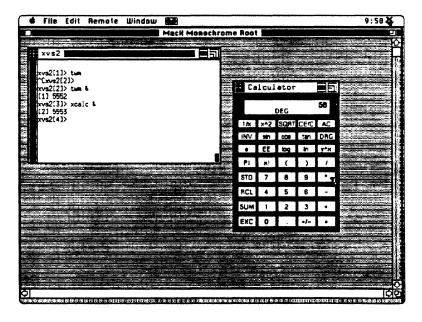
Chapter 1: Malcolm Overview

1

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 the applications you invoke in windows on your screen. Essentially, this process converts your screen into a lot of miniature screens on 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**.

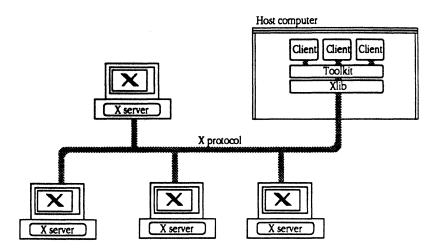
Although you can run X applications on your own microcomputer under an operating system that supports them (such as A/UX®), the primary reason for using X is that it lets you access programs remotely, over a network, and ensures that they function smoothly and reliably, as if the network weren't there. 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 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 has a two-part structure 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 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.

X protocol

X Window System protocol is a low-level graphics description language used by clients and the X server in your desktop computer (see next paragraph) 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 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—that each perform a certain task when called by a client program. For example, they 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, by convention, 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, 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 on 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. 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, by convention, 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 is sensitive to a mouse pointer, clients often draw tiny subwindows that look like menus, buttons, scroll bars, and other shapes. When a mouse pointer enters or is clicked in one of these subwindows, the client receives the feedback.

Another important point about the window hierarchy is the restrictions that higher-level windows impose 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 Malcolm, Apple's version of X for a Macintosh running the Macintosh operating system.

Introducing Malcolm

Malcolm is an X server for the Macintosh. By permitting a Macintosh using the Macintosh operating system to run X applications, Malcolm 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, Malcolm 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, Malcolm fully conforms to the standards published in the Inter-Client Communications Conventions Manual (ICCCM) and to the latest version of X: version 11, release 3.

What you get with Malcolm

In addition to the X server, Malcolm software includes:

■ Malcolm Window Manager

The Malcolm Window Manager allows you to use X windows like 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** mode, a mode in which the root window is invisible. Malcolm provides the option to display or hide the root window.

■ Malcolm 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.

■ Malcolm 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.

■ Malcolm Colors file

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

X with a Macintosh flair



Malcolm application icon



Malcolm file icon

What you see on the screen

Malcolm is designed to blend in with your Macintosh working environment, adding another dimension to your desktop without impeding or limiting Macintosh facilities. While Malcolm is running you can use MultiFinder™ and your desk accessories. You can run Macintosh applications and various X clients simultaneously. Functions such as cut, paste, save, and quit operate the same way, and you can cut and paste text between Macintosh and X applications. Malcolm uses the Macintosh method for handling windows plus some added features, but gives you the option to use a different window manager, too. Like standard X products, Malcolm 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 Malcolm 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.

Each time you start up Malcolm, it creates a settings document in which you can store commands and preferences. If you name and save this file when you quit Malcolm, a Malcolm file 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 on the file icon. Through these individual settings documents, Malcolm enables you to tailor X environments to different hosts, clients, types of tasks, or working styles.

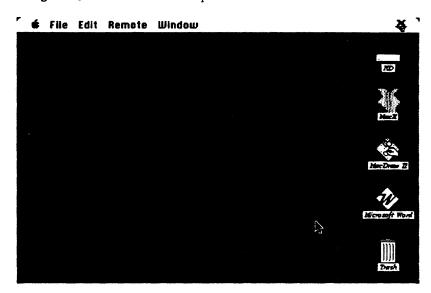
Malcolm offers an alternative to the standard X root window display. Although the option for displaying the root window exists, Malcolm is preset to hide the root window and display the Macintosh desktop instead. Figures 1-3 and 1-4 illustrate both cases.

For instructions on changing the root window setup, see Chapter 4, "Setting Preferences."

Invisible root window

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

■ Figure 1-3 Malcolm on the desktop with the root window invisible

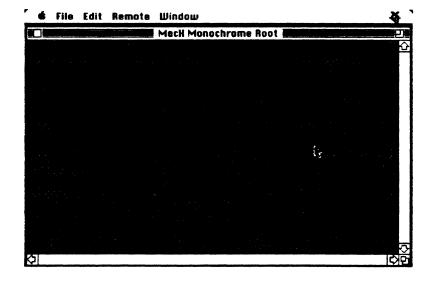


Visible root window

Malcolm is preset to create a visible root window if one of your clients requests one. Alternatively, you can set a preference to *always* or *never* display the root window and save this preference in a settings document.

(See Chapter 4 for more information.) Figure 1-4 shows what your desktop looks like, assuming that your Macintosh has just one screen attached. (Read the next section, "Using Multiple Screens," if you have more than one screen.) Usually the root window covers the entire screen unless someone has set a preference to make the root window bigger or smaller. You'll notice that the window in Figure 1-4 has a size box in the lower right corner. Although labeled "root," this window is actually a Macintosh window that reveals all or part of a logical root window stored in memory. In standard X environments, the root window occupies the entire screen and is frozen in that position. It cannot be moved to another screen or resized. However, the Macintosh was designed to permit you to move and modify windows. Unfortunately, the consequences of moving or resizing a root window that contains client windows are unknown because X was not designed to deal with this case. Malcolm avoids this uncertainty by creating the actual root window in memory and allowing you to view it through a Macintosh window. To see how this approach works, try decreasing the size of the Macintosh rootviewing window. Even though you can't see the whole root window now, you can use the scroll bars to move around and view all of it. You can also get to things on your desktop more easily.

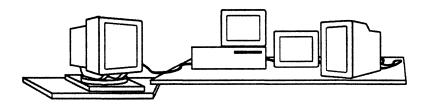




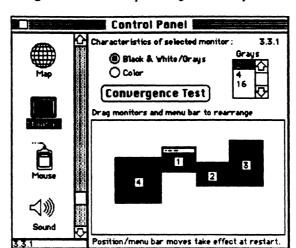
Using multiple screens

In accordance with X specifications, Malcolm supports single or multiple screens. The screens can be color, gray scale, or monochrome in any combination. Figure 1-5 illustrates a multiple screen setup. Malcolm distinguishes the types of screens attached to your Macintosh by checking the System File.

■ Figure 1-5 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 half way 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, select the Control Panel from the Apple menu (4) and click the Monitors icon. You will see a diagram similar to the one in Figure 1-6. The screens in the Control Panel should be arranged like the monitors on your desk, except with no spaces in between them.



■ Figure 1-6 A control panel diagram of multiple screens

Now draw an imaginary rectangle around all of the screens in your Control Panel, like the red one in Figure 1-6. You will notice some blank areas, if the screens are different sizes. Malcolm 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 2 in Figure 1-6 straight across to the blank area above screen 1, which is shorter, part of the window would not appear until you pull it down into screen 1. The window was always there, but Malcolm just couldn't re-create it in thin air.

In Malcolm, the rooted and rootless modes coexist unless you suspend rooted mode by setting the root window option to *never*. When issuing a command to display a client, you can specify whether you want it to appear in rooted or rootless mode on either a color or monochrome screen by adding a *screen number* to your command. (See "Starting a Terminal Emulator" in Chapter 2 for a complete explanation of issuing remote commands and specifying screen numbers.) Table 1-1 supplies a list of screen numbers.

Apple Confidential Chapter 1: Malcolm Overview 13

■ Table 1-1 Screen numbers

Screen number	Description	
0	Monochrome screen, rootless mode	
1	Monochrome screen, rooted mode (visible root window)	
2	Color screen, rootless mode (invisible root window)	
3	Color screen, rooted mode	

If your Macintosh has more than one screen of the same type attached to it, Malcolm will display 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 Malcolm will display the client on the main screen, that is, the one displaying the menu bar.

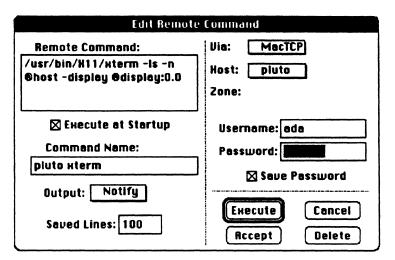
Overview of Malcolm features

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

Remote commands

Since Malcolm operates under the Macintosh operating system, instead of the UNIX or VMSTM operating systems, X applications cannot run directly on the Macintosh. In order to display clients on your screen, you must issue a remote command to a host on which the clients you want to use reside. Figure 1-7 shows the Remote Command window in which you type your command, username, password, and other options. Selecting New Command from the Remote menu, shown in the margin, brings up the Remote Command window.

■ Figure 1-7 The Remote Command window



After entering a remote command, you have the option to execute, accept, cancel, or discard it. Executing or accepting the command enters it in the Edit Command and Execute Command submenus. These submenus are dimmed if no commands exist. The Command Output submenu, if not dimmed, displays a list of commands that are currently executing or have command output. Choosing a command in the list displays an output window that shows any information generated by or about the command. 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."

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 1-8, 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 1-8 Malcolm Window styles





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

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 (the second window from the left in Figure 1-8). 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 Malcolm window, such as the Remote Command window or the Color Namer window if it frontmost. If a client window is frontmost, the legend changes to Kill Client. This command causes an abrupt disconnection. Before killing a client read "Logging Off Versus Killing Clients" in Chapter 2. To close all client windows and exit from Malcolm, select Quit from the edit menu. Again, Chapter 2 contains a complete explanation of this procedure.

Ldit Remote Undo	Wir %2
Cut	98 H
Copy Paste	88 C
Clear	æn
Preferences	•
Fonts	₩F
Color Namer.	

Cut, copy, paste, and other editing features

Malcolm enables you to cut, copy, and paste text between a Macintosh application or desk accessory and a Malcolm dialog window that contains text, such as the Remote Command dialog window. 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, Malcolm 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 from a Macintosh file to a client window you must use the client's command for pasting or inserting (not Command-V). Malcolm 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 only functions with desk accessories. Malcolm does not support it. Clear erases a selected item without saving it to the Clipboard.

Preferences

Monochrome Rooted Screen...
Color Rooted Screen...
Windows...
Miscellaneous...

The Preferences submenu, also located in the Edit menu, allows you to select default settings for the items shown in the left margin. Selecting either of the first two items, displays a Rooted Screen Setup window, which allows you to set the size of the root window and specify a display option.

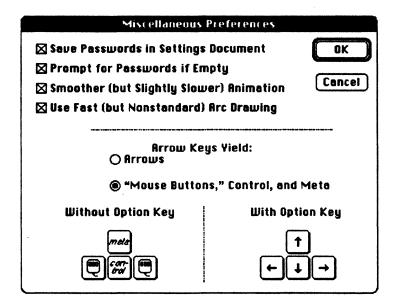
Normally, most people need not alter the default root window preferences. Unless you are an experienced user of X, you would be better off not experimenting with them. However, those who need more information should read "Root Window Preferences" in Chapter 4.

Windows, the next item in the Preferences submenu, permits you to choose a default window style for client windows from the ones shown in Figure 1-8. The next time you start a client, it's window should appear in that style, unless the client specifies its own preference. The window styles in Figure 1-8 apply only to client windows in rootless mode, in the areas outside the boundaries of a visible root window.

Figure 1-9 depicts the last entry in the Preferences submenu, Miscellaneous, which contains five options. Clicking the first checkbox 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 checkbox will cause MacX to prompt you for these passwords when you execute such commands.

The next two options for smoother animation and fast arc drawing should usually be left checked. See Appendix A for an explanation of these options. The last option enables you to choose whether to use the Option key to make the Arrow keys scroll or act as extra mouse buttons. See the section called "Mouse Differences" in Chapter 2 for an explanation.

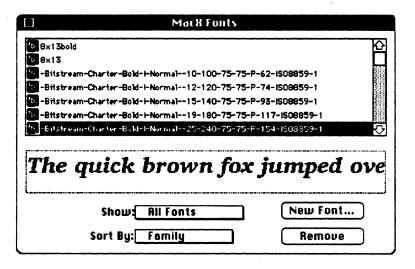
■ Figure 1-9 The Miscellaneous Preferences



Fonts

Choosing Fonts from the Edit menu brings up the Font Director window pictured in Figure 1-10. 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. Since font names tend to be long and unwieldy, the Font Director allows you to include aliases that are shorter and easier to remember as substitutes for font names. Aliases also have a distinguishing icon.

■ Figure 1-10 The Font Director Window

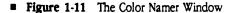


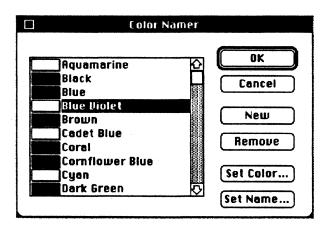
Registry
Foundry
Family
Weight
Siant
Set-Width
Style
Pixel Size
Point Size
K Resolution
Y Resolution
Spacing
Average Width
Character Set

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

Color Namer

The last item in the Edit menu, Color Namer, displays the Color Namer window, when selected. This window, depicted in Figure 1-11, displays a sample of all the colors available for use with clients that support color. 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 order. See Chapter 5 for more information about using the Color Namer.

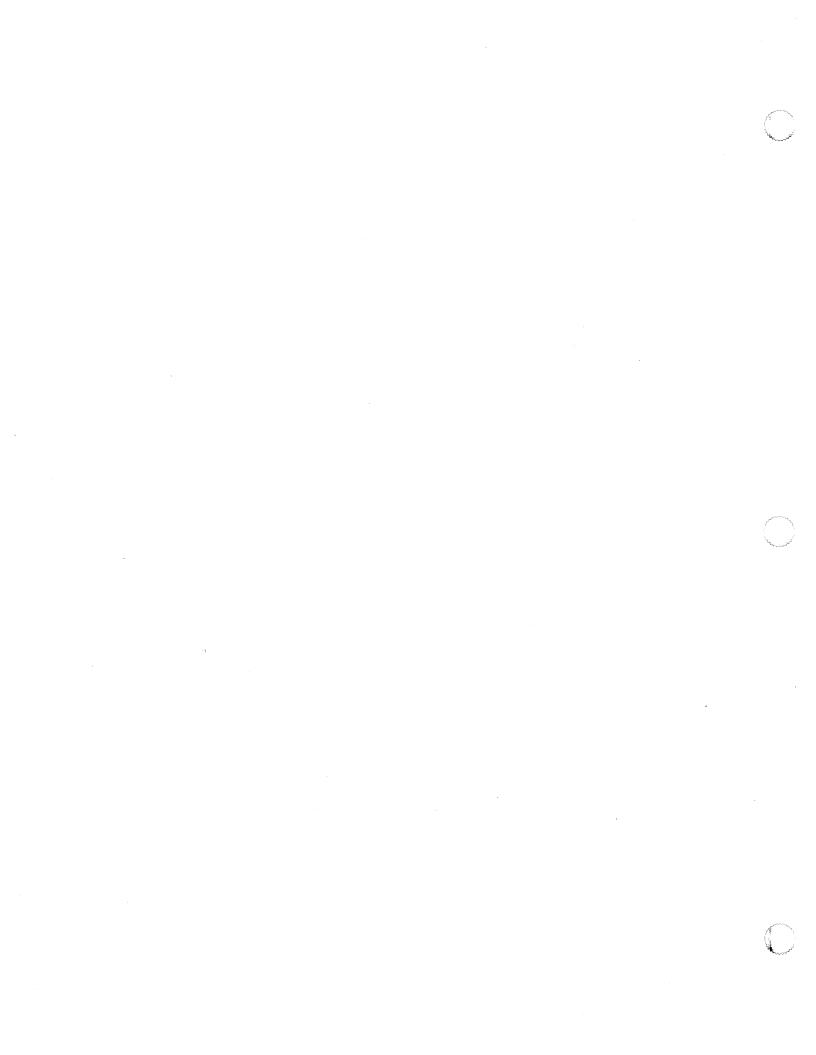




Page Satup... Print Clipboard... ©P Quit #Q

Save, Kill, Quit, and other File menu commands

Like the Edit menu, the File menu contains a number of options 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 settings document, provided that one is not currently open. The Close command closes the settings document currently open. If the file is new, the name "Untitled Settings" 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 Settings," 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.



Getting Started

Before reading this chapter, be sure to install your communications software and Malcolm. Chapter 7 covers installation procedures. Once Malcolm is up and running, you'll be ready to read this chapter and learn the basics about Malcolm.

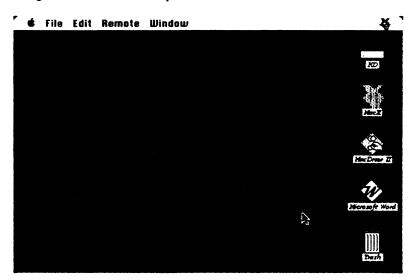
This chapter covers the minimum you need to know to function in an X environment using a Macintosh. You'll learn how to open Malcolm, log on to a host and start a terminal emulator on your screen, log off the host, and, finally, close Malcolm. Once you feel comfortable with these fundamentals, you can consult the remaining chapters to customize your X environment and learn 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.



Opening Malcolm

To open Malcolm, double-click the Malcolm 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 Malcolm menu headings in the menu bar at the top of your screen and, if Multifinder is active, the Malcolm icon at the far right. If you have not already done so, read Chapter 1, "Malcolm Overview," for an explanation of the menus and an overview of Malcolm.

■ Figure 2-1 Malcolm Desktop



Although you don't see it, Malcolm 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 window style you choose, and so forth. When you quit, Malcolm will prompt you to name and save your settings document. If you don't, Malcolm will discard everything you've set up. If you do, Malcolm will create a document icon labeled with the name you have entered. Later, you

can 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 comer—and save it in a settings document called "budget."

Making a fast exit

If you want to exit from Malcolm and you haven't made any changes that you want to save, such as creating a remote command, simply select Quit from the File menu or press Command-Q and Malcolm will close. If you have made changes, the dialog box "Save changes before closing?" will appear. See the section called "Quitting Malcolm" 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

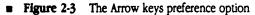
Because the Macintosh uses a one-button rather than a three-button mouse, you cannot issue commands with the mouse like you normally would in other X environments. As an alternative for the missing two buttons, use the arrow keys instead, as shown in the chart in Figure 2-2:

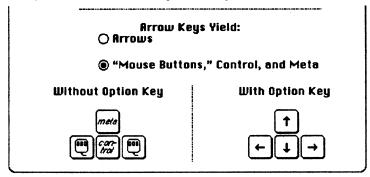
■ Figure 2-2 Substitute mouse buttons

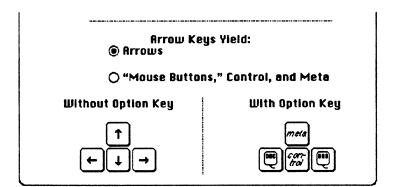
Key	Malcolm Function	
Mouse button	Left mouse button	
Left arrow	Middle mouse button	
Right arrow	Right mouse button	
Up arrow	Meta modifier (mod1)	
Down arrow	Control key	

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

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 Arrows button in the Miscellaneous Preferences window, as shown in Figure 2-3.







To find Miscellaneous Preferences, select the Preferences submenu from the Edit menu and slidedown to Miscellaneous.

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 Malcolm is running.

Substituting keys for mouse clicks

Malcolm 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 or select a field. For such instances, here is a list of substitute keys and the action each performs:

■ Table 2-1 Keyboard substitutes for clicking buttons

Key(s)	Action	When to use
Y, O, 1	yes or OK	Use to select an outlined button 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.
CR, Enter	yes or OK	The carriage return or Enter key. Use to specify <i>yes</i> or OK in any dialog window, dialog box, or alert box, including those with text fields.
N, C, 0	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 <i>no</i> or cancel in any dialog window, dialog box, or alert box, including those with text fields.
Esc	Cancel	Escape key. Use to select the cancel button in any dialog window, dialog box, or alert box, including those with text fields.

Key(s)	Action	When to use
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.

◆ 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

Once you've opened Malcolm, 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 Malcolm application icon and not a Malcolm document icon, pictured at left. If you click a Malcolm 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 Malcolm, 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 UltrixTM (two versions of the UNIX operating system).





Starting xterm

Xterm 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, and with a VAX running Ultrix, Digital Equipment Corporation's implementation of 4.2 BSD UNIX. To start up xterm, follow these steps:

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

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

■ Figure 2-4 The Remote Command window

Edit Remot	e Command
Remote Command: /usr/bin/X11/xterm -ls -n &host -display &display:0.0	Via: MacTCP Host: pluto Zone:
⊠ Execute at Startup Command Name:	Username: ada Password:
Pluto Herm Output: Notify Saved Lines: 100	⊠ Save Password
	Execute Cancel Accept Delete

2. Enter the remote command to start xterm.

Type the command shown in the Remote Command field in Figure 2-4 and press the Tab key. Do not type carriage returns. Your UNIX documentation should explain xterm and all its options; however, some of the options are Malcolm enhancements which require explaining here.

-n ®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 a placeholder for the name of the host you want to contact. You can enter the actual host name if you want to.



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

-display ®display:0.0

The &display option is a placeholder 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.

In the :0.0 extension, the first 0 represents the display station number and the second 0 represents the screen number. The display station number for a Macintosh is always zero because Malcolm can support only one display station. The screen number represents a mode (rooted or rootless) and and a type of screen (color or monochrome). In this case, the number zero (0) specifies a monochrome screen in which the root window does not appear (rootless). Malcolm will display the xterm window on the largest screen that fits these parameters. For further instructions on specifying screen numbers see Table 1-1 in Chapter 1.

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

Execute at startup Click this box so that the xterm window will appear as soon as you open your Malcolm settings document.

Command name: Malcolm automatically enters 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 it, execute it or view its output. If you don't like the name Malcolm has devised, enter another one.

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 Malcolm icon to appear at the right end of the menu bar when Malcolm receives output from the host See Chapter 3 for an explanation of the other output options.

Saved lines: Specify in this field how many lines of command output you want to save. The maximum number is 5,000 lines. The number 100 is probably sufficient because the xterm command will normally produce little or no output.





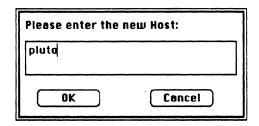




Via: Via is another pop-up menu that displays a list of communications protocols, called **connection tools**, that enable you to contact hosts on which clients (X-based application programs) reside. Select the one used by the host that will receive this command. In Figure 2-4, MacTCP™, Apple's implementation of TCP/IP communications protocol for the Macintosh, is the connection method chosen. If you don't know which connection method to select, contact the system administrator who maintains the host you want to log on to.

Host: This pop-up menu displays the names of the hosts that use the connection method selected in the Via menu. Select the appropriate host name or, if it is not in the list, choose *Other* from the menu. When the dialog box shown in Figure 2-5 appears, type a host name.

■ Figure 2-5 The host name dialog box



Zone: This field is for information purposes only. It indicates your AppleTalk network zone or your MacTCP network domain, depending on which connection method selected in the Via menu. If you selected neither of these tools, the Zone field will not appear.

User name: This field contains the user name recorded in the Chooser (located in the menu). Change it to 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.

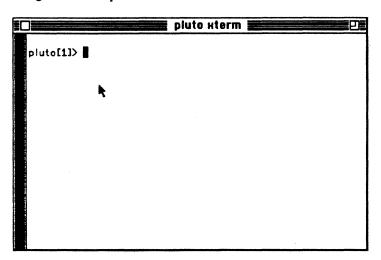
Save password This check box should be dimmed. If not, leave it unchecked for now. For more information, see Chapter 3

Apple Confidential Chapter 2: Getting Started 31

4. Click the Execute button or press the carriage return to execute your command.

After a brief pause, an xterm window similar to the one in Figure 2-6 should appear on your screen.

Figure 2-6 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. See the next section, "Handling Errors," for advice.

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.

You have now successfully transmitted your first remote command to a host computer using Malcolm. 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 can enter commands to start up other clients stored on the host. You'll find starting up clients this way much faster and easier than issuing remote commands.

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 Malcolm, read "The basics of using windows" in Chapter 4.

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

If you want to learn more about issuing remote commands, read Chapter 3.

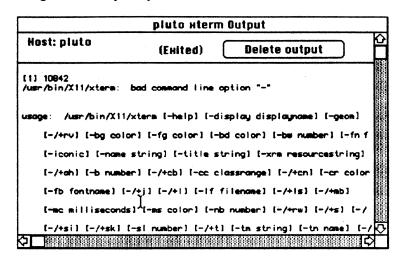
Handling errors

Under certain circumstances an error can occur without Malcolm notifying you of it. This situation 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 a bit friendlier because it makes your Macintosh beep and display a small, blinking Malcolm icon in the menu bar when Malcolm 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.

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, select Command Output from the Remote menu and slide the mouse to the right until you see the short name for your command. Select this name in order to display the output window. You'll see a window, like the one in Figure 2-7.

■ Figure 2-7 Sample output window



This window contains an error message indicating that a hyphen was used incorrectly in the remote command string. Once you determine what your error is, select 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.

Other problems

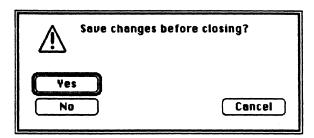
For other types of problems, such as a misspelled host name or a network malfunction, alert boxes describing the error will appear on your screen. Most of these problems should be easy to fix. However, if you need more help, see Chapter 6, "Troubleshooting."

Quitting Malcolm

When you want to exit from Malcolm, 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

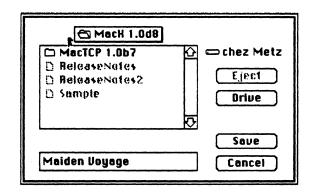
equivalent). Next, choose Quit from the File menu or press Command-Q. The "Save changes before closing?" dialog box, depicted in Figure 2-8, will appear.

■ Figure 2-8 "The Save changes before closing?" dialog box



To save the remote command you created earlier in this chapter and any other changes, such as window styles, click the Yes button. You will next see the "Save as" dialog box depicted in Figure 2-9. As discussed at the beginning of this chapter, Malcolm has created a settings document to preserve everything you have set up, that is, your entire X environment. Unless you save the settings document, Malcolm will discard everything. The document is currently labeled "Untitled Settings." Enter another name in the highlighted field and either press the Return key or click the Save button. Malcolm 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.

■ Figure 2-9 The "Save as" dialog box

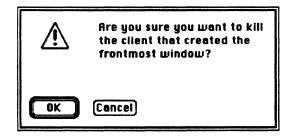


Logging off versus killing clients

You may be tempted to close your terminal emulator window by clicking the close box or by choosing Kill Client from the File 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. 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.

Fortunately, Malcolm does not immediately execute a Kill Client request. A dialog box appears first inquiring whether you really want to kill the client (see Figure 2-10).

■ Figure 2-10 The Kill Client dialog box



Closing a settings document without quitting Malcolm

To leave the environment in which you are currently working without quitting Malcolm, choose Close "Untitled Settings" from the File menu. Both the "Save changes" and the "Save as" dialog boxes will appear as described above. Follow the same procedure to save and rename your untitled settings document. Malcolm will create a document icon labeled with the name you have entered, just as it does when you specify Quit. The difference is the Malcolm application will still be open. The Malcolm menus will still be there; however, you will only be able to create a new settings document or quit Malcolm, unless you reopen an existing Malcolm settings document.

Using Remote Commands

HAPTER 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 Malcolm 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. Malcolm will prompt you for a name, if the document is new, or else ask you if you want to save your changes.

The Remote menu

Malcolm enables you to issue commands to a remote host using the Remote menu illustrated in Figure 3-1. This menu is always located in the menu bar at the top of the screen.

■ Figure 3-1 The Remote menu



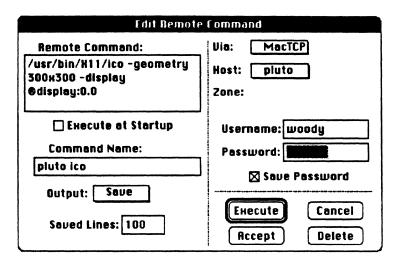
Choosing New Command from the menu causes a dialog box to appear in which you can type a new command and execute or save it, if desired. Edit and Execute Command are submenus containing a list of existing commands. Choosing a command from the list enables you to edit or execute an existing command. Selecting the Command Output submenu displays a list of commands that are currently executing. To see the output and status of a particular command, select its name from the list. For more information on using these submenus, read "Viewing, editing, and executing commands" at the end of this chapter.

Use remote commands sparingly: Remote commands are a very timeconsuming and memory-intensive way of starting up clients. For quicker results, start up a terminal emulator and issue commands through it to launch clients.

Creating new commands

To create a new command select New Command from the Remote menu. The Remote Command window, illustrated in Figure 3-2 will appear.

■ Figure 3-2 The Remote Command window



This box contains a number of text fields and pop-up menus. All the fields scroll up and down and from left to right. To scroll through them, use the mouse or the arrow keys.

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, if desired. Type the command the way you normally would, using the syntax required by the host for which the command is destined, 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, 6, when using these macros so that the Malcolm server can distinguish the macro name from regular text.

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

®display

Replaced with the display name or network address 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, 90.1.3.13, for example.

®host

Replaced with the host name specified in the host field of the Remote Command dialog window. 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.

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 bypass this option later, hold down the Option key while opening the settings document.

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. Malcolm automatically places a name here composed of the host name and either the first word in the command string or, if it's too long, the last part of the first word. (Alan: What did you decide to do here?) Malcolm defines a word as a collection of alpha-numeric characters separated by blank spaces. The characters include A through Z, 0 through 9, underscore (__), dash (__), period (.), and dollar sign (\$). Accordingly, in the sample command shown in Figure 3-2, Malcolm indentifies /usr/bin/X11/ico as the first word. It combines the last part, "ico," with the host name "pluto" to produce the short name "pluto ico." (Note that the slash (/) does not belong to set of alphanumeric characters defining a word.) If you don't like the name Malcolm has devised, change it to something else.

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. Malcolm will put this output in the command's Output window as it arrives.

Notify

This selection causes a beep and displays a small, blinking Malcolm icon at the right end of the menu bar. To cancel the notification and see the command output, select the command name from the Command Output menu. A Macintosh window displaying the output will appear, 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 auto-

matically appear on the screen. If you are using Multifinder, Malcolm must be in the foreground for the window to pop

up.

Save If you want to save command output, but not be notified,

select this option. When you want to see the output, select

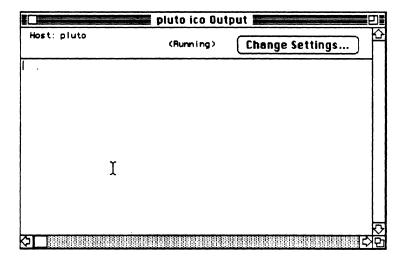
the command's short name from the Active menu.

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



Saved Lines:

Enter the number of lines of command output you want to save here. Up to 5,000 lines can be saved.





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

· Via:

Via is another pop-up menu that displays all the communications protocols, or connection tools, installed by means of the Macintosh Communications Toolbox. Select the connection tool used by the host that will receive your command. In Figure 3-2, the connection method chosen is MacTCP, Apple's implementation of TCP/IP communications protocol for the Macintosh.

Host:

This pop-up menu, also shown in the margin, displays the host names presently associated with the connection tool selected in the Via menu. To add a new host name, choose *Other* from the menu.

Zone:

Zone lists the network zones in AppleTalk® or the network domains in TCP/IP. This field does not appear if the connection method in use does not support zones or domains. The Via field shows what connection method is currently selected.

User name:

This field automatically shows the user name recorded in the Chooser. Change it to the ID assigned to you on the host receiving your command.

Password:Save password

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 chose "Save Passwords..." in the Preferences submenu, 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, Malcolm will automatically send the password to the host. Malcolm encrypts all passwords in the settings document, thus ensuring a reasonable measure of security for most business purposes.

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, the Remote Command dialog window will appear so that you can type your password in the Password field.

Action Buttons

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

Execute

Executes the command and automatically saves it

Accept

Saves the command without executing it

Cancel

Erases a new command and the current revisions of an

existing command

Delete

Erases the command

Saving passwords

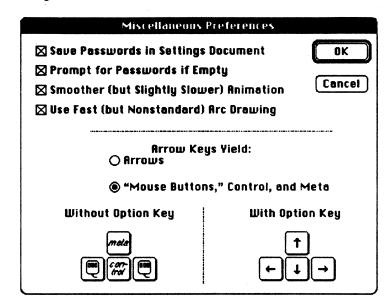
If you would rather not type your passwords every time you issue a remote command, you can have Malcolm save them in the settings document. To save passwords, choose Miscellaneous from the Edit menu. Figure 3-4 shows the Preferences submenu.

■ Figure 3-4 Selecting Miscellaneous Preferences

Edit		
Undo	% Z	
Cut	₩ H	
Сору	₩ C	i
Paste	36 U	
Clear		
Preferences		Monochrome Rooted Screen
Fonts	96 F	Color Rooted Screen
Color Namer		Windows
		Miscellaneous

When the Miscellaneous Preferences window appears, click the Save Passwords checkbox and click OK. As a result, the corresponding Save Password checkbox in every Remote Command window will be highlighted so that you can check it, if desired.

■ Figure 3-5 Miscellaneous Preferences window



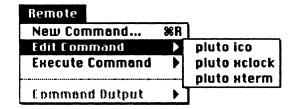
Malcolm encrypts all password in the settings document, providing a level of security sufficient for the needs of most businesses. Taking additional precautions to protect secret or sensitive information, however, is a wise practice.

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 Passwords If Empty." This option ensures that Malcolm 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.

Viewing, editing, and executing commands

The Remote menu has three submenus that enable you to edit and execute existing commands and view the output of commands currently being executed. Figure 3-6 shows an example of the Edit Command submenu. Each item listed is the name of a command. When you select one of the items, the dialog window for that command appears and you can edit, delete, or execute it.

■ Figure 3-6 An example of the Edit Command submenu



The format of the Execute Command and the Command Output submenus is the same. However, when you select a command from the Execute Command submenu, the Remote Command window does not appear.

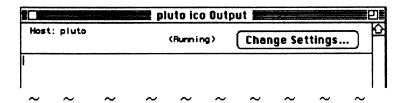
Selecting an item from the Command Output menu displays the Output window for that command, like the one in Figure 3-3. For more information about the Output window read the following section.

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 the Output window, select the command's short name from the Command Output submenu. The window appears, regardless of the method chosen to handle output (see the option displayed in the output field of the Remote Command dialog window). If you have selected the Discard option, however, the window will be empty. To make the Output window pop up automatically when a command starts executing, select the Pop up option. The window will pop up as long as Malcolm is in the foreground.

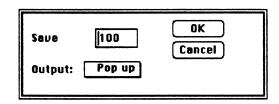
Figure 3-7 illustrates the top portion of the Output window. The name of the host on which the command is running appears on the left, in this case "pluto." A word indicating the status of the command appears in the middle, in parentheses: either "running" or "exited."

■ Figure 3-7 Top portion of the Output window



Clicking the button on the right enables you to change the output status of your command. A dialog box appears, as shown in Figure 3-8.

■ Figure 3-8 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. You can also change the number of command output lines you want saved in the Save Lines field. Remember that the Output window will contain no more than the number of lines you specify here. Malcolm 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.

Handling Windows

HAPTER 1 COVERED THE ESSENTIALS ABOUT WINDOWS, INCLUDING HOW THEY function in rooted and rootless mode. This chapter describes how to handle windows in more detail. Also covered are instructions for setting root window and window style 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 Malcolm. In addition to these familiar features, though, Malcolm 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. Be sure to have an active client window on your screen before attempting to use this menu; otherwise, the menu items are dimmed. For instructions on starting up a client, consult Chapter 2 "Getting Started" or Chapter 3 "Using Remote Commands."

■ Figure 4-1 The Window menu

Window	
iconify	% 1
Set Window Style	•
Temporarily Adorn	兼和
Close Window	≋ ₩

Styling and manipulating windows

While Malcolm 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 Malcolm 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 Malcolm creates for the client. They can make requests, called *hints*, but, ultimately, the X server, 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 comer. If you prefer a different style, follow these simple steps:

- 1. Bring the window you want to restyle to the front by clicking on it.
- 2. Select Set Window Style from the Window menu to display the five window styles.

Figure 4-2 shows these styles. The one currently in effect will have a check mark next to it. Simply highlight another style to change the selection.

Client Specified, the last item in the Set Window Style menu, reverts the window style to the one specified by the border width of a top-level client window. See the next section, "Specifying a window style in a command," for more information.

■ Figure 4-2 Malcolm 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" for details.

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 is specified by the client when it is invoked. In rootless mode, this border is not displayed, however, you can use the border width to specify one of the window styles in Figure 4-2, provided the client supports the border width option. When entering a command to start up a client, specify a window style by changing the argument to the border width option.

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	Title bar
2	Title bar and size box
3	Rounded corners, solid title bar
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

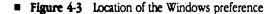
The Set Window Styles submenu contains one last item, Client Specified. Selecting this item causes the window style to revert to the one specified by the border width of the client window. For example, if you issued a command to start up xclock with a border width of 4 (shadow box) and later change it to 1 (title bar), selecting Client Specified will revert the style of the xclock window to a shadow box. If you did not specify a border width, clients usually have a default border width that Malcolm will convert to a window style. Should this default argument be out of range (less than 1 or larger than 5), Malcolm substitutes a "2" in its place.

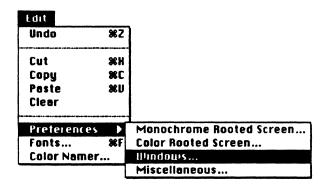
In a rooted environment, where the root window is visible, the border width option functions as originally intended, that is, it specifies the thickness of the window border.

Setting window preferences The Edit menu contains a Windows preference that sets a default window style for client windows (see Figure 4-3). The default takes effect the next time clients starts up, unless you override the default by issuing a command with a border width option (see the previous section, "Specifying a window style in a command" for information about this option). The window styles in Figure 4-2 apply only



to client windows in rootless mode, where the Malcolm Window Manager is in control.





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 grow bar, use Temporarily Adorn to move or size it. First, make sure the window is active. Next select Temporarily Adorn from the Window menu. Your window will acquire a temporary title bar and size box so that you can manipulate it like any Macintosh window. You have one opportunity to move or size your 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.

Stacking windows: Malcolm 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. Malcolm also allows you to move and resize a root window displayed on your screen just as you would a client window. Actually, this so-called root window is a Macintosh window through which you can view 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. 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.

Scrolling quickly with the hand: To scroll quickly in root windows, change the mouse pointer to a hand by holding down the Option key and then move the root-viewing window around by holding down the mouse button and moving the mouse.

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

Converting windows to icons

them, that is, convert them into icons on your screen. You can iconify both client windows and root-viewing windows. In the case of client windows, the client can determine what the icon looks like, but if it doesn't specify anything, Malcolm 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 selecting Iconify from the Window menu.

An alternative to moving windows when they get in the way is to *iconify*



A window icon



Root window

Malcolm provides three 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
- Hold down the Option key and click the window's zoom box, if it has one. For a root-viewing window, click the close box.

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

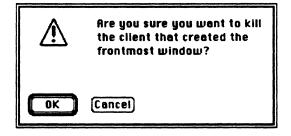
Killing clients and closing windows

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 Malcolm first starts up and no windows have opened, the Window menu displays Close Window, as a dimmed item, in that spot. When enabled by an active Malcolm dialog window, such as the Remote Command window, selecting 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 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, Malcolm does not immediately execute a Kill Client request. An alert box appears first inquiring whether you really want to kill the client (see Figure 4-4).

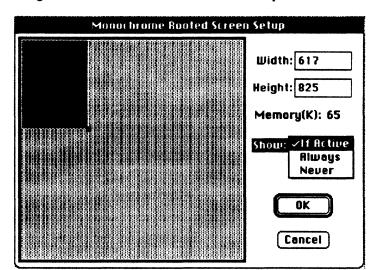
■ Figure 4-4 The Kill Client alert box



◆ Another way to kill: Clicking the close box in the left-hand corner of a client window is equivalent to choosing Kill Client.

Root window preferences

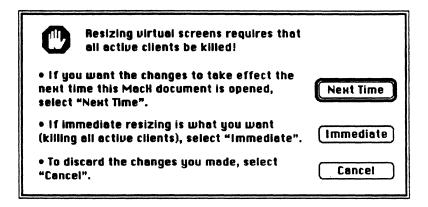
The Preferences submenu, located in the Edit menu, allows you to select default settings for a monochrome or color root window. Selecting either of the first two items, displays a Rooted Screen Setup window. Figure 4-5 shows the monochrome version; the format for the color root window dialog is the same. Both dialogs 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. You can also select one of three display options: *always* display a root window, *never* display a root window, or display a root window when a client that requires one is *active*.



■ Figure 4-5 The Monochrome Rooted Screen Setup window

Any change to the Show option takes effect immediately after you click OK. Should you attempt to resize a root window that has clients running in it, however, Malcolm displays an alert box, shown in Figure 4-6.

■ Figure 4-6 Root window warning message



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

Managing Fonts and Color

HAPTER 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 Malcolm

The method devised for implementing fonts in Malcolm 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, located in your System file, holds files that contain the standard set of X fonts. Each file contains one variation of a font. For example, one file contains Helvetica 18 point bold, a variation of the Helvetica font family. Malcolm stores a list of these fonts in the Font Directory file, also located in the Font 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 file names. Consequently, files in the Font folder have shorter, more obvious names, like Charter 10 bold.



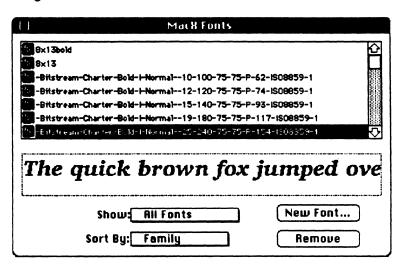
The primary purpose of the Font Directory file is to supply the Font Director with a current list of X fonts, any Macintosh fonts you have chosen to use in Malcolm, and any font aliases you have created to act as substitutes for font names or other font aliases. Malcolm keeps the file up to date by revising it any time you make a change that alters the modification date of the font folder. Typically, these changes include adding, deleting, or renaming a font file. Compiling a font in the MacX Fonts folder also updates the Font Directory file.

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 type face. Besides viewing fonts, you can sort fonts in a variety of ways, assign aliases to font names, and compile fonts from BDF files in a straight-forward 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





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.

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 (specifically an ISO Latin 1 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 their order of appearance in the font name string, starting from the left side. For more information, refer to Appendix D.

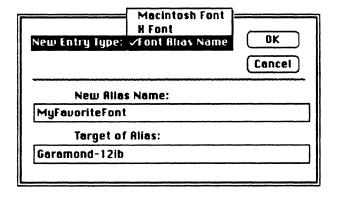
■ Table 5-1 X logical font name format

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, e.g. bold or medium
Slant	A one or two-letter code indicating the posture of the font, e.g., R for Roman, I for italic
Set-width name	One or two words designating the width of a font, e.g. Normal, Condensed, Double Wide
Add style name	A term that identifies additional type style information, e.g., 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)
Spacing	A letter that indicates the escapement class or pitch of a font, e.g., 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 a each character in a font with its glyph or image

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, select a font name in the Font Director window and click the New Font button. A dialog box illustrated in Figure 5-2 will appear. The font name you selected should appear in the Target of Alias field. Click the New Entry Type pop-up menu and select Font Alias Name, as shown in Figure 5-2. Then, type any combination of words, letters, or numbers you prefer in the New Alias Name field and click OK. The Font Director will insert the alias, along with any other aliases you create, into of the font name list. Aliases will be sorted, along with other font names, according to the font element you have selected from the Sort By pop-up menu. For more information about this process, see "Sorting Fonts" later in this chapter

■ Figure 5-2 Entering a font alias



Copying and pasting font names

Using the Macintosh Copy command enables you to 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 Command-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 Command-V or by choosing Paste from the Edit menu.

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 a font or font alias from the Font folder and deletes its name from the Font Directory file and the font names list in the Font Director window. 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 5-3). Click OK to remove the font or Cancel to retain it.

■ Figure 5-3 Removing a font



Registry
Feundry
Feundry
Weight
Slent
Set-Width
Style
Pixel Size
Pixel Size
Pixel Size
W Resolution
Y Resolution
Spacing
Ruerage Width

Sorting fonts

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-1-normal--12-120-
-adobe-times-bold-1-normal--14-140-
-adobe-times-bold-1-normal--22-220-
-adobe-times-bold-1-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-
```

63

The following font name elements are numeric:

Pixel Size
Point Size
X Resolution
Y Resolution
Average Width

Compiling fonts from BDF files

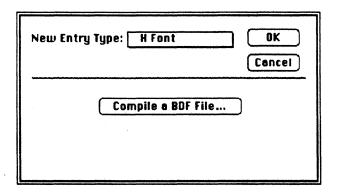
The Font Director enables you to compile BDF files into fonts without exiting from Malcolm. Although BDF files can be compiled in the MacX Fonts folder, compiling them in another folder and moving them to the fonts folder afterwards saves considerable time. You can trigger Malcolm to update the Font Directory file once you have moved the compiled font files into the fonts folder by opening the folder. Otherwise, the Font Directory file will automatically be updated the next time you start up Malcolm.

To compile a font, follow these steps:

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

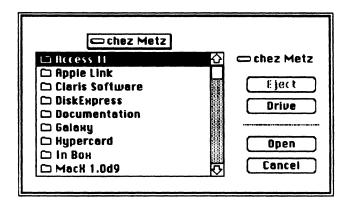
When the New Entry dialog box appears, select X Font from the New Entry Type pop-up menu. The layout of the box will change, as depicted in Figure 5-4.

■ Figure 5-4 Compiling an X font



2. Click the Compile a BDF File button.

The Directory dialog box will appear.



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 on the bar will reveal the hierarchy of directories (folder names) through the top level. Upon finding the right directory level, if your folder isn't visible you can type a letter, word, or number to scroll directly to the folder name, if desired.

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

An alert box advising you that the font compiler is compiling your BFD source file will appear. When the box disappears, the compiled font file will be in your folder.

The font compiler is compiling your BFD source file.

Please stand by...

5. Repeat steps 2 through 4 for as many files as you want to compile. When you have finished, drag the files into the MacX Fonts folder in your System file.

Malcolm will update the Font Directory file. This process will take at least 30 seconds. If you don't care to wait that long, defer this step until later.

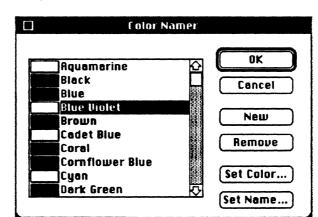
Using Color in Malcolm

In a departure from other X servers, which represent colors in numeric form only, Malcolm 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 Malcolm, these sets of numbers are stored with corresponding color names in the MacX Colors file. Malcolm converts these values into colors and displays them in a dialog window called the Color Namer.

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 5-5, 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.



■ Figure 5-5 The Color Namer window

Sorting, scrolling, saving, and cancelling

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 upper and lowercase, spaces, and diacritical marks, such as an umlaut ("). Any numbers at the end of a color name are sorted numerically, as whole values. For example, the color Grey8 would appear before the color Grey10.

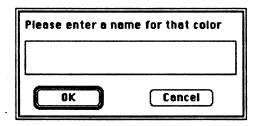
Scrolling To expedite the task of finding colors in a long list, the Color Namer supports all the standard Macintosh scrolling methods. Besides responding to the scroll bar and the directional arrows (illustrate up, down, left, right arrows here), the Color Namer will scroll to names in the list that 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 "grey" and the Color Namer will scroll to the first color named "grey" in the list. Similarly, typing "50" will produce the first color name starting with that number.

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 will prevent it from taking effect (along with any other changes as well). Clicking the close box also closes the Color Namer. However, if you modified any colors or color names, an alert box appears asking whether you want to save these changes before closing the Color Namer.

Adding new colors

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

■ Figure 5-6 Entering a new color name



Next, the Color Wheel dialog box, displayed in Figure 5-7, will appear. Clicking anywhere in the color wheel will display the selected color in the top half of the box directly beneath the legend "Please select a 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.

■ Figure 5-7 The Color Wheel dialog box

The scroll bar to the right of the color wheel controls color brightness. Scrolling the elevator 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.*

Removing colors

To remove a color, click its color name and then the Remove button (see Figure 5-5). A warning box enquiring "Are you sure you want to remove that color?" will appear. 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 5-7 will appear. At the top left, the sample color box displays the existing color. As you click the color wheel, the top half of the box will reflect 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 will reappear showing the changed color.

If, for some reason, you made a mistake and want the original color back, selecting Cancel will 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 will restore the color, too.

To change a color name, select the name by double-clicking or using the Set Name button. The Color Namer will display a dialog box like the one in Figure 5-6 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 Command-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.

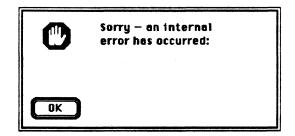
Troubleshooting

ONSULT THIS CHAPTER WHEN MALCOLM 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 Malcolm itself. If the problem does involve Malcolm and you aren't able to find a solution in this chapter, the Technical Support Group at Apple Computer is ready to help.

About errors

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

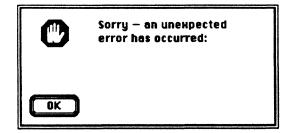
■ Figure 6-1 Sample internal error alert box



Usually, you can do nothing to fix this kind of problem yourself. Any time Malcolm displays an internal error message, call Technical Support at Apple Computer.

Another set of errors will make Malcolm quit or prevent Malcolm from starting up. The phrase identifying these errors is "Sorry—an unexpected error has occurred...." Figure 6-2 illustrates the alert box displayed in this case.

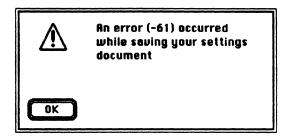
■ Figure 6-2 Sample alert box for an unexpected error



Fortunately, you can usually recover from these types of errors. Refer to the section in this chapter that covers the type of problem you encounter.

The last type of error message Malcolm 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.

■ Figure 6-3 Sample warning message



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

Problems opening Malcolm

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

Could not open the default font

Malcolm 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. Malcolm will not open if it cannot open this default font.

If you see the message "...could not open the default font.," follow these steps:

- 1. Verify that the "fixed" font file is in your MacX Fonts folder.
 - 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 MacX Fonts folder on your master disk into the MacX Fonts folder in your System Folder.
- 2. Try opening your Malcolm application or settings document again.

Could not open the default cursor font

A font, called "cursor," comprises all the cursors used in X. Malcolm will not open if it cannot open the cursor font.

If you see the message "...could not open the default cursor font.," follow these steps:

- 1. Verify that the "cursor" font file is in your MacX Fonts folder.
 - 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 MacX Fonts folder on your master disk into the MacX Fonts folder in your System Folder.
- 2. Try opening your Malcolm application or settings document again.

Unable to open MacX Temp

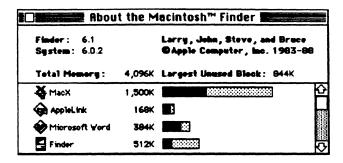


The message, "...unable to open MacX Temp.," appears when you try to launch Malcolm or a Malcolm settings document and Malcolm cannot create the MacX Temp file. This problem can occur for a number of reasons. Read the following check list to ascertain which problem applies in your case and what corresponding solution to apply:

1. Do you have sufficient RAM?

Select About the Finder in the menu and check the Total Memory field to see how much RAM your Macintosh has. (In Figure 6-4, 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 Malcolm again.

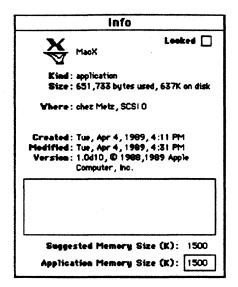
■ Figure 6-4 About the Finder display



If your system has plenty of memory available to support Multifinder, you still may not have enough memory left over to run Malcolm. To determine whether this case applies, click the Malcolm icon, select Get Info from the File menu, and check the field labeled Application Memory Size (K). See Figure 6-5 for an illustration. This amount should not be larger than the largest unused block shown in "About the Finder." If it is, you have a number of alternatives.

- Close other applications running under Multifinder. In Figure 6-4, AppleLink® and Microsoft Word could be closed to free more memory.
- Remove unnecessary inits (initialization programs), 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, if you are running Multifinder, reboot your Macintosh to make these changes take effect.
- □ Select 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 Malcolm below the size of the largest unused block. Malcolm requires a minimum of 768K to operate. Make sure Malcolm isn't locked or you won't be able to make this change; the box labeled Locked in the upper right comer of Get Info should not be checked.

■ Figure 6-5 Get Info Display for Malcolm



As a last resort, if none of these alternatives is a good solution in your case, you can always buy more RAM for your machine.

2. Is your System Folder on a floppy disk?

Make sure the disk isn't write-protected. Verify you cannot see through the square hole on the top right side of your disk. Otherwise, slide the black locking tab down so that the hole is blocked.

3. 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* contains instructions for using Disk First Aid.

After fixing the problem that occurred in your case, check your System Folder and remove any existing MacX Temp file. Be sure to quit Malcolm first and then drag the MacX Temp file from your System Folder to the trash. Now try starting up Malcolm again.

No connection tools are working at the moment

When starting up Malcolm, if the error message "...none of the connection tools are working at the moment. Check that they are properly installed." appears, check with your network administrator to verify that the network is up. If the network is up, consult the installation procedures for your connection tools to verify they are installed correctly. If everything is in order, restart your Macintosh and try starting up Malcolm again. If the same message recurs, call Technical Support for further assistance.

Error opening or reading a settings document

If your settings document is corrupted, Malcolm will display the warning message, "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 Malcolm has opened a new settings document since it could not open the one you specified. Be sure to throw this damaged document away.

Auto Execute commands could not be started

The full text of this warning message is "One or more 'Auto Execute' commands could not be started because of insufficient memory." Although your settings file was able to open, your system does not have enough RAM to automatically execute one or more remote commands. To ensure that you have at least 768 Kilobytes of RAM available to run Malcolm, select About the Finder in the menu. If the Finder indicates less than 768K of RAM is available, try one or more of the following alternatives:

- □ Close other applications running under Multifinder. In Figure 6-4, AppleLink and Microsoft Word could be closed to free more memory.
- Remove unnecessary inits (initialization programs) and cdevs (Control Panel devices) from your System Folder. Simply drag the inits and cdevs you don't need out of your System Folder. In both cases, if you are running Multifinder, reboot your Macintosh to make these changes take effect.
- □ Select 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 Malcolm below the size of the largest unused block. Malcolm requires a minimum of 768K to operate.
 Make sure Malcolm 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.

If none of these alternatives is a good solution in your case, you can always buy more RAM for your machine.

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.

Host name not translated into a network address

If, while preparing to execute a remote command, Malcolm cannot translate the host name specified into a network address, the warning "Remote command did not succeed: the specified host name could not be translated into a network address." will appear. To correct this problem, you must properly define the host name and its corresponding network address. Consult the documentation for the connection tool being used for instructions or contact your network administrator for assistance.

Network communications error occurred

The warning "Remote command did not succeed: a network communications error occurred.," can appear for an number of reasons, enumerated below:

1. The host specified in the remote command or the network is down.

Call your system and network administrators to verify the status of the host and network.

2. The connection tool specified in the remote command is malfunctioning.

Consult the troubleshooting section of the documentation for that connection tool.

3. Malcolm has depleted its allocated memory.

Some operations in Malcolm use a lot of memory. Furthermore, doing a lot of things at once can also deplete memory resources allocated to Malcolm. 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. (How can one tell?)
- 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 to 100 lines or fewer.
- □ Close the Font Director or Color Namer windows if they are open.

Internal error

When you see the message, "Remote command did not succeed: internal error," call Technical Support for assistance.

Not enough memory

Malcolm displays one of two messages when it runs out of memory while processing a command. These messages are: "Remote command did not succeed: there was not enough memory to perform the operation." and "There is not enough memory to store the remote command in the Malcolm document." The latter warning will appear when you attempt to save or execute a remote command you have just edited or created and Malcolm does not have enough memory to store it. To create more memory, follow these steps:

1. Verify that Malcolm hasn't used up all the RAM allocated to it.

Some operations in Malcolm use a lot of memory. Furthermore, doing a lot of things at once can also deplete memory resources allocated to Malcolm. 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. (How can one tell?)
- 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 to 100 lines or less.
- □ Close the Font Director or Color Namer windows if they are open.

2. Allocate more memory to Malcolm.

First, quit from Malcolm if it is running. Next, click the Malcolm icon and select Get Info from the File menu and increase the amount of memory in the field labeled Application Memory Size (K). (See Figure 6-5 for an illustration of the Get Info display.) The minimum amount of memory Malcolm needs to operate is 768K, but 1,200K is preferable. Depending on your system configuration and how you use Malcolm, however, you may need more or less than 1,200K. In any case, be sure not to make the application memory size close to or larger than the total RAM in your Macintosh. To find out what the total memory is select About the Finder from the menu.

Problems with fonts

Messages in this section describe errors that prevent Malcolm 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.

Error updating Font Directory file

The full text of this warning is "An error occurred while attempting to update the 'Font Directory' file. It will be left unchanged." You will probably not be able to open new fonts until the Font Directory file can be updated. Since this problem can occur for a number of reasons, perform the instructions for each of the following steps until you solve the problem that applies in your case:



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. Is the disk you are using defective?

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

3. Does Malcolm have sufficient RAM?

Some operations in Malcolm use a lot of memory. Furthermore, doing a lot of things at once can also deplete memory resources allocated to Malcolm. 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. (How can one tell?)
- 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 to 100 lines or less.
- □ Close the Remote Command, Font Director, or Color Namer windows if they are open.
- Delete the remote command output for commands that have exited.
 To delete this output, select the Command Output submenu from the Remote menu.

4. Is your System Folder on a write-protected disk?

Make sure the disk isn't write-protected. If it is a floppy disk, verify you cannot see through the square hole on the top right side of your disk. Otherwise, slide the black locking tab down so that the hole is blocked.

5. Is your disk full?

Check the upper right corner of your disk directory folder for the amount of available RAM.

A font property could not be loaded

If a font file is corrupted, Malcolm will display the warning "For some reason, the font property of the font file font name 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 Malcolm distribution disk or compile a new font file. See Chapter 4 for instructions about compiling fonts.

Problems saving a settings file

Error occurred while saving document

Several types of errors can prevent Malcolm from saving a settings document. Each of these errors generates the warning "Sorry—an error occurred while saving your document." Perform the instructions for each of the following steps until you solve the problem that applies in your case:

1. Does Malcolm have sufficient RAM?

Some operations in Malcolm use a lot of memory. Furthermore, doing a lot of things at once can also deplete memory resources allocated to Malcolm. 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. (How can one tell?)
- 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 to 100 lines or less.
- Close the Font Director or Color Namer windows if they are open.

2. 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* contains instructions for using Disk First Aid.

3. Is your System Folder on a write-protected disk?

Make sure the disk isn't write-protected. If it is a floppy disk, verify you cannot see through the square hole on the top right side of your disk. Otherwise, slide the black locking tab down so that the hole is blocked.

4. Is your disk full?

Check the upper right corner of your disk directory folder for the amount of available RAM.

Installation

O INSTALL MACX ON YOUR MACINTOSH SIMPLY VERIFY THAT YOU HAVE THE right equipment and software and then follow the instructions to set up a communications link and install MacX. Besides providing this information, this chapter illustrates different ways you can set up equipment to best suit your needs.

Apple Confidential Chapter 7: Installation 87

What you need to run MacX

Read this section to ensure that you have the equipment, system software, and communications software required to run MacX properly in your network environment.

Hardware

MacX, version 1.0, will run on any Macintosh with at least 1 Megabyte of memory, a 128K or larger ROM, and a minimum of two floppy disk drives. With the exception of early models, the current line of Macintoshes, ranging from the Macintosh IIx through the Macintosh Plus meet these requirements. MacX will operate at its best, however, if your Macintosh has 2MB or more of RAM and a hard disk drive.

Like all X servers, MacX requires monitors with bit-mapped screens. MacX will support both color and monochrome displays.

Software

MacX requires that Macintosh operating system software (MacOS) be version 6.0.2 or later. MacX is fully compatible with MultiFinder, but requires at least 2MB of RAM for best performance.

Communications

Version 1.0 of MacX uses the Communications Toolbox to manage communications and to install connection tools, an Apple term for network and transport-layer communications protocols. The connection tools supplied with the first version of MacX are MacTCP, Apple's TCP/IP product for the Macintosh, and ADSP (AppleTalk Data Stream Protocol). Third-party vendors are supplying other connection tools. Basically, MacX will support any duplex-reliable, 8-bit byte-stream protocol that operates under the Communications Toolbox.

Your MacX package

Your MacX package should contain the items illustrated in Figure 7-1:

- Figure 7-1 What you should have
- Three MacX disks (how labeled?)
- One disk containing the Communications Toolbox and the MacTCP and ADSP connection tools (Comm Toolbox and ADSP provided later.)
- The MacX User's Guide
- The MacTCP Network Administrator's Guide
- This installation booklet
- A license agreement and software registration card

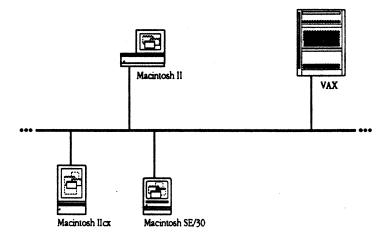
If any of these items are missing, contact the authorized Apple dealer from whom you purchased the product.

Sample configurations

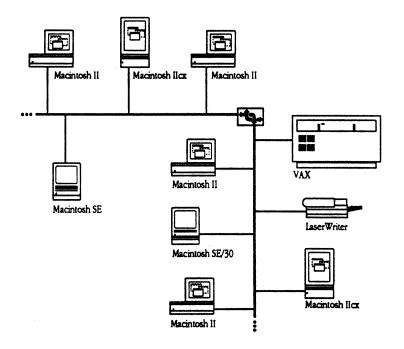
MacX will operate in a number of network configurations. To demonstrate the extent of these possibilities, Figures 2 through 4 present examples ranging from the simplest setup to an extensive multiple-network configuration.

Apple Confidential Chapter 7: Installation 89

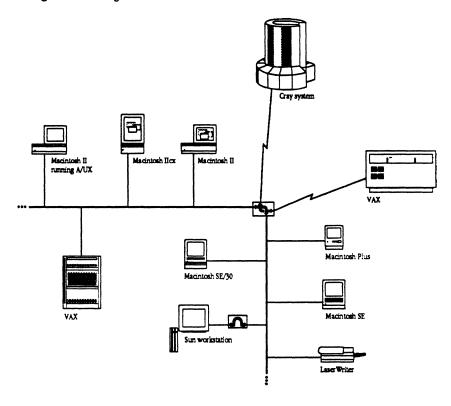
■ Figure 7-2 A small Local Area Network



■ Figure 7-3 A small internet



■ Figure 7-4 A large internet



Installing the Communications Toolbox

The Communications Toolbox is a nifty piece of software that makes your connection tools talk to MacX whether they want to or not...

Congratulations, you finished installing the Comm Toolbox! Now you're ready to install a connection tool.

Which tool should I install?

If you don't know whether the hosts with which you want to communicate on your network are using AppleTalk or TCP/IP protocols, call your network or systems administrator. Once you've identified which tool to install, you will need to get configuration information pertaining to that tool. Therefore, before ending your conversation, turn to the section about installing ADSP or MacTCP and get the additional information you need.

Installing ADSP

This section describes how to install an ADSP connection tool. The connection tool disk that came with your MacX disks contains an ADSP driver, software that enables your Macintosh to communicate over an AppleTalk network. Because the low-level AppleTalk protocols are already included in your system software, installing the ADSP driver takes just a couple short steps.

Installing MacTCP

This section describes how to install a MacTCP connection tool. The connection tool disk that came with your MacX disks contains a MacTCP driver, software that enables your Macintosh to communicate over a TCP/IP network. The procedure for installing MacTCP includes dragging the MacTCP driver into your System Folder, then bringing up the Control Panel and configuring the driver to conform with your network's implementation of TCP/IP. In order to configure the driver properly, you need to call your network administrator and ask him or her a variety of questions. This section explains this whole process. It addresses the needs of the network user who simply wants to connect a Macintosh to an existing TCP/IP network. If you are a network administrator or are installing a TCP/IP network, please consult the MacTCP Administrator's Guide, included in the product package.

Follow these steps to install MacTCP on a hard disk or a floppy disk:



1. Lock your connection tool disk before inserting it into your floppy disk drive.

Locking the disk ensures that you don't accidentally erase something while you are transferring MacTCP onto your hard disk. The disk is locked when you can see through the square hole on the top right side of the disk.

2. Insert the disk in your floppy disk drive and double-click the disk icon to open the disk.

Figure 7-5 illustrates the contents of this disk.

Figure 7-5 Contents of the connection tool disk

If you do not have a hard disk, insert a floppy disk that contains a System Folder with version 6.0.2 or later of the Macintosh operating system into your other floppy drive. Label this disk "MacX System Disk" or something similar.

- 3. Drag the MacTCP icon into your System Folder
- 4. Open the Control Panel, scroll until you see the MacTCP icon, and click it.

■ Figure 7-6 The MacTCP Control Panel

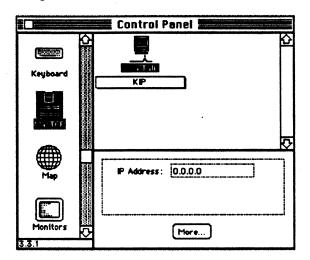


Figure 7-6 illustrates the MacTCP Control Panel. The upper part of the panel displays one or more icons labeled LocalTalk, EtherTalk, or Ethernet. These represent the cabling method employed in your network. Configuration of MacTCP depends on the type of cabling method in place. What icon or icons do you see?

- □ LocalTalk: Click the icon and go to "LocalTalk or EtherTalk cabling method"
- □ EtherTalk: Click the icon and go to "LocalTalk or EtherTalk cabling method"
- □ Ethernet: Click the icon and go to "Ethernet cabling method"
- One or more of these icons: Click one of the icons and go to the related

In the case of multiple icons, selecting an EtherTalk or Ethernet icon, if available, will produce faster data transmission.

LocalTalk or EtherTalk Cabling Method

If you selected the LocalTalk or EtherTalk icon from the MacTCP Control Panel (Figure 7-6), as described in step 4 above, follow the steps in this section to configure the MacTCP driver.

94

How to proceed Configuring MacTCP is a two-part process. The first part involves calling your network administrator to obtain the information specified in steps 1 through 3. Read these directions before calling the administrator so that you are familiar with the questions you have to ask. After obtaining the information required, you need only follow the instructions for filling in the Configuration Panel and you're finished.

Ask your network administrator the following questions:

1. Does the network use a domain name server?

Yes: Go to step 2

No: Ask for the location of the Hosts file. This file maps machine

names to internet addresses. Go to step 2.

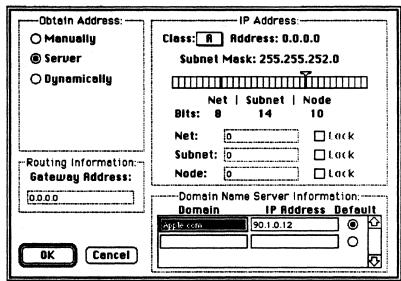
2. Is the network using a DDP-IP gateway, like Kinetics Fast Path?

If not, one must be installed for MacTCP to work. (DDP-IP stands for Datagram Delivery Protocol-Internet Protocol.)

- 3. Upon verifying that your network has a DDP-IP gateway, ask for the following information:

 - What domain am I in?
 A domain is a text version of the IP address.
 Record the name of the domain in the space provided. If you are in two domains, see the note below.
 - What is my IP address?
 An IP address is a binary number that identifies a host and the network on which it resides. It is inserted in data packets to specify their destination
 - ◆ Two domains: If you must enter two domain names, find out which domain is the default. Later, when filling out the Configuration Panel, you will click the default button for that domain.
- 4. Select the zone from the pop-up menu on the Control Panel. Figure 7-6 identifies the location of this menu.

- 5. Click the More button shown at the bottom of the Control Panel.
 - When the Configuration Panel appears, enter the domain information you obtained. Figure 7-7 illustrates the Configuration Panel.
- Figure 7-7 The MacTCP Configuration Panel



- Domain name, IP address, and default button Enter this information in the section labeled Domain Name Server Information and click the default button. If you enter two domain names, click the button for the domain your network administrator indicated is the default.
- 6. Click the Server button in the Obtain Address area of the Configuration Panel and click OK to return to the Control Panel. Close the Control Panel.
- 7. If your network has a domain name server, restart your Macintosh to activate the configuration changes and you're ready to go.

If your network does not have a domain name server, drag your network's Hosts file into your System Folder. Then restart your Macinstosh and your're ready to go.

Your network administrator will explain where to get the Hosts file. If no such file exists, you or your network administrator will have to create one. (This situation seldom occurs.) See Appendix A in the *MacTCP Administrator's Guide* for instructions.

Ethernet Cabling Method

If you selected the Ethernet icon from the MacTCP Control Panel (Figure 7-6), as described in step 4 above, follow the steps in this section to configure the MacTCP driver. The first part of this process involves calling your network administrator to obtain the information specified in steps 1 through 2 and in one of the protocol-specific subsections that follows. Read these directions before calling the administrator so that you are familiar with the questions you have to ask. After obtaining the information required, you need only follow the instructions for filling in the Configuration Panel and you're finished.

Ask your network administrator the following questions:

1. Does the network use a domain name server?

Yes: Go to step 2

No: Ask for the location of the Hosts file. This file maps machine

names to internet addresses. Go to step 2.

2. Which of the following protocols, if any, does the network use?

- □ RARP (Reverse Address Resolution Protocol)
- □ BootP (Bootstrap Protocol)
- □ RIP (Routing Information Protocol)

the following possibilities and go to the related section e next page:
all three: Go to "Entering configuration settings"
RARP and BootP: Go to "RARP and BootP setting"
RARP and RIP: Go to "RARP and RIP settings"
BootP and RIP: Go to "Entering configuration settings"
RARP only: Go to "RARP settings"
BootP only: Go to "BootP settings"
RIP only: Go to "RIP settings"
none of them: Go to "Settings when no protocols apply"
otP setting
ration information from your network administrator:
fault gateway address?ividual networks within a group of interconnected internet)
ering configuration settings" to put this information into a Panel.
e settings
ration information from your network administrator:
am I in?
n of the IP address. Ime of the domain in the space provided. If you are in two the note below.
address? ber that identifies a host and the network on which it nserted in data packets to specify their destination
s: If you must enter two domain names, find out which e default. Later, when filling out the Configuration Panel, you default button for that domain.

Chapter 7: Installation Apple Confidential 5/30/89

98

Now go to "Entering configuration settings" to put this information into the Configuration Panel.

RARP settings

Get this configuration information from your network administrator:
 What is the default gateway address? Identifies individual networks within a group of interconnected networks (an internet)
■ What is the address class? A code used to identify three classes of networks. Each class specifies a manimum number of hosts permitted
■ What is the subnet mask? Bit values set in the IP address to distinguish host addresses from subnetwork addresses. By using a subnet mask the network administrator can divide a network into smaller subnetworks.
Now go to "Entering configuration settings" to put this information into the Configuration Panel.
BootP settings
Get this configuration information from your network administrator:
 What is the default gateway address? Identifies individual networks within a group of interconnected networks (an internet)
Now go to "Entering configuration settings" to put this information into the Configuration Panel.

RIP settings

Get this configuration information from your network administrator:

	Do	oes the network use dynamic addressing?
	Ye	es: Ask for the following:
	0	What are the node range end points? Numbers demarcating the range of node numbers associated with an IP address. A node is a specialized computer that handles communications traffic in a network.
	No	Ask for the following:
	0	What is my IP address?
	0	What is the address class? A code used to identify three classes of networks. Each class specifies a manimum number of hosts permitted
	0	What is the subnet mask?
		go to "Entering configuration settings for networks using RIP or none protocols" to put this information into the Configuration Panel.
		ngs when no protocols apply
Ge	et th	nis configuration information from your network administrator:
	Do	oes the network use dynamic addressing?
	Ye	s: Ask for the following:
	0	What are the node range end points?

No	Ask for the following:
	What is the default gateway address? Identifies individual networks within a group of interconnected networks (an internet)
0	What is my IP address? A binary number that identifies a host and the network on which it resides. It is inserted in data packets to specify their destination
0	What is the address class? A code used to identify three classes of networks. Each class specifies a manimum number of hosts permitted
	What is the subnet mask? Bit values set in the IP address to distinguish host addresses from subnetwork addresses. By using a subnet mask the network administrator can divide a network into smaller subnetworks.

Now go to "Entering configuration settings for networks using RIP or none of the protocols" to put this information into the Configuration Panel.

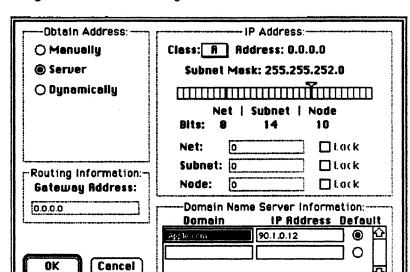
Entering configuration settings

After obtaining the configuration settings for your type of network, the next step is to complete the Configuration Panel depicted in Figure 7-7. If your network uses the BootP and RIP protocols or all three protocols(RARP, BootP, and RIP), skip step 2; otherwise, perform all the following steps:

Apple Confidential

Chapter 7: Installation

101



■ Figure 7-7 The MacTCP Configuration Panel

- 1. Click the More button shown at the bottom of Figure 7-6.
- 2. Find the item or items from the following list for which you obtained settings and follow the related instructions. (If your network uses RARP, BootP, and RIP or just BootP and RIP, skip to step 3.)
 - Default gateway address
 Enter this address in the section labeled Routing Information
 - Domain name, IP address, and default button Enter this information in the section labeled Domain Name Server Information and click the default button. If you enter two domain names, click the button for the domain your network administrator indicated is the default.
- 3. Click the Server button in the Obtain Address area of the Configuration Panel and click OK to return to the Control Panel (Figure 7-6). Close the Control Panel.
- 4. If your network has a domain name server, restart your Macinstosh to activate the configuration changes and you're ready to go.

If your network does not have a domain name server, drag your network's Hosts file into your System Folder. Then restart your Macintosh and you're finished.

Your network administrator will explain where to get the Hosts file. If no such file exists, you or your network administrator will have to create one. (This situation seldom occurs.) See Appendix A in the *MacTCP Administrator's Guide* for instructions.

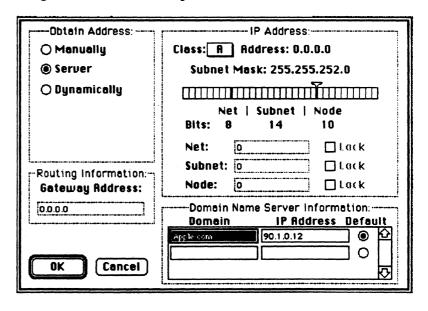
5. If you installed and configured MacTCP on a floppy disk, close the disk and eject it before going to the next section.

Now you're ready to install MacX, the last step in the installation process. See "Installing MacX" at the end of this chapter for instructions.

Entering configuration settings for networks using RIP or none of the protocols

After obtaining the configuration settings for your type of network, the next step is to complete the Configuration Panel depicted in Figure 7-7. Follow these steps:

■ Figure 7-7 The MacTCP Configuration Panel



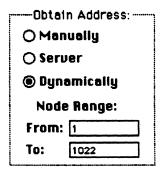
103

- 1. Click the More button shown at the bottom of the MacTCP Control Panel (Figure 7-6).
- 2. Select Manually in the Obtain Address area of the Configuration Panel illustrated in Figure 7-7.
- 3. Find the item or items from the following list for which you obtained settings and follow the related instructions.
 - Default gateway address
 Enter this address in the section labeled Routing Information
 - Domain name, IP address, and default button Enter this information in the section labeled Domain Name Server Information and click the default button. If you enter two domain names, click the button for the domain your network administrator indicated is the default.
 - Address class
 Select a class from the pop-up menu in the IP Address area.
 - Subnet mask
 Drag the slider along the address bar until the correct Subnet Mask
 number appears above it.
- 4. If your network is not using dynamic addressing, go to step 6.

If it is, select the button labeled Dynamically in the Obtain Address section.

Two node range fields will appear, as illustrated in Figure 7-8.

■ **Figure 7-8** Entering the node range endpoints



5. Enter the node range end-point numbers in the boxes labeled To and From.

Click OK to return to the Control Panel. Close the Control Panel and go to step 8.

- 6. Click OK to return to the Control Panel.
- 7. Enter the IP address for your Macintosh in the IP address field shown in Figure 7-9 and close the Control Panel.
- Figure 7-9 Entering the IP address



8. If your network has a domain name server, restart your Macinstosh to activate the configuration changes and you're ready to go.

If your network does not have a domain name server, drag your network's Hosts file into your System Folder. Then restart your Macintosh and you're finished.

Your network administrator will explain where to get the Hosts file. If no such file exists, you or your network administrator will have to create one. (This situation seldom occurs.) See Appendix A in the MacTCP Administrator's Guide for instructions.

9. If you installed and configured MacTCP on a floppy disk, close the disk and eject it before going to the next section.

Now you're ready to install MacX, the last step in the installation process. See "Installing MacX" at the end of this chapter for instructions.

Installing MacX

Before installing MacX, be sure to install the Communications Toolbox and a connection tool first. MacX will not start up without a functioning network connection. MacX will run on a Macintosh with a hard disk drive or one with just two floppy drives.

Installing MacX on a hard disk

Follow these instructions to install MacX on a Macintosh with a hard disk. If your Macintosh has one floppy disk drive, eject each disk after transferring its contents. After installing MacX, you'll be ready to start it up. Consult Chapter 2 for instructions on starting and running MacX.

1. Lock your three MacX disks before inserting them into your floppy disk drive.

Locking the disks ensures that you don't accidentally erase something while you are transferring MacX onto your hard disk. The disk is locked when you can see through the square hole on the top right side of the disk.

2. Insert the MacX Program disk in your floppy disk drive and double-click the disk icon to open the disk.

This disk contains the MacX application, a settings document called "Sample," and the MacX Colors file.



- Figure 7-10 Contents of the MacX Program disk
- 3. Drag the MacX application and the Sample document onto the directory window for your hard disk.

This folder is also called the Finder directory window. After the Finder has copied these files onto your hard disk you can move them anywhere on your desktop. Create a folder for them if you like.

4. Drag the MacX Colors file onto your directory window or into your System Folder.

If you put MacX Colors onto your desktop, place it in the same folder as your MacX application. If your MacX application isn't in a folder, create one and drag MacX and the MacX Fonts folder into it.

5. Create a new folder on your desktop labeled MacX Fonts or open your System Folder and create the MacX Fonts folder there.

If you create the folder on your desktop, place it in the same folder as your MacX application. If your MacX application isn't in a folder, create one and drag MacX and the MacX Fonts folder into it.

- 6. Insert one of the MacX Fonts disks into your floppy drive.

 If you have just one floppy drive, eject the program disk.
- 7. Select all of the fonts in the disk and drag them to the MacX Fonts folder.
- 8. Insert the other MacX Fonts disk into your floppy drive and repeat step 7.

That's all there is to it! Now you're ready to to start up MacX.

Preparing MacX to run on two floppy drives

MacX will run on a Macintosh that has two floppy disk drives instead of a hard disk drive. Essentially, you will have to put the System Folder on one disk and the MacX application on the other. After installing MacX, you'll be ready to start it up. Consult Chapter 2 for instructions on starting and running MacX.

1. Lock your three MacX disks before inserting them into your floppy disk drive.

Locking the disks ensures that you don't accidentally erase something while you are transferring files onto your working pair of floppy disks. The disk is locked when you can see through the square hole on the top right side of the disk.

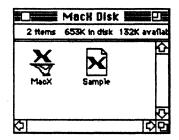
2. Insert the MacX Program disk into one floppy disk drive and a blank disk into the other drive. Double-click the program disk icon to open the disk.

This disk contains the MacX application, a settings document called "Sample," and the MacX Colors file.

- Figure 7-11 Contents of the MacX Program disk
- 3. Drag the MacX application and the Sample document onto the blank disk.

Close the blank disk and label it "MacX Disk" or something similar. This disk will be the one from which you run MacX. Now eject this disk or drag it to the trash and insert another blank disk.

■ Figure 7-12 Contents of the MacX Disk



4. Insert another disk containing a System Folder with version 6.0.2 or later of the Macintosh operating system. Label the disk "MacX System Disk" or something similar, if you have not already done so, and open the disk.

You should have created this disk earlier when you installed a connection tool.

5. Drag the MacX Colors file from the MacX Program disk into the System Folder on the system disk.

Close the MacX Program disk and drag it to the trash.

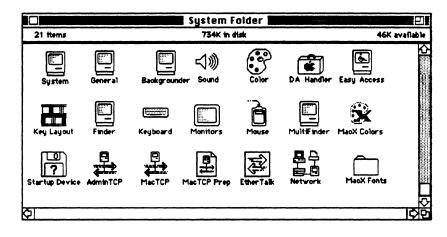
- 6. Create a new folder in your System Folder labeled "MacX Fonts."
- 7. Insert one of the MacX Fonts disks into your floppy drive and open the disk.
- 8. Find and select the font files labeled "cursor" and "fixed" and drag them to the MacX Fonts folder.

If *cursor* and *fixed* aren't on the fonts disk, eject it, insert the other MacX Fonts disk and locate these two fonts there. You can also drag other fonts into your MacX Fonts folder, but take care not to fill up the disk.

9. Close the MacX Fonts disk and drag it to the trash.

The System Folder on your MacX Systems Disk should look something like the one illustrated in Figure 7-13.

■ Figure 7-13 Sample System Folder



This System Folder shows icons for the MacTCP connection tool. You should have already installed a connection tool on your system disk by following instructions in the related installation section located after the section on installing the Comm Toolbox (in this case, "Installing MacTCP").

Once you have installed the Communications Toolbox, a connection tool, and MacX on your two floppy disks you'll be ready to to start up MacX. See Chapter 2, "Getting Started," for further instructions.

Malcolm Versus Other X Servers

THIS APPENDIX STATES THE DIFFERENCES BETWEEN MALCOLM AND CONVENTIONAL X servers.

Font path searching

In order to reduce the complexity of handling fonts and maintain a Macintosh-like user interface, Malcolm will not search through multiple font folders to locate fonts. It ignores font path search requests from clients. The only impact on the person using fonts in Malcolm is that all X Fonts must be kept in the MacX Fonts folder.

Typing international characters

Malcolm 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*.

Appendix E contains matrices indicating the hexadecimal equivalents for ISO Latin 1 and Macintosh Extended ASCII characters. Since these characters do not always have the same hexadecimal value when translated from one character set to the other, this appendix also contains a map of character-set translations.

Client incompatibilities

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



Reproducing uwm animation requests

Malcolm cannot accurately reproduce some animation requests made by the Universal Window Manager (uwm) because it uses an off-screen bit map and does not update the screen often enough to reproduce some effects. For example, when uwm wants a 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, Malcolm 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 release the middle mouse button (the Left Arrow key on a Macintosh keyboard), the window reappears, revealing its size. This problem occurs only with uwm in rooted mode. In rootless mode, the Malcolm Window Manager takes effect.

Printing images of invisible root windows

An XGetImage request is like a screen snapshot.

Clients, like xwd (the X Window Dumper), that request an image of an invisible root window using the XGetImage request will get a white pixmap. Essentially, this request produces a blank, white square because, in rootless mode, Malcolm cannot capture an image of a root window that isn't visible. However, if the XGetImage request specifies an image of client windows in the rootless environment, Malcolm can comply.

Listing fonts

Certain clients, such as *xlsfonts*, use the XListFonts or XListFontsWithInfo requests to list all fonts currently available. In Malcolm, 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.

The Malcolm Window Manager

HE MALCOLM WINDOW MANAGER CONTROLS THE APPEARANCE AND manipulation of windows in rootless mode, extending the Macintosh look and feel to the conventional X environment. Appendix B identifies its notable features.

Transferring text between clients

In conformance with the ICCCM, the Malcolm Window Manager enables you to cut, copy, and paste text from one client to another. To transfer text, use the client's method to perform the cut, copy, or paste operation. Malcolm is advised of a cut or copy event and places the selected text in the clipboard. When you indicate where you want to paste this text, by issuing the appropriate command within the client you want to receive it, Malcolm will transfer 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 Malcolm to pressing the middle mouse button.

As discussed in earlier chapters, the Malcolm Window Manager also enables you to cut, copy, and paste text from Macintosh applications to clients and to copy and paste text, such as font or color names, from Malcolm dialog windows (the Remote Command window, Font Director window, and Color Namer) to clients using subtantially the same operation. See Chapter 1 for a summary of this procedure.

Bitmap Distribution Format 2.1

HIS 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

Copyright © 1984, 1987, 1988 Adobe Systems, Inc.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

Bitmap Di	stribution	Format
-----------	------------	--------

	t	

Table of Contents

1.	Introduction	1
2.	File Format	1
3.	Metric Information	
4.	An Example File	

List of Figure	L	ist	of	Fi	gure	S
----------------	---	-----	----	----	------	---

Figure 3-1:	An example of a descender	3
Figure 3-2:	An example with the origin outside the bounding box	4
Figure 4-1:	A short example file	5

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.

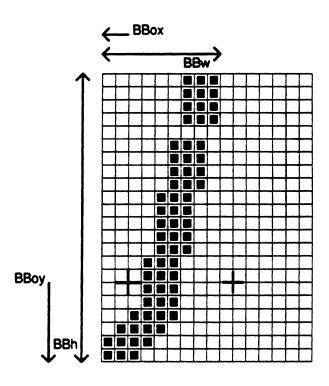
A font bitmap description file has the following general form, where each item is contained on a separate line of text in the file. Items on a line are separated by spaces.

- 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 the full name of the font. 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 a positive integer representing the Adobe Standard Encoding value. If the character is not a member of the Adobe Standard Encoding, ENCODING is followed by -1 and an optional integer specifying the glyph index; if the glyph index is not specified, a compiler will typically ignore the character segment.
- 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. 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.

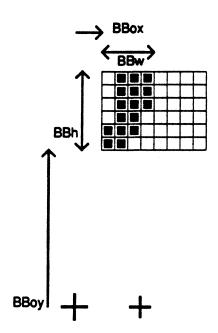
3. Metric Information

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



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

Figure 3-1: An example of a descender



BBh = 6, BBw = 4, BBox = +2, BBoy = +12 Rounded character width = 50

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

4. An Example File

Figure 4-1 is an abbreviated example of a bitmap file containing the specification of two characters (the j and quoteright in 3).1

```
STARTFORT 2.1
  COMMENT This is a sample font in 2.1 format.
  FORT Helvetica-Bold
  SIZE 8 200 200
  FONTBOUNDINGBOX 9 24 -2 -6
  STARTPROPERTIES 2
  MinSpace 4
  Copyright "Copyright (c) 1987 Adobe Systems, Inc."
  EMDPROPERTIES
  CHARS 2
. STARTCEAR 1
 ENCODING 106
  SWIDTE 355 0
  DWIDTH 8 0
  BBX 9 22 -2 -6
  BITMAP
  0380
  0380
  0380
  0380
  0000
  0700
  0700
  0700
  0700
  OEOO
  OEOO
  OEOO
  OEOO
  OEOO
  1000
  1000
  1000
  1C00
  2C00
  7800
 FOOO
 EOOO
 ENDCHAR
 STARTCHAR quoteright
 ENCODING 39
 SWIDTH 223 0
 DWIDTH 5 0
 BBX 4 5 2 12
 ATTRIBUTES 01C0
 BITTOP
 70
 70
 60
 EO
 CO
 ENDCEAR
 ENDFONT
```

Figure 4-1: A short example file

¹Helvetica® is a registered trademark of Allied Corporation.

X Logical Font Description Conventions

HIS 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 X Windows System, Version 11 Draft X Standard V1.2 for public review

Abstract

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.

Conventions are established which identify fonts via a standard parsable font name and optional font properties in a server-independent way, thus providing X clients a mechanism to query and understand fonts available on foreign servers.

Copyright © 1988, 1989 by Digital Equipment Corporation, Maynard MA.

Permission to use, copy, modify, and distribute this documentation for any purpose and without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. MIT and Digital Equipment Corporation make no representations about the suitability for any purpose of the information in this document. This documentation is provided as is without express or implied warranty. This document is only a draft standard of the MIT X Consortium and is therefore subject to change.

Digital Equipment Corporation

[†] The X Window System is a registered trademark of the Massachusetts Institute of Technology.

Contents

1	Introduc	ction		 	 	
2			and goals			
_	2.1		e, descriptive font names			
	2.2	-	ort multiple vendors and character sets			
	2.3		r, operating and file system independent			
	2.4		orts arbitrarily complex font matching/substitution			
	2.5		sible			
3			Description			
•	3.1		ame			
	3.1		FontName Syntax			
	3.1		FontName Field Definitions			
		2.1	FOUNDRY x-registered-name			
		2.2	FAMILY_NAME string			
		2.3	WEIGHT_NAME string			
		.2.4	SLANT code-string			
		2.5	SETWIDTH_NAME string			
		2.6	ADD_STYLE_NAME string			
		2.7	PIXEL_SIZE integer-string			
		2.8	POINT_SIZE integer-string			
		2.9	RESOLUTION_X integer-string			
	• • •	2.3				
		2.10	RESOLUTION_Y integer-string			
		2.11 2.12	SPACING code-string			
			AVERAGE_WIDTH integer-string			
		.2.13	CHARSET_REGISTRY x-registered-name			
		.2.14	CHARSET_ENCODING registered-name			
	3.1		Examples			
	3.2		roperties			
	3.2	-	FOUNDRY ATOM			
	3.2		FAMILY_NAME ATOM			
	3.2		WEIGHT_NAME ATOM			
	3.2		SLANT ATOM			
	3.2	.5	SETWIDTH_NAME ATOM	 	 	. 12

	3.2.6	ADD_STYLE_NAME ATOM	
	3.2.7	PIXEL_SIZE CARD32	
	3.2.8	POINT_SIZE CARD32	
	3.2.9	RESOLUTION_X CARD32	
	3.2.10	RESOLUTION_Y CARD32	
	3.2.11	SPACING ATOM	
	3.2.12	AVERAGE_WIDTH CARD32	
	3.2.13	CHARSET_REGISTRY ATOM	
	3.2.14	CHARSET_ENCODING ATOM	
	3.2.15	MIN_SPACE CARD32	
	3.2.16	NORM_SPACE CARD32	
	3.2.17	MAX_SPACE CARD32	
	3.2.18	END_SPACE CARD32	
	3.2.19	AVG_CAPITAL_WIDTH INT32	
	3.2.20	AVG_LOWERCASE_WIDTH INT32	
	3.2.21	QUAD_WIDTH INT32 (DEPRECATED)	
	3.2.22	FIGURE_WIDTH INT32	
	3.2.23	SUPERSCRIPT_X INT32	
	3.2.24	SUPERSCRIPT_Y INT32	
	3.2.25	SUBSCRIPT_X INT32	
	3.2.26	SUBSCRIPT_Y INT32	
	3.2.27	SUPERSCRIPT_SIZE CARD32	
	3.2.28	SUBSCRIPT_SIZE CARD32	
	3.2.29	SMALL_CAP_SIZE CARD32	
	3.2.30	UNDERLINE_POSITION INT32	
	3.2.31	UNDERLINE_THICKNESS CARD32	
	3.2.32	STRIKEOUT_ASCENT INT32	
	3.2.33	STRIKEOUT_DESCENT INT32	
	3.2.34	ITALIC_ANGLE INT32	
	3.2.35	CAP_HEIGHT CARD32	
	3.2.36	X_HEIGHT CARD32	
	3.2.37	RELATIVE_SETWIDTH CARD32	19
	3.2.38	RELATIVE_WEIGHT CARD32	20
	3.2.39	WEIGHT CARD32	21
	3.2.40	RESOLUTION CARD32 (DEPRECATED)	21
	3.2.41	FACE_NAME ATOM	21
	3.2.42	COPYRIGHT ATOM	22
	3.2.4 3	NOTICE ATOM	22
	3.2.44	DESTINATION CARD32	22
3.3	Built-ii	n Font Property atoms	22
Affe	cted elemen	ts of XLIB and the X protocol	24
BDF	Conformar	nce	24

	5.1	XLFD Conformance Requirements
	5.2	FONT_ASCENT, FONT_DESCENT and DEFAULT_CHAR24
	8	5.2.1 FONT_ASCENT <i>INT32</i>
	5	5.2.2 FONT_DESCENT <i>INT</i> 32
	5	5.2.3 DEFAULT_CHAR CARD32
6	Typefa	face Copyrights
Tab	loo	
<u> </u>	162	
	1	FontName Syntax
	2	SLANT encodings
	3	SPACING encodings
	4	Examples of X Font Names
	5	X FontProperties Syntax
	6	RELATIVE_SETWIDTH encodings
	7	RELATIVE_WEIGHT encodings
	8	DESTINATION encodings
	9	Built-in Font Property Atoms

·

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.

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 & encodings;
- is X server & operating/file system independent;
- provides adequate descriptive font information for arbitrarily complex font matching/substitution;
- is extensible.

[†] See Typeface Copyrights section for copyright information

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 vendors and character sets

The font name and properties should distinguish between fonts that were supplied by different font vendors, but possibly sharing 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. Certain 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 (*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 FontProperty names and their associated data values may also appear as items of the STARTPROPERTIES ... ENDPROPERTIES list in the BDF 2.1 specification.

3.1 FontName

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

All font names conforming to this specification will 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 supporting 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 STRINGS), whose Backus-Naur Form syntax description is as follows:

Table 1: FontName Syntax

FontName ::=	XFontNameRegistry XFontNameSuffix PrivFontNameRegistry PrivFontNameSuffix
XFontNameRegistry ::=	Delim 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 defining an extension to font name syntax, e.g. "1.3"
XFNExtPrefix ::=	OCTET-the value of ISO8859-1 PLUS (Column/Row 02/13)

Table 1 (Cont.): FontName Syntax

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 a 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 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 upon 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 & WEIGHT) which 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:

Table 2: SLANT encodings

Code	English String	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

[†] See Typeface Copyrights section for copyright information

Table 2 (Cont.): SLANT encodings

Code	English String	Description
"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 upon 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 bedy 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 0 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 0 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 & 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 (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).

Table 3: SPACING encodings

Code	English String	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"	Monospace	 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;

Table 3 (Cont.): SPACING encodings

Code	English String	Description
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 "ink" outside of the 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 & 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 répertoire), as maintained by the X Consortium character set registry. CHARSET_REGISTRY is a 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 a 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.

The X Consortium shall define procedures for registering character set names and shall maintain and publish in a timely manner a registry of such registered 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.

Table 4: Examples of X Font Names

Font	X FontName
Courier 8pt @75dpi	-Adobe-Courier-Medium-R-Normal8-80-75-75-M-50-ISO8859-1
Courier 10pt	-Adobe-Courier-Medium-R-Normal10-100-75-75-M-60-ISO8859-1
Courier 12pt	-Adobe-Courier-Medium-R-Normal12-120-75-75-M-70-ISO8859-1
Courier 14pt	-Adobe-Courier-Medium-R-Normal14-140-75-75-M-90-ISO8859-1
Courier 18pt	-Adobe-Courier-Medium-R-Normal18-180-75-75-M-110-ISO8859-1
Courier 24pt	-Adobe-Courier-Medium-R-Normal24-240-75-75-M-150-ISO8859-1
Courier Bold 10pt	-Adobe-Courier-Bold-R-Normal10-100-75-75-M-60-ISO8859-1
Courier BoldOblique 10pt	-Adobe-Courier-Bold-O-Normal10-100-75-75-M-60-ISO8859-1
Courier Oblique 10pt	-Adobe-Courier-Medium-O-Normal10-100-75-75-M-60-ISO8859-1
Times Roman 10pt	-Adobe-Times-Medium-R-Normal14-100-100-P-74-ISO8859-1
Times Bold 10pt	-Adobe-Times-Bold-R-Normal14-100-100-P-76-ISO8859-1
Times BoldItalic 10pt	-Adobe-Times-Bold-I-Normal14-100-100-P-77-ISO8859-1
Times Italic 10pt	-Adobe-Times-Medium-I-Normal14-100-100-P-73-ISO8859-1
Charter 12pt @75dpi	-Bitstream-Charter-Medium-R-Normal12-120-75-75-P-68-ISO8859-1
Charter BoldItalic 12pt	-Bitstream-Charter-Bold-I-Normal12-120-75-75-P-75-ISO8859-1
Charter Bold 12pt	-Bitstream-Charter-Bold-R-Normal12-120-75-75-P-76-ISO8859-1
Charter Italic 12pt	-Bitstream-Charter-Medium-I-Normal12-120-75-75-P-66-ISO8859-1
SYMBOL 8pt @100dpi	-Adobe-Symbol-Medium-R-Normal11-80-100-100-P-61-Adobe-FONTSPECIFIC
Symbol 10pt	-Adobe-Symbol-Medium-R-Normal14-100-100-P-85-Adobe-FONTSPECIFIC
Symbol 12pt	-Adobe-Symbol-Medium-R-Normal17-120-100-100-P-95-Adobe-FONTSPECIFIC

Table 4 (Cont.): Examples of X Font Names

Font	X FontName
Symbol 14pt	-Adobe-Symbol-Medium-R-Normal20-140-100-100-P-107-Adobe-FONTSPECIFIC
Symbol 18pt	-Adobe-Symbol-Medium-R-Normal25-180-100-100-P-142-Adobe-FONTSPECIFIC
Symbol 24pt	-Adobe-Symbol-Medium-R-Normal34-240-100-100-P-191-Adobe-FONTSPECIFIC

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 given below:

Table 5: X FontProperties Syntax

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

Table 5 (Cont.): X FontProperties Syntax

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

where "/" denotes real division, "*" denotes real multiplication and "CEIL" denotes a function that rounds its real argument up 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 = CEIL ((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 = CEIL (POINT_SIZE / 10)
ENspace = CEIL (POINT_SIZE / 20)
THINspace = CEIL (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_X 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 algorithm:

```
if (AVERAGE_WIDTH undefined) then
  AVERAGE_WIDTH = CEIL (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 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 = CEIL(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 algorithm:

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 algorithm:

```
if (MAX_SPACE undefined) then
  MAX_SPACE = CEIL(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 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 algorithm:

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

where 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/substitution.

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

```
if (AVG_LOWERCASE_WIDTH undefined) then
  if (lower case exists) then
    AVG_LOWERCASE_WIDTH = CEIL (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 & 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 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 algorithm:

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

where 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].

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

```
if (SUPERSCRIPT_Y undefined) then
  SUPERSCRIPT Y = CEIL(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 algorithm:

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

```
if (SUBSCRIPT_Y undefined) then
SUBSCRIPT_Y = CEIL(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 & SUPERSCRIPT_Y.

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

```
if (SUPERSCRIPT_SIZE undefined) then
  SUPERSCRIPT_SIZE = CEIL(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 & 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 = CEIL(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 algorithm:

```
if (SMALL_CAP_SIZE undefined) then
   SMALL_CAP_SIZE = CEIL(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], then 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 algorithm:

```
if (UNDERLINE_POSITION undefined) then
   UNDERLINE_POSITION = CEIL(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 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, then the upper-left corner of the strikeout box is at [X, Y - STRIKEOUT_ASCENT] and the lower-right corner 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 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, then the upper-left corner of the strikeout box is at [X, Y - STRIKEOUT_ASCENT] and the lower-right corner 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 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 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, 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 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, 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 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:

Table 6: RELATIVE_SETWIDTH encodings

Code	English String	Description
0	undefined	– undefined or unknown
10	UltraCondensed	- lowest ratio of average width to height

Table 6 (Cont.): RELATIVE_SETWIDTH encodings

Code	English String	Description	
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 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)
```

where 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:

Table 7: RELATIVE_WEIGHT encodings

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,	

Table 7 (Cont.): RELATIVE_WEIGHT encodings

Code	English String	Description
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 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 0 is the "lightest" weight.

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

```
CapStemWidth = average width of the stems of capitals
if (WEIGHT undefined) then
  WEIGHT = CEIL ((CapStemWidth * 1000) / CAP HEIGHT)
```

A calculated value for weight is necessary when matching fonts from different families since 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 accommodate displays with non-square aspect ratios, and are given by the RESOLUTION_X and RESOLUTION_Y font name fields/properties. The units of the original defintion 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, 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, 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 which 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:

Table 8: DESTINATION encodings

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:

Table 9: Built-in Font Property Atoms

Font Property	Atom Name	Property Type
MIN_SPACE	MIN_SPACE_ATOM	CARD32
NORM_SPACE	NORM_SPACE_ATOM	CARD32
MAX_SPACE	MAX_SPACE_ATOM	CARD32
END_SPACE	END_SPACE_ATOM	CARD32
SUPERSCRIPT_X	SUPERSCRIPT_X_ATOM	INT32
SUPERSCRIPT_Y	SUPERSCRIPT_Y_ATOM	INT32
SUBSCRIPT_X	SUBSCRIPT_X_ATOM	INT32
SUBSCRIPT_Y	SUBSCRIPT_Y_ATOM	INT32
UNDERLINE_ POSITION	UNDERLINE_POSITION_ ATOM	INT32
UNDERLINE_ THICKNESS	UNDERLINE_THICKNESS_ ATOM	CARD32
STRIKEOUT_ASCENT	STRIKEOUT_ASCENT_ATOM	INT32
STRIKEOUT_ DESCENT	STRIKEOUT_DESCENT_ATOM	INT32
FONT_ASCENT	FONT_ASCENT_ATOM	INT32
FONT_DESCENT	FONT_DESCENT_ATOM	INT32
ITALIC_ANGLE	ITALIC_ANGLE_ATOM	INT32
X_HEIGHT	X_HEIGHT_ATOM	INT32
QUAD_WIDTH	QUAD_WIDTH_ATOM	INT32 —deprecated
WEIGHT	WEIGHT_ATOM	CARD32
POINT_SIZE	POINT_SIZE_ATOM	CARD32
RESOLUTION	RESOLUTION_ATOM	CARD32 —deprecated
COPYRIGHT	COPYRIGHT_ATOM	ATOM
FACE_NAME	FACE_NAME_ATOM	ATOM
FAMILY_NAME	FAMILY_NAME_ATOM	ATOM
DEFAULT_CHAR	DEFAULT_CHAR_ATOM	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 V1.2 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 FontProperty definitions.

A simple method of testing for conformance would entail first verifying that the Font-NameRegistry 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 povided 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 follow 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+1.

FONT_ASCENT can be approximated if not provided as a font property, according to the 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 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 it 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.

6 Typeface Copyrights

Helvetica, Times are registered trademarks of Linotype Company.

ITC Avant Garde Gothic is a registered trademark of International Typeface Corporation.

Times Roman is a registered trademark of Monotype Corporation.

Bitstream Amerigo is a registered trademark of Bitstream Inc.

Stone is a registered trademark of Adobe Systems Inc.

[end]

Character Conversion Charts

Since X WINDOW SYSTEM AND MACINTOSH APPLICATIONS USE DIFFERENT character sets, Malcolm must translate any data transmitted between...

(To be supplied later.)

