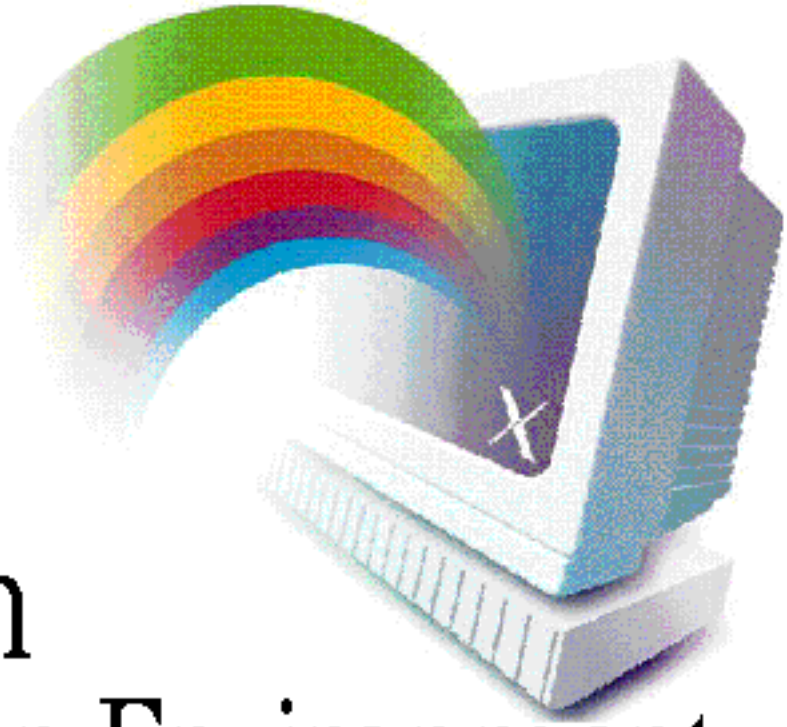


Macintosh Application Environment

Administrator’s Manual

Apple Computer, Inc.

This manual is copyrighted by Apple or by Apple's suppliers, with all rights reserved. Under the copyright laws, this manual may not be copied, in whole or in part, without the written consent of Apple Computer, Inc. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased may be sold, given, or lent to another person. Under the law, copying includes translating into another language.

Portions of this manual are derived from *FLEXlm End User's Guide*, copyrighted by Globetrotter Software, Inc., and modified by Apple. Globetrotter assumes no responsibility with regard to the contents of this manual.

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., 1996. All rights reserved.

1 Infinite Loop
Cupertino, CA 95014-2084
408-996-1010
http://www.apple.com

Apple, the Apple logo, AppleScript, AppleShare, AppleTalk, Chicago, EtherTalk, HyperCard, LaserWriter, LocalTalk, Mac, Macintosh, MacTCP, New York, OpenDoc, and QuickTime are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Balloon Help, Extensions Manager, Finder, Geneva, Monaco, QuickDraw, and Sound Manager are trademarks of Apple Computer, Inc.

Adobe, Acrobat, and PostScript are trademarks of Adobe Systems Incorporated or its subsidiaries and may be registered in certain juristdictions.

Claris, ClarisDraw, ClarisImpact, ClarisWorks, and Filemaker are registered trademarks of Claris Corporation.

Helvetica, Times, and Palatino are registered trademarks of Linotype-Hell AG and/or its subsidiaries.

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

X Window System is a trademark of the Massachusetts Institute of Technology.

Simultaneously published in the United States and Canada.

Mention of third-party products is for informational purposes only and constitues neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the performance of these products.

Contents

About This Manual / 11

What you need to know / 11

New features in MAE version 3.0 / 12

What's covered in this manual / 13

Terminology used in this manual / 14

Documentation, online help, and other information / 16

1 Installation From a Server / 19

Setting up MAE on a server for remote installation / 19

- Making the MAE CD-ROM disc accessible directly / 20
- Making the MAE files accessible from a server's hard disk / 21
- Installing and starting the license manager software / 22
- Installing MAE on a server and sharing through NFS / 22

2 Customizing MAE and Your Workstation / 23

Customizing MAE with command options and X resources / 24

- Command options / 25
- X resources / 26
  - -appledir / 26
  - -filedouble / 26
  - -fstimeout / 26
  - -geometry / 26
  - -iconic / 26
  - -installcmap / 26
  - -macdir / 27
  - -maxdepth / 27
  - -nosound / 27
  - -nocursorscale / 27
  - -pixelscale / 27
  - -title / 27
  - -sysfol / 27
- The loading order for X resources / 28

Configuring NFS mounts / 29
Installing kernel modules / 30
Providing a customized command line / 30
Customizing the Easy Install procedure / 30
Installing applications using a temporary installer volume (TIV) / 32
Creating and using a TIV / 32
Making a TIV persistent / 33
Customizing System Folders / 34
Distributing common extensions, control panels, / 35
Using global fonts to minimize disk usage / 35
Improving MAE performance / 36
Applications and processor performance / 38
Improving workstation performance when not using MAE / 38
Configuring MAE to create aliases that appear at startup / 38
MAE independent volumes / 40
Creating independent volumes / 41
Independent volumes you define / 41
Independent volumes for UNIX mount points / 42
Independent volumes created by MAE to access system resources / 42
Defining independent volumes / 42
Unmounting independent volumes / 43
Renaming independent volumes / 43
Symbolic links with MAE / 43
The Macintosh Desktop database / 44
Using APPLEPATH to simplify launching / 45
Locating applications for optimal performance / 46
AppleSingle and AppleDouble file formats / 46
Troubleshooting / 47
Insufficient memory on the SparcStation / 47

3 Structure and Function of MAE / 49

The contents of the apple directory / 50
The apple/Apple Extras directory / 50
The apple/bin directory / 51
The apple/lib directory / 52
The apple/license.dat file / 53
The apple/man directory / 53
The apple/shared directory / 53
The apple/sys directory / 53

The contents of the .mac directory / 54
Optional directory to specify a default .mac directory / 54
How MAE works / 55
Graphics in MAE / 56
Color flash / 57

4 Resources Required by MAE / 59

Understanding memory usage with MAE / 60
Native code and data / 60
apple/lib/engine: the 680x0 scaffolding and / 60
Mac OS system and application code / 61
Graphics buffer / 61
Swap space / 63
Physical memory / 63
UNIX I/O caches / 64
CPU caches / 64
Terminology used for caches / 65
How MAE uses caches / 67
Components of MAE / 69
The .mac directory / 71
Other overhead considerations / 71
Modifying system resources / 72
Installing applications / 72
Opening folders with the Finder / 72

5 AppleTalk Administration / 73

Overview / 74
Implementation / 74
Getting started with AppleTalk for MAE / 75
Hardware requirements / 75
Software requirements / 76
Installing AppleTalk for MAE / 76
What the MAE Installer does on HP-UX 10.10 / 76
What the MAE Installer does on HP-UX 9.0x / 77
What the MAE Installer does on Solaris / 79
Verifying the presence of streams / 79
Removing AppleTalk from the kernel / 80

Using AppleTalk for MAE / 80
Initializing the AppleTalk kernel stack / 80
Turning AppleTalk on or off / 80
Accessing named resources / 81
Selecting a network interface with the Network control panel / 81
Personal file sharing / 82
Controlling access to shared folders / 82
Using the `/etc/mae_fileshare` file to limit access / 83
Limiting the owner's access / 83
Determining permissions / 84
Sharing UNIX directories / 85
Turning off guest access / 85
Sharing folders among several users / 87
Monitoring the use of file sharing / 87
Running several MAE sessions simultaneously / 87
Extensions / 88
UNIX AppleTalk commands / 89
Using `appletalk -s` to determine the status of AppleTalk / 89
Using `appletalk -D` to turn off AppleTalk / 89
Using `atlookup` to list the NVEs in a zone / 90
Using `appleping` to ensure that a node is alive / 90
Troubleshooting / 91
No AppleTalk Zone list in the Chooser / 91
Selecting AppleTalk Zones / 91
Using network diagnostic tools / 91
Reinstalling AppleTalk for MAE / 91
Restoring network extensions / 91
An AppleTalk router comes up / 92
(Sun only) File modes for non `-le0` interfaces / 92
(Sun only) Unloading AppleTalk stream modules on Solaris systems / 92
(HP only) HP-UX 9.0.1 and STREAMS/UX / 92
(HP only) Booting HP-UX from the saved kernel / 93
References / 94

6 The Sound Manager / 95

Using the MAE Sound Manager / 96

Hewlett-Packard workstations / 96

Troubleshooting for HP workstations / 97

Sun workstations / 98

Troubleshooting for Sun workstations / 98

7 General Troubleshooting / 99

Desktop / 100

Can’t empty Trash / 100

You cannot move “file name” to the Trash, because a disk error occurred / 100

Generic icons appear for files of installed applications / 100

Display / 100

Adjusting mouse movements / 100

Full-screen MAE displays / 101

Resizing the Finder and application windows / 101

Flashing menu bar after exporting to X / 101

Cursor disappears / 101

Screen fades to black / 101

Changing the X modifier map / 102

Modifier map sharing between MAE and X applications / 102

MAE independent volumes / 103

Problems with icons and windows with newly mounted MIV volumes / 103

Exceeded maximum number of mounted volumes (MIVs) / 103

“The disk name cannot be used, because the folder cannot be found” message / 103

“One or more items could not be shared” error message / 104

File systems / 104

“Too many open files” error / 104

Network file system performance / 104

Accessing files and directories with a space / 105

Permission problems / 105

Using automounter (HP only) / 105

Sharing `apple` directories across systems / 106

Optional directory to specify a default .mac directory / 106

Using Fetch / 106

Problems with aliases / 106

Aliases not created during initial startup / 106

Using CellNames with MAE / 107

MAE hangs after an NFS server shuts down / 107

Space for `/users` on HP Series 700 workstations / 107

Installing applications / 108
Installation of incompatible system extensions / 109
Installing PPDs / 109
Keyboard / 109
Shift-Option key sequence on IBM keyboards / 109
Performance / 110
Starting up MAE takes a long time / 110

Appendix The License Manager / 111

License manager quick start / 112
Overview / 112
License management components / 112
The AppleMAE Daemon / 113
The License Manager Daemon (`lmgrd`) / 113
The License File / 113
The license request process / 113
Configuring license management / 114
Setting up license management / 114
Selecting license server nodes / 115
Resources required by the server / 115
Processes / 115
Sockets / 116
CPU time / 116
Disk space / 116
Memory / 117
Network bandwidth / 117
Remote mounted disks and diskless nodes / 117
Multiple servers / 118
Determining license server hostids / 119
Obtaining license files / 120
Determining if you need an options file / 120
Determining where to put license management components / 120
The license manager daemon, `lmgrd` / 120
The vendor daemon, `AppleMAE` / 120
The license file, `license.dat` / 121
License management utilities / 122
The log file / 122
Starting the license manager, `lmgrd` / 123

The format of the license file / 123
SERVER lines / 124
DAEMON lines / 125
FEATURE or INCREMENT lines / 126
FEATURESET lines / 129
PACKAGE lines / 130
UPGRADE lines / 132
A sample license file / 132
Limits on modifiable parameters / 132
Types of licenses / 133
Floating (concurrent) licenses / 133
Node-locked licenses / 134
Floating licenses with limited availability / 134
Mixed node-locked and floating licenses / 135
The license finder / 135
Starting the license finder / 136
Finder file format / 136
Preventing conflicts with other licensed applications / 136
Combining license files / 137
Using separate license files on the same server node / 137
Using a different server node / 138
The options file / 139
The EXCLUDE option / 140
The EXCLUDEALL option / 140
The GROUP option / 140
The HOST_GROUP option / 141
The INCLUDE option / 141
The INCLUDEALL option / 141
The LINGER option / 142
The NOLOG option / 142
The REPORTLOG option / 142
The RESERVE option / 143
The TIMEOUT option / 143
Rules of precedence in options files / 144
Simple options file example / 144
Limiting access for multiple users / 145
EXCLUDE example / 146
INCLUDE example / 146

The daemon log file / 147
Informational messages / 147
Configuration problem messages / 149
Daemon software error messages / 151
Troubleshooting license problems / 153
General hints / 153
Host ID problems / 154
Connection problems / 154
Other client problems / 156
Other server problems / 156

Index / 159

About This Manual

The *Macintosh Application Environment Administrator's Manual* describes the structure and function of the Macintosh Application Environment (MAE) and offers information on how to administer the system. This manual supplements the *Macintosh Application Environment User's Manual* (referred to here as the *MAE User's Manual*). If you need information about getting started and basic MAE functions, consult the *MAE User's Manual*.

This manual is not meant to replace your UNIX® administration documentation. It deals only with information that is specific to MAE.

What you need to know

This manual assumes that you can perform all appropriate UNIX system administrative tasks and that you are familiar with the Macintosh user interface.

New features in MAE version 3.0

MAE version 3.0 contains many new features that considerably expand its usefulness:

- Improved performance for remote X display servers
- Improved Find File utility
- Support for multiple TCP/IP interfaces
- System 7.5.3 support
- Enhanced installer
- Floating licenses
- Additional CD-ROM formats
- X Window pixel scaling
- Independent volume support
- Smaller system folders
- Faster floating-point operations
- More simultaneously open files
- Improved keyboard support
- Improved file sharing security

Implementation details of concern to you as the system administrator are described in this manual. How to use these features is described in detail in the *MAE User's Manual*.

What's covered in this manual

This manual contains the following chapters:

- Chapter 1, "Installation From a Server," describes how to set up MAE on a server for remote installation.
- Chapter 2, "Customizing MAE and Your Workstation," teaches you how to create aliases that appear at startup, use X resources, boot HP-UX from a saved kernel, mount NFS and AFS, manage the Temporary Installer Volume for installing application software, customize System Folders, improve performance, and troubleshoot startup problems.
- Chapter 3, "Structure and Function of MAE," describes the contents of the `apple` and `.mac` directories, the structure and function of MAE, and MAE's use of graphics.
- Chapter 4, "Resources Required by MAE," details MAE's use of I/O caches, CPU caches, swap space, physical memory, and disk space.
- Chapter 5, "AppleTalk Administration," discusses hardware and software requirements for AppleTalk as well as how to install, remove, and use AppleTalk. This chapter also describes Personal File Sharing, access permissions, and security issues.
- Chapter 6, "The Sound Manager," teaches you how to configure and use the Sound Manager and troubleshoot MAE's sound production.
- Chapter 7, "General Troubleshooting," provides help for common problems.
- Appendix, "The License Manager," describes MAE's license management.

Terminology used in this manual

This section explains the terminology used in this manual to refer to modifier keys and to various mouse and keyboard operations.

Modifier keys

This manual refers often to several modifier keys that appear on standard Apple keyboards: Command (⌘), Option, Control, and Caps Lock. Since most workstation keyboards do not come with these modifier keys, MAE assigns the Macintosh modifier keys to keys on your workstation keyboard. MAE also assigns two commonly used X Window System™ modifier keys, Alt and Mode Switch, to keys on your keyboard. When this manual mentions the name of a Macintosh modifier key, such as Option, the manual is actually referring to the key on your keyboard to which that particular modifier function has been mapped.

To check your modifier key settings, open the MAE Keyboard control panel. (Choose Control Panels from the Apple (🍎) menu and double-click the icon for the MAE Keyboard control panel.) To learn how to reassign modifier keys, see “Reassigning Modifier Keys” in Chapter 9 of the *MAE User’s Manual*.

Mouse buttons

This book often refers to *the* mouse button, because the Macintosh operating system (Mac OS) was designed to be used with a single-button mouse. The phrase “the mouse button” refers to the first button (often called “the leftmost button” or “Mouse Button 1”) on a mouse or trackball with multiple buttons. For example, if a step instructs you to “press the mouse button,” you would press the first button on a three-button mouse.

You can use X window features to change the mapping of the mouse buttons.

Basic mouse and keyboard operations

In this manual, some terms represent a series of actions that you perform with the mouse, keyboard, or both. For example, the word “enter” indicates that you type a character (or a series of characters), then press the Return key.

Here is a list of standard terms for mouse and keyboard operations and the meaning of each term.

Term	Action
Click	Press and then immediately release the mouse button.
Double-click	Press and release the mouse button twice in quick succession without moving the mouse.
Drag	Position the pointer on an item on the screen, such as an icon that you want to move. Press and hold down the mouse button while moving the mouse until you have accomplished the desired task, such as moving the icon to a new location, and then release the mouse button.
Choose	Activate a command in a menu by positioning the pointer on the menu title and dragging (usually downward) through the menu until the command you want is highlighted. Then release the mouse button.
Select	Designate where on the screen the next action will take place. To select an object such as an icon, position the pointer on the object and click the mouse button once. When selected, the object appears highlighted. To select text, drag across the text until the entire range of text that you want to select is highlighted.
Type	Type a series of characters *without* pressing the Return key.
Press	Press a single key or key combination without pressing the Return key afterward.
Enter	Type a series of characters and then press the Return key.

Documentation, online help, and other information

You can find helpful information about MAE in a number of different sources. In addition to the online documents such as this one, MAE includes online help in the form of Balloon Help, Apple Guide, and Toolbar help. Late-breaking MAE information is provided in the Read Me file. You can find updated information on the MAE Web page.

To use an online document, first install the Acrobat™ Reader software by double-clicking the Adobe™ Acrobat Reader v2.1 icon located in the doc/Adobe Acrobat Reader folder. Then double-click the PDF file document icon to open the document you want to read. For details on using the online documentation, see the Read Me file to get started; for more information, consult the Acrobat Reader online documentation.

In addition to this manual, the *Macintosh Application Environment User's Manual* is shipped with MAE. This is an online manual that contains information needed to use MAE, including starting and quitting, setting up and using printers, CD-ROM discs, floppy disks, installing and using Macintosh applications, managing UNIX files with MAE, networking with MAE, and customization.

The Read Me file (`README.TXT`)

MAE also includes an online Read Me file, which provides late-breaking information about MAE and troubleshooting information about installing and using MAE on your workstation. The Read Me file also explains how to get MAE information that Apple provides on the World Wide Web.

The Read Me file (named `README.TXT`) is located in the root directory of the *Macintosh Application Environment* CD-ROM disc. You can view the Read Me file by using the SimpleText application or a standard UNIX tool, such as `more`(1) or `vi`(1).

Balloon Help

MAE comes with a feature called Balloon Help, which provides immediate onscreen help when you need it. If you want to know more about a window, button, menu, command, or icon in the MAE window, use Balloon Help to get the information you need.

Many Macintosh programs also have Balloon Help.

To use Balloon Help, click on the question mark at the right end of the menu bar and choose the Show Balloons command. Now, balloons containing explanatory text appear next to objects that you point to on the screen. To turn off Balloon Help, choose Hide Balloons from the question mark menu.

MAE Toolbar help

MAE provides immediate onscreen help for the MAE Toolbar when you need it. If you want to know how to use a feature of the Toolbar, click the Get Toolbar Help button at the right end of the Toolbar.

MAE information on the Web

Apple maintains a World Wide Web page to provide you with updated information about MAE. You can also find current versions of the MAE manuals on the MAE Web page. You can reach the MAE Web page at http://www.mae.apple.com.

MAE Guide

MAE Guide provides immediate onscreen Apple Guide help for MAE when you need it. If you want to know how to use a feature of the Toolbar, open the Help menu by clicking the question mark button at the right end of the menu bar and selecting MAE Guide.

1 Installation From a Server

This chapter tells you how to place MAE on a server so that users can use the server to install MAE on their local workstations.

Setting up MAE on a server for remote installation

You can simplify the installation process for users by making the MAE installation files accessible on a Network File System (NFS) server on the network. This is particularly useful for enabling remote installations for users who do not have a CD-ROM drive.

Make sure that your company or organization has enough licenses to accommodate the maximum number of simultaneous users of MAE. For more information, see the license agreement that comes with MAE and the Appendix, “The License Manager.”

You have two options for setting up MAE for remote installation:

- Making the MAE CD-ROM disc directly accessible on a server
- Making the MAE files available on a server’s hard disk

Once you give users the permission to access the files, they can run the Installer from the server.

Both options require that you have superuser privileges and that the NFS client and server computers are correctly configured.

Note: The instructions in the following sections assume that you are familiar with UNIX networking commands and capabilities, including how to configure and manage NFS software. For details, see the documentation that came with your computers.

Making the MAE CD-ROM disc accessible directly

You need superuser (`root`) privileges to perform these steps.

To mount the MAE CD-ROM disc on an NFS server, follow these steps:

1. **Insert the MAE CD-ROM disc into a CD-ROM drive attached to the server.**
2. **Give users read permission for the directory where the MAE CD-ROM disc is mounted.**
3. **Mount the MAE CD-ROM disc on the server.**
4. **Export the directory in which you mounted the MAE CD-ROM disc.**
5. **Tell users how to install the software.**

   Users can remotely mount the directory on their workstations, start the MAE Installer, and install the MAE files from the server onto their workstations, according to the instructions given in Chapter 1 of the online *MAE User's Manual.*

   IMPORTANT Make sure that you tell users to run the MAE Installer from the directory that contains the MAE installation files, and that you have given users read permission for that directory.

6. **Ensure that the users have access to the license file, `license.dat`.**

   Users can install MAE without the license file, but they can't run MAE until a valid license file is in place. Once you obtain a valid license file, make sure that all users of MAE have access to it. See the Read Me file (`README.TXT`) for instructions on obtaining the license file.

   By default, MAE expects `license.dat` to be in the `apple` directory. You can change the location of the license file. See the Appendix, "The License Manager," for information about managing licenses with MAE.

Making the MAE files accessible from a server's hard disk

You need superuser privileges to perform these steps.

To copy the MAE installation files from the MAE CD-ROM disc to an NFS server, follow these steps:

1 **Insert the MAE CD-ROM disc into a CD-ROM drive attached to the server.**

2 **Mount the MAE CD-ROM disc.**

3 **Change to the directory where the MAE CD-ROM disc is mounted.**

4 **Copy the files from the MAE CD-ROM disc to the server.**

Enter the following command:

`find . -depth -print | cpio -pduv` *`destination-directory`*

Replace *destination-directory* with the name of the directory on the server on which you are placing the files.

5 **Export the directory in which you mounted the MAE CD-ROM disc.**

6 **Tell users how to install the software.**

Users can remotely mount the directory on their workstations, start the MAE Installer, and install the MAE files from the server onto their workstations, according to the instructions given in Chapter 1 of the online *MAE User's Manual*.

IMPORTANT Make sure that you tell users to run the MAE Installer from the directory that contains the MAE installation files, and that you give users read permission for that directory.

7 **Ensure that the users have access to the license file, `license.dat`.**

Users can install MAE without the license file, but they can't run MAE until a valid license file is in place. Once you obtain a valid license file, make sure that all users of MAE have access to it. See the Read Me file (`README.TXT`) for instructions on obtaining the license file.

By default, MAE expects `license.dat` to be in the `apple` directory. You can change the location of the license file and set the `LM_LICENSE_FILE` variable to the location. See the Appendix, "The License Manager," for information about managing licenses with MAE.

Installing and starting the license manager software

Users can set up and use license management by performing the following two steps:

1. **Copy the license file, `license.dat`, into the `apple` directory for each workstation to run MAE.**
2. **Start the license manager daemon, `lmgrd`.**

   For example, you can use the following command to start the license manager:

   ```
   /opt/apple/bin/lmgrd -c /opt/apple/license.dat
   ```

   The `-c` option must be used to specify the location of the license file.

This quick start approach allows users to start running MAE immediately, but it is not the most efficient approach to license management for workgroups. At a minimum, you should eventually consider using a single network-mounted copy of the license file, and you should modify your `rc` files to start the license manager daemon automatically.

Installing MAE on a server and sharing through NFS

As an alternative to having each user install a copy of MAE, you can install MAE on a server and share the `apple` directory through NFS. In this scenario, users mount the file system and run MAE.

With this approach, you can customize your environment and easily maintain your configuration. For example, you might choose to establish Finder Preferences settings for your entire site by modifying the contents of the `apple` directory. Additionally, you might want to install fonts in the `apple` director to avoid having users install multiple copies of those fonts.

2 Customizing MAE and Your Workstation

This chapter provides information about customizing the MAE startup environment. This chapter includes the following topics:

- using command options and X resources to customize MAE
- setting up NFS mounts
- installing kernel modules such as AppleTalk
- using a temporary installer volume (TIV) to install applications
- customizing System Folders
- improving performance
- creating aliases that appear at startup
- creating MAE independent volumes
- using the `APPLEPATH` environment variable

This chapter also offers help in troubleshooting startup problems.

Customizing MAE with command options and X resources

You can use command options to modify the behavior of MAE. For example, by entering the command

```
/opt/apple/bin/mae -iconic -pixelscale
```

you can start MAE in an iconified (minimized) state, reducing its processor usage until you open its window, at which time it is displayed magnified to twice its normal size.

X resources allow control over an X client such as MAE and its options, similar to the control offered by UNIX command options and environment variables. By specifying X resources in a common file, you can configure the MAE environments of many users at the same time.

Command options are loaded after X resources, so if you specify X resources, you can use the command options to temporarily override them.

Command options

The following table lists the command options for the `mae` command.

Command option	Description
`-appledir` *directorypath*	Specify the location of the apple directory. The default is the directory in which the MAE package was installed.
`-applepath` *directorypath*	Set the MAE environment variable APPLEPATH to specify a directory to be searched for application information when the desktop database is rebuilt.
`-display` *display*	Specify the X display server on which you want MAE to appear. The default is specified by the DISPLAY environment variable.
`-filedouble`	Force MAE to use AppleDouble format for files. Creates a separate file for the resource fork, indicated by a filename starting with %.
`-fstimeout` *seconds*	Specify the time in seconds that MAE will wait for a response from an NFS-mounted file system.
`-geometry` *width* x*height+X+Y*	Set the size and position of the MAE window (excluding the Toolbar) to *width* pixels wide by *height* pixels high (default is 640 x 480 pixels), with the top left corner at *X* pixels to the right and *Y* pixels down from the upper-left corner of the screen.
`-help`	Display all the MAE command options.
`-iconic`	Start MAE iconified. The default is for MAE to start up in an open window.
`-installcmap`	Specify that MAE should force the installation of its X colormap. Use this option for window managers or X servers that do not directly install colormaps.
`-macdir` *directorypath*	Specify the location of the .mac directory. The default is the user's home directory.
`-maxdepth` *bitdepth*	Specify the maximum frame-buffer bit depth. Setting maxdepth affects the number of colors available in the Monitors control panel. The default is dynamically set to the maximum supported by the X server. Valid *bitdepth* values are 1, 2, 4, 2, 4, 8, 16, and 24.
`-memory` *memorysize*	Specify the memory size for applications within MAE (in MB). The default is 16 MB.
`-noextensions`	Disable Macintosh system extensions (has the same effect as holding down the Shift key while booting a Macintosh computer).
`-nocursorscale`	Defeat scaling of the mouse pointer if specified with the -pixelscale option.
`-nosound`	Disable MAE sound support.
`-pixelscale`	Magnify display by doubling the number of pixels in both the X and Y axes.
`-rebuild`	Rebuild the desktop database when MAE is started. This option affects all MIVs mounted at startup.
`-sane`	Prevent MAE from using native RISC floating-point instructions. Provides greater compatibility with SANE at the expense of runtime performance.
`-sysfol` *system-folder*	Specify the directory *system-folder* to use as the System Folder. The default is $HOME/System\ Folder.
`-title` *string*	Specify the title to appear in the MAE window's title bar. The default is Macintosh Application Environment [*version*].
`-version`	Print the MAE version information.
`-zappram`	Specify that the virtual PRAM be reset when MAE is started.

X resources

X resources have been defined for several of the `mae` command arguments as follows:

`-appledir`

X resource	`Mae.appleDirectory`
Description	Pathname of apple directory to use
Example	`Mae.appleDirectory: /u/benjamin/apple`

`-filedouble`

X resource	`Mae.fileDouble`
Description	Cause MAE to use AppleDouble file format
Example	`Mae.fileDouble: True`

`-fstimeout`

X resource	`Mae.fsTimeout`
Description	Set the time MAE waits for a response from an NFS-mounted file system
Example	`Mae.fsTimeout: 15`
Default	10 seconds

`-geometry`

X resource	`Mae.geometry`
Description	Geometry string of the following form: <*width*>x<*height*>{±}<*xoffset*>{±} <*yoffset*>
Example	`Mae.geometry: 640x480+0+0`

`-iconic`

X resource	`Mae.iconic`
Description	Starts MAE up in an iconified state
Example	`Mae.iconic: True`

`-installcmap`

X resource	`Mae.installColormap`
Description	Forces MAE to install its own colormap
Example	`Mae.installColormap: True`

-macdir

X resource `Mae.macDirectory`

Description Pathname of the `.mac` directory to use

Example `Mae.macDirectory: /u/benjamin/mac.dir`

-maxdepth

X resource `Mae.maximumDepth`

Description Maximum bit depth for MAE

Example `Mae.maximumDepth: 8`

-nosound

X resource `Mae.noSound`

Description Disable sound support

Example `Mae.noSound: True`

-nocursorscale

X resource `Mae.noCursorScale`

Description If pixelscale is specified, do not scale cursor

Example `Mae.noCursorScale: True`

-pixelscale

X resource `Mae.pixelScale`

Description Magnify display by two in X and Y axes

Example `Mae.pixelScale: True`

-title

X resource `Mae.title`

Description String to use as title of MAE window

Example `Mae.title: MAE-1`

-sysfol

X resource `Mae.systemFolder`

Description Pathname of System Folder to use

Example `Mae.systemFolder: /u/benjamin/System Folder`

Note: X resources are not available for the `applepath`, `help`, `memory`, `noextensions`, `rebuild`, `sane`, `version`, or `zappram` options.

The loading order for X resources

MAE loads X resources from several different places and in a specific order.

Note: The expression $*VARIABLE* denotes an environment variable named `VARIABLE`. Environment variables can be set in the C shell with the following notation:

`setenv VARIABLE` *value*

Most other shells use the following notation:

`VARIABLE=`*value*`; export VARIABLE`

Note that when you enter a value for `$LANG`, `$XAPPLRESLANGPATH`, and `$XAPPLRESDIR`, you need to terminate the value with a slash (`/`).

MAE loads X resources in the following order.

1. **Load `/usr/lib/X11/${LANG}app-defaults/Mae`.**
2. **If step 1 fails, load `/usr/lib/X11/app-defaults/Mae`.**
3. **Load `${XAPPLRESLANGPATH}Mae`.**
4. **If step 3 fails, load `${XAPPLRESDIR}Mae`.**
5. **Load data from the X server.**
6. **If step 5 fails, load `$HOME/.Xdefaults`.**
7. **Load file specified by `$XENVIRONMENT`.**
8. **If `$XENVIRONMENT` is not set, load `$HOME/.Xdefaults-<`*`host`*`>` where *host* is the hostname of the machine on which MAE is running.**
9. **Load command options.**

   Note: Command options override all other resources. Also, using `xrdb` to load MAE defaults into an X server will set defaults for all MAE sessions on that X server, not just for a single user.

The order of occurrences is described in more detail here.

By default, MAE first searches for the following file:

`/usr/lib/X11/${LANG}app-defaults/Mae`

If `$LANG` is set and this file exists, MAE loads its values into a resource database. Otherwise, MAE searches for the following file:

`/usr/lib/X11/app-defaults/Mae`

Next, MAE looks for the file:

`${XAPPLRESLANGPATH}Mae`

The variable $XAPPLRESLANGPATH is a language-independent version of the variable $XAPPLRESDIR, which specifies a directory containing application-specific resource files. If $XAPPLRESLANGPATH is not set, MAE looks for the file:

${XAPPLRESDIR}Mae

After this, MAE looks for resources in the X server. These resources can be accessed through xrdb.

If this fails, MAE loads data from the .Xdefaults file in the user's home directory.

If the environment variable $XENVIRONMENT is defined, MAE loads the contents of any resource file that $XENVIRONMENT contains. But if $XENVIRONMENT is not defined, MAE loads the contents of the file in the user's home directory:

.Xdefaults-<*host*>

where *host* is the name of the workstation on which MAE is running.

After processing all the previously-mentioned resource files, MAE merges its command options into the resource database.

Configuring NFS mounts

There are three ways to mount NFS file systems that trade off data integrity against workstation availability:

- soft
- hard/interruptible
- hard/non-interruptible

MAE's ability to recover from an NFS server failure depends on whether NFS file systems are mounted as soft or hard/interruptible. Requests to file systems mounted hard/non-interruptible cannot be interrupted, so MAE will appear to freeze when trying to access such a file system if its server is down. Most administrators will find it preferable to mount NFS file systems as hard/interruptible.

Installing kernel modules

Users cannot run kernel modules (such as AppleTalk) unless they are installed on the users' local workstations. To simplify installation for users, you can provide them a command with options to install quietly (without warning messages), or you can customize the Easy Install procedure to automate the installation of the kernel modules. These techniques are described in the following sections.

Providing a customized command line

To perform a quick installation of the AppleTalk kernel module without any warning messages, run the following command (you must have superuser privileges):

`INSTALL.SH [-q][-R] -c E2_ATALK [-d` *apple_directory*`]`

The options are as follows:

`q` Suppress warning messages.

`R` Reboot the system upon completion of the installation.

`c` Perform the Custom installation process, installing the listed MAE product component package (in this case, just the AppleTalk kernel module, `E2_ATALK`.)

`d` Specify the pathname (represented by *apple_directory)* of the `apple` directory, where MAE is to be installed. Note that you must specify the full pathname.

Customizing the Easy Install procedure

Some sites install MAE on a central NFS or Apple File System (AFS) server. MAE users at that site share a common copy of the software. While MAE is designed to work well in this scenario, it does require that all users who want to use AppleTalk network services or Macintosh floppy disks (on Solaris) install the related module packages on their local workstations.

The following procedure allows you to eliminate one or more packages from the MAE Installer to simplify the installation of AppleTalk and the Solaris Volume Manager Extensions. This allows you to use the Easy Install procedure to install only the packages that you want with a single command.

Note: You must be logged in as `root` and the MAE CD-ROM must be mounted at `/cdrom/mae` to complete the following steps.

1 Copy the contents of the MAE Installation CD-ROM disc to a file server.

For example, you can use the following commands to copy the files into the directory `/share/MAE_Install`:

```
cd /cdrom/mae
mkdir -p /share/MAE_Install
find . -depth -print | cpio -pdumv /share/MAE_Install
```

2 Remove all of the unwanted installation packages.

Packages beginning with the letter “E” are part of the Easy Install. Packages beginning with the letter “P” are demonstration packages.

The packages are as follows:

`E1_MAE`	Macintosh Application Environment 3.0, the MAE executable program, and support files.
`E2_ATALK`	AppleTalk Stack kernel module, which provides AppleTalk support. The user must be logged in as root to install this package.
`E3_VOLD`	Solaris Volume Manager Extension, which allows the use of Macintosh (HFS) floppy disks in the floppy drive on a Sun workstation.
`E4_DOCS`	MAE online manuals, in PDF format, and the Acrobat Reader application.
`E6_APPLX`	Extras from Apple, including OpenDoc and sample applications.
`E7_DSKTP`	Common Desktop Environment (and VUE, if applicable).
`L8_LMGRD`	FLEXlm License Manager server software. Only one is needed per network.
`P1_WORKS`	Demonstration version of the ClarisWorks application.
`P2_IMPCT`	Demonstration version of the Claris Impact application.
`P3_DRAW`	Demonstration version of the ClarisDraw application.
`P4_FMPRO`	Demonstration version of the FileMaker Pro application.

The packages are located in a directory named for the operating system to which they correspond. Following the previous example, you could enter one of the following commands to change to the correct directory for your system:

```
cd /share/MAE_Install/SOLARIS
cd /share/MAE_Install/HPUX
cd /share/MAE_Install/HPUX9
```

To remove the unwanted packages, enter

`rm E1_MAE.*` (to remove the main MAE package)

`rm E4_DOCS.*` (to remove the online documentation package)

`rm P*` (to remove all of the demonstration software packages)

Note: You should remove only those packages that you do not want to install.

3 **On the workstation on which the kernel module packages are to be installed, have the user enter the command line to start an Easy Install process.**

For example, the following commands cause the installer to automatically install all of the remaining "easy installation" packages (those whose file names begin with an "E"):

```
mount server:/share/MAE_Install /mnt
```

```
/mnt/INSTALL.SH -e /tmp
```

Note: The AppleTalk and Solaris Volume Manager Extensions are installed into the UNIX kernel, not in `/tmp`.

Packages other than AppleTalk or the Solaris Volume Manager Extensions will be installed in the directory specified on the command line. If you do not specify a directory, the packages will be installed in the default directory. In the example above, you would replace `/tmp` with a more appropriate location for the other packages.

4 **Be sure the user reboots the workstation before attempting to use AppleTalk.**

Installing applications using a temporary installer volume (TIV)

The temporary installer volume (TIV) is an option in MAE that helps the user install application software. Some application installers try to install directly to the root directory rather than to a user-defined location. If this problem occurs, you can advise the user to install into a TIV first, then move the application to the appropriate location.

Note: The temporary installer volume feature is supported to maintain continuity with previous versions of MAE. It is recommended that the user first try copying the application files to a MAE independent volume (MIV) and running the installer from the MIV. See "MAE Independent Volumes," later in this chapter, for more information.

Creating and using a TIV

The user can create and use the TIV with the MAE General control panel and can set the application's installer to install into this volume. The user must then copy the newly installed files to the appropriate place in the file system.

IMPORTANT When MAE is shut down, the TIV will be lost; however, the user will be warned about this and will be given the opportunity to cancel the shutdown procedure.

The TIV will survive if you restart MAE instead of shutting it down. Installer programs often restart MAE automatically after installation.

Note: Because a TIV file is temporary and frees its disk space when MAE shuts down, it is not linked to the file system so you cannot see the TIV from the UNIX side using the `ls` command.

For more detailed information, see the *MAE User's Manual*, Chapter 3.

Making a TIV persistent

The TIV is temporary because of access issues. The TIV is an HFS file system enclosed entirely within a UNIX file. To MAE, a TIV appears as a Macintosh hard disk, but to the UNIX environment, it is a single file. The files contained within the TIV are not accessible from the UNIX environment. If the TIV file is corrupted, the data it contains (that is, all files on the TIV) will be inaccessible. Be aware of this before attempting to use a TIV for your daily work. Save data files to the UNIX file system to prevent a potential loss of data.

To make a TIV persistent across MAE sessions, edit the user's `TIV_pref` file (which is in the Preferences folder of the user's System Folder) to add a line similar to the following:

`container1:`*`disk_name`*`:/user/home/`*`user_name`*`/TIV_FILE:20000000:keep`

This command creates a TIV approximately 20 MB in size. The fields are arranged as follows:

- The first field is a keyword; it must contain at least the first four characters `cont`.
- The second field is the name of the disk in the MAE environment. It may be renamed in the Finder, but the change will not appear here.
- The third field contains the location of the disk file that is to be created.
- The fourth field is the size of the disk in bytes. It will be rounded up to the nearest block size (512 bytes).
- The fifth field must contain `keep` to indicate to MAE that the TIV that the line represents is to be persistent.

You can create up to ten persistent TIVs.

> **WARNING** The files contained within the TIV are not accessible from the UNIX environment. If the TIV file is corrupted, all files in the TIV will be lost.

Customizing System Folders

The System Folder is crucial to the Mac OS. Aside from holding the system software, it serves as a central storage site for such items as fonts and control panels. Users of the Mac OS often customize their user experience by modifying the contents of their System Folder. Moreover, applications routinely install their required files (for example, preference files and device drivers) in well-defined folders inside the System Folder.

The MAE System Folder is different from a Macintosh System Folder in some important ways:

- The MAE System Folder contains special versions of the System, Finder, extensions, and control panels specifically designed to run on UNIX workstations. Don't replace MAE system files with those from a Macintosh computer or allow installation programs to replace them. See "Extensions" in Chapter 5.
- In MAE, many common resources are stored in the file `apple/lib/data` rather than locally. This minimizes the disk space requirements for each user's System Folder.

The first time MAE starts, it looks for a System Folder in the user's home directory or in the directory specified with the `-sysfol` option or `Mae.systemFolder` X resource. If it does not find one, it creates one using the `apple/sys` directory as its source. You can customize `apple/sys` to suit your particular environment. For example, you can add fonts, desk accessories, or extensions to the `apple/sys` directory. You add items by putting them into the same folders within `apple/sys` as they would be put in by installing them into a personal System Folder.

Whenever a new user runs MAE, your additions to the `apple/sys` directory are put into the new System Folder created for that user. If the user already has a personal System Folder, changes you make to `apple/sys` are *NOT* applied to that user's personal System Folder. Such changes must be made for each personal System Folder on the workstation.

You can force the creation of a new System Folder by removing the existing one and restarting MAE. However, this process also removes any changes that have been made to that System Folder. Keep in mind that installing applications may have caused preference files or files required by the application to be put into the System Folder. Removing the System Folder removes such files, potentially causing problems for the application and the user.

Note: If `apple/sys` is corrupted, you might have to reinstall MAE.

Distributing common extensions, control panels, and preference files

If numerous users will be utilizing the same mail application, meeting planner, or other productivity application, it is likely that one or more files will need to be added to their System Folders. By adding these files (typically extensions and control panels) to the `apple/sys` directory (in their respective folders), a new System Folder can be created for each new user that already contains the necessary files. This way, only the application may need to be copied to the user's home directory. In a similar fashion, you can distribute preset preferences files as well.

Note: This method of distributing common resources does not affect existing System Folders.

For example, you might wish to place all the necessary extensions and control panels in the `apple/sys` directory for a network-based mail application, and include the preferences file for that application. This would ensure that the user already had the proper name and location of the mail server as well as any additional system resources required by the application.

You can also use the System Folder to preset user values for MAE that appear in the MAE General control panel. You can preset these performance values and prevent individual users from changing them by following these steps:

- Set the values as described above, thereby creating a customized `Finder Preferences` file in the local `Preferences` folder.
- Copy the resulting `Finder Preferences` file into the directory `apple/sys/Preferences`.
- Remove the MAE General control panel from the directory `apple/sys/Control Panels`.

This prevents users from changing the values of MAE General.

Using global fonts to minimize disk usage

By default, MAE users do not share their individual Fonts directories. Since each user must be able to modify the directory (for both adding and deleting fonts), management of the Fonts directory as a modifiable shared resource would be difficult. As an alternative, have the system administrator create a globally accessible Fonts directory that is read-only and allow only the superuser to add or delete fonts. In this manner, the directory can contain many fonts that would otherwise be duplicated for each user, thus preserving more disk space. Once this has been done, users can delete their private Fonts directories and create a symbolic link to the shared Fonts directory (using `` `ln -s' ``).

Improving MAE performance

This section describes some things you can do to improve the performance of MAE. For further information on improving performance, see Chapter 4.

- **Put the System Folder on the local machine.**

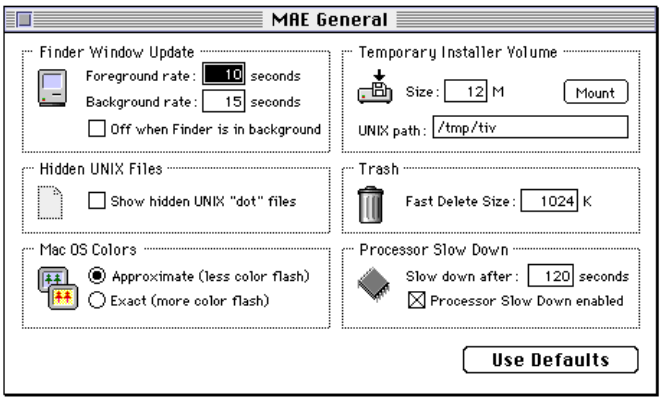
  The MAE Finder and Macintosh applications use resources from the System Folder frequently. If your System Folder is on a remote machine, move the System Folder to your local machine, then use the `-sysfol` command option to specify its location.

- **Run MAE in a small memory partition.**

  Reduce the amount of memory used by MAE, either by using the `-memory` command option or by using the Memory control panel.

- **Set the Finder Window Update option to a higher value.**

  The Finder Window Update option in the MAE General control panel determines how often the Finder checks for new files that were created outside MAE.



For example, if you created a text file in your home directory (using a utility such as `vi`), and your home directory was open in MAE, you wouldn't see that file until the next Finder check occurred. You can force a Finder check for a window in MAE by closing and opening that window. If you do not need frequent Finder checks, you can improve performance by setting this number to a high value (such as 120 seconds) and turning off Finder Window Update when the Finder is in the background.

Note: Setting Finder Window Update to a high value does not affect the files that appear in a Macintosh application's Open and Save dialog boxes, since those dialog boxes get their information directly from the File Manager when they are displayed.

You can preset these performance values and prevent individual users from changing them by following these steps:

—Set the values as described previously, thereby creating a customized `Finder Preferences` file in the local `Preferences` folder.

—Copy the resulting `Finder Preferences` file into the directory `apple/sys/Preferences`.

—Remove the MAE General control panel from the directory `apple/sys/Control Panels`.

This prevents a user from changing the values.

- **Choose the Approximate option in the Mac OS Colors section of the MAE General control panel, if you are using color.**

  This option prevents MAE from allocating extra memory for a graphics buffer. For more information, see Chapter 4.

- **Increase the number of seconds in the Processor Slow Down setting.**

  In order to prevent MAE from using the workstation's processor unnecessarily when MAE is not in use, the Processor Slow Down setting causes MAE to check for activity at a reduced rate after a certain number of seconds of inactivity have elapsed. The default is 120 seconds.

- **Run fewer copies of MAE.**

  The more copies of MAE you run from a single workstation, the slower each copy will be. You might want to try running MAE locally even if the local machine is typically slower than the remote machine.

- **When running on Solaris, increase the shared memory size.**

  The default shared memory size may not be optimum for running MAE. Try to set shared memory to at least twice the size of the graphics buffer. (See "Graphics Buffer" in Chapter 4 for information about estimating the buffer size.) If you are using the pixel-scaling option, try increasing shared memory to six times the size of the display selected in the Monitors control panel or twice the size of the X window in which MAE is displayed.

Note: Due to network traffic limitations in previous versions of MAE, setting the X server display to black and white improved MAE's performance when running remotely. These limitations have been removed, so using color on a remote display is not significantly slower than using black and white.

Applications and processor performance

Mac OS applications, whether they are running on MAE or on a Macintosh, normally dispatch each "event" as it occurs from a main program loop. Examples of events are mouse movements and screen updates. To await the next event, Macintosh programs normally use the WaitNextEvent routine, which waits for an event or the expiration of a timeout while yielding processing time to other Macintosh programs. While WaitNextEvent is waiting (and provided no events need attention), MAE sleeps, imposing a negligible load on the processor.

Some applications (such as SimpleText) increase responsiveness at the cost of higher processor load by using very short timeouts while waiting for events. Running under MAE, such programs use the processor heavily. Programs with longer timeout values, such as MacWrite II, work well in the MAE environment.

Improving workstation performance when not using MAE

If you iconify (minimize) MAE, it significantly reduces its processor use. You can improve workstation performance by iconifying MAE when you are not using it. By iconifying MAE, you approximate the UNIX `sleep` command.

Alternatively, open the MAE General control panel and set Finder Window Update to "Off when Finder is in the background." Then open the Stickies application (from the Apple menu) and leave this application in the foreground. This application uses very few CPU resources and significantly reduces MAE's processor use.

Configuring MAE to create aliases that appear at startup

You can configure MAE to create *aliases* to specified files or directories so that users have convenient access to these items. (For information on aliases, see Appendix A of the *MAE User's Manual*.) This section tells you how to set up MAE so that it automatically creates an alias for an item and puts the alias's icon in the user's startup volume (or on the MAE desktop) when the user starts MAE.

Note: An alias created in this manner is different from a UNIX directory that has been made an independent volume. For more information, see "Creating Independent Volumes," later in this chapter.

To configure MAE to create and display an alias when a user starts MAE, follow these steps:

1 Move to the `apple/lib` directory.

For example, if you installed MAE in the `apple` directory in your home directory (and your home directory is your current directory), type `cd apple/lib` in your terminal emulator window.

2 Open the `AliasList` file with a text editor.

The file contains a series of command lines similar to the following:

type | *alias-location* | *pathname*

The following table lists the syntax elements used in the command line.

Element	Value	What it specifies
alias-location		Where the alias appears
	`apple`	In your Apple Menu Items folder
	`desk`	On the MAE desktop
	`extensions`	In your Extensions folder in the System Folder
	`extras`	In your Apple Extras folder
	`home`	In your home directory
	`launcher`	In the Launcher Items folder in your System Folder
	`launcher-third_party`	In System Folder/Launcher Items/•Third Party
	`launcher-demos`	In System Folder/Launcher Items/•Demos
	`launcher-docs`	In System Folder/Launcher Items/•Documentation
pathname		The pathname of the file or directory you want the alias to represent. You can specify either absolute or relative pathnames. If the path is a relative pathname (does not start with a slash), it is considered relative to the apple directory. For example, you might specify the path as Apple Extras/SimpleText.
type		Whether the alias reappears if you delete its icon and then restart MAE
	`nonpersistent`	The alias *does not* reappear on restart
	`persistent`	The alias *does* reappear on restart

Note: Nonpersistent aliases that have been deleted *will* reappear if the user deletes the System Folder and then restarts MAE.

3 Edit the command lines as appropriate to create an alias for a particular file or folder.

For example, you might want MAE to create an alias to a folder called `claris.docs` (located in the directory `/applications/claris`) on the desktop every time the users start or restart MAE (even if they have deleted the icon for the alias during the previous MAE session). To have MAE create the alias, you might type the following:

```
persistent|desk|/applications/claris/claris.docs
```

4 **Save your changes to the `AliasList` file.**

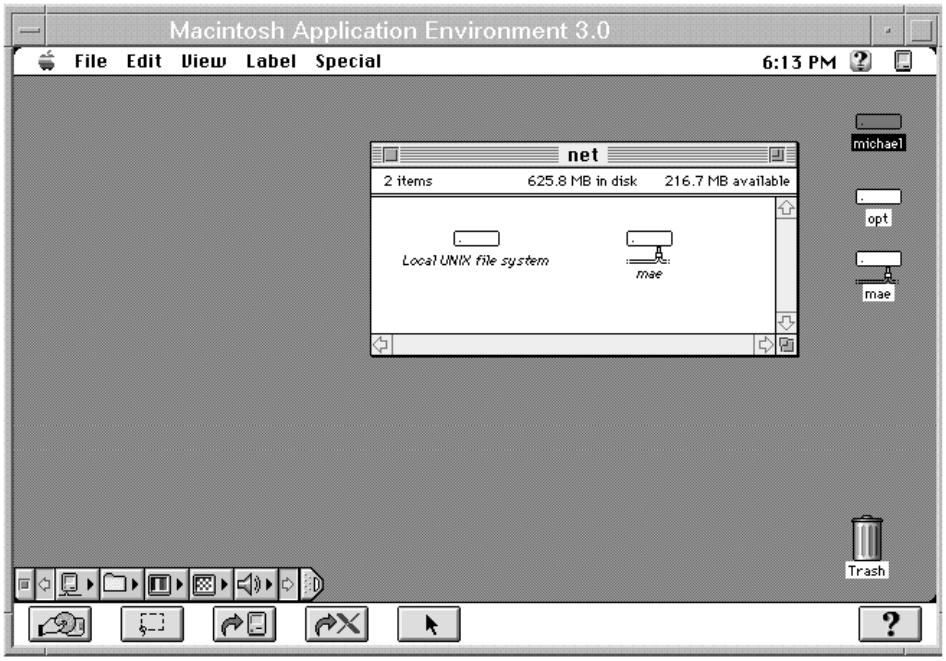
The next time a user starts MAE, the alias appears in the location that you specified. If you followed the example in step 3, an alias folder icon labeled `claris.docs` would appear on the MAE desktop.

MAE independent volumes

A UNIX file system appears to the user as a single hierarchical tree, with all files and directories branching off from the single root directory, named `/`. Other file systems, local or remote, can be mounted at some point (often specified by the user) on that tree below the root node.

In the Mac OS, the file system can appear as several hierarchical trees that do not share a single root. Each of these trees is called a volume. A volume can be a file system on a local or remote hard disk, CD-ROM, or other storage device. Since such devices can be divided into multiple partitions, there can be multiple volumes on a single device. The Finder displays an icon on the desktop for each mounted volume.

MAE portrays each UNIX file system as a Mac OS volume. Additionally, you may select UNIX directories that will represent the root of a Mac OS volume. A UNIX directory or a UNIX file system mount point portrayed in this way is called an *independent volume*. MAE displays an icon on the desktop for the directory and a Macintosh alias to that directory in its UNIX parent directory. For example, in the following illustration, the directory `mae` is a remote UNIX file system that has been made an independent volume.



Note that the `net` folder contains the alias to the `mae` volume shown as an icon on the desktop. Double-clicking either the alias or the volume icon opens a window for the `mae` volume. No change has been made from the perspective of the UNIX file system. Listing the contents of the `mae` directory in UNIX shows the same contents as the open MAE window.

An alias representing an independent volume is considered to be locked; that is, it cannot be moved or removed, and its name cannot be changed.

Independent volumes are mounted during startup or when a UNIX mount point is referred to. This can happen either at the mount point or through a symbolic link to the mount point. MAE creates an independent volume for each mounted UNIX file system when it discovers the existence of the mounted file system. At that time, the alias is created in the parent directory, but the icon does not appear on the desktop. If you double-click the alias, the icon appears on the desktop and a window for the folder opens.

Note: By default, the UNIX root (/) file system does not appear as a MIV on the desktop. If you want this MIV to appear, you must define it in your `.MIV_file`. See the section "Defining Independent Volumes" later in this chapter.

Creating independent volumes

An independent volume can be created in one of the following ways:

- by the administrator editing the `.MIV_file`
- by MAE creating them for UNIX mount points
- by MAE creating them to access system resources

You can have up to 50 MIVs.

Each type of independent volume looks like a normal Mac OS volume, but to MAE they have a few differences.

Independent volumes you define

You can designate a UNIX directory as an independent volume by creating an entry for the directory in the independent volume definition file, the `.MIV_file`. The default `.MIV_file` is in the `/appledir/lib` directory. MAE copies this file to `.mac/$HOST` when the `.mac` directory is created. This approach lets you use a single customized definition for your site. The active `.MIV_file` is always in `.mac/$HOST`. MAE uses this version to determine which MIVs to mount on the MAE desktop at startup.

Independent volumes for UNIX mount points

MAE creates an independent volume for each UNIX mount point it finds. This includes local mount points and remote NFS mount points. At startup, MAE scans the mount table and creates (but does not mount) an independent volume. (Only independent volumes defined in the `.MIV_file` are mounted at startup.) MAE periodically scans the mount table to see if entries have been added or removed. These independent volumes are represented just like the independent volumes you define. In MAE you see an alias to a Mac OS volume where the mount point is in the UNIX file system. For example, if `/usr/jerry/mount_here` is a mount point, then you will see an alias at `/usr/jerry` to an independent volume named `mount_here`.

Independent volumes created by MAE to access system resources

At startup, MAE ensures critical resources are always accessible to the Mac OS file system. These resources exist in the `apple` directory, System Folder, and `.mac` directory. If any of these directories are not in an independent volume you define in the `.MIV_file`, an independent volume will be created at the parent of the directory. For example, if your `apple` directory is in `/externaldisk/appledirs/apple-3.0` and you only defined an independent volume at `/export/home/me`, then MAE creates an independent volume at `/externaldisk/appledirs`. Since these independent volumes contain resources the MAE file system must access, you will not be able to unmount these independent volumes.

Defining independent volumes

The definition file (`.MIV_file`) has a simple format. Each entry must be on a line by itself (it cannot continue to another line). The maximum line length is 1152 characters. Each entry takes the form

pathname volume_name

The value *pathname* is the full UNIX pathname of the directory.

The optional value *volume_name* is the name to be used for the desktop icon. If the *volume_name* value is not specified, the directory name (last element of the pathname) is used instead. The volume name cannot exceed 27 characters.

The two values must be separated by one or more space or tab characters.

The following listing shows the contents of a simple `.MIV_file`. (A comment begins with a pound sign [#] and can occur at the end of an entry line or on a line by itself.) The tilde (~) is a wild card that represents the user's home directory as defined in `/etc/passwd`. Do not use this wild card for other path names.

```
#Filename .MIV_file
#Simple independent volume definition file
~  # The tilde represents the user's home directory.
/u/dir1 Mac\ Things   # The directory /u/dir1, named Mac
                      # Things on the desktop.
```

IMPORTANT The definition file must include at least one valid entry. This file must also have read and write access permission. If the file does not have read and write permission, MAE does not start.

Note that pathnames or volume names with embedded space (blank) characters must be enclosed in double quotation marks if each space is not escaped with a backslash character (\).

Note: You cannot move or delete a defined MIV. To move or remove a MIV, you must shut down MAE, edit the `.MIV_file`, and then move or remove the directory using UNIX commands.

Unmounting independent volumes

To unmount an independent volume, drag its icon to the Trash. Within MAE, as in the Mac OS, dragging a volume's icon to the Trash does not delete its files or directories, but simply unmounts it *within MAE*. The alias at the original mount point is not removed. You can double-click the alias to make MAE mount the original directory again as an independent volume.

Renaming independent volumes

To rename an independent volume, use one of the following methods:

- shut down MAE and edit the name in the `.MIV_file`
- select the volume name in the Finder and type a new name

Symbolic links with MAE

In MAE, UNIX symbolic links are represented as alias files. MAE builds the alias files for symbolic links as they are encountered by the system (usually when you open a folder that contains the link). This icon representing the link is the same icon as the item it represents. For example, if you have a symbolic link to a directory, the icon representing the link will be a folder. If you have a symbolic link to a mount point, the icon will be a volume icon.

Note: If you get a folder icon representing a mount point, the MAE mount table does not have an entry for the UNIX mount point. MAE will automatically create this entry and change the icon. This icon has no effect on the alias file—it still resolves to the correct volume.

MAE treats all UNIX mount points as separate Macintosh volumes. When you double-click an icon representing a symbolic link, MAE mounts a new volume on the desktop and opens a window displaying the contents of the mount point. The volume on your desktop is like any Macintosh volume. When you are finished with the volume, you can drag the volume icon to the trash to unmount it.

The Macintosh Desktop database

In MAE, files created by Macintosh applications are mapped to the application that created them. This allows Macintosh and MAE users to double-click a file in order to launch the correct application and open the file no matter where the application is located. The Finder coordinates this using the information in the Desktop database, which also maintains other information such as icon information and comments entered by the user.

The Desktop database is created the first time a user runs MAE or when the user launches MAE with the `-rebuild` command option. As MAE starts, the user sees the message "Rebuilding desktop" and the name of the directory currently being searched as the database is created. By default, the Desktop information is gathered from the `apple` directory. When a folder is opened for the first time since the last Desktop rebuild, information is added to the Desktop database concerning the creator/type and icon for all files contained at the top level of that directory.

Note: You cannot use the Command-Option key shortcut to rebuild the desktop—you must use the `-rebuild` option.

Once an application is in the Desktop database, any file created by that application, regardless of where it is located on the file system, can be properly opened by double-clicking the file. If the application is not in the Desktop database when the document is double-clicked, the user sees the following message:

This document "*document name*" could not be opened, because the application "*application name*" could not be found.

To avoid this problem, the user can open the folder that contains the application. At that time MAE updates the Desktop database with the information for that application.

Note: When you rebuild the desktop, only items in the `apple` directory and listed in the `APPLEPATH` variable are rebuilt. See the next section for information about the `APPLEPATH` variable.

Using APPLEPATH to simplify launching

You can use the APPLEPATH environment variable to simplify launching applications for first-time MAE users.

APPLEPATH represents a complete path name to a directory (or directories) that will be searched for application information whenever the Desktop is rebuilt; that is, the first time a user uses MAE or whenever the user launches MAE with the `-rebuild` option. The directories specified by APPLEPATH (and all their subdirectories) will be searched regardless of their physical location on the file system.

Note: The `-applepath` command option overrides the APPLEPATH environment setting.

The APPLEPATH variable is similar to the PATH variable. You can separate directory path names with colons.

For example, if several Macintosh applications are maintained on a remote server named `silverware`, the following APPLEPATH definition could be used:

```
setenv APPLEPATH /nfs/silverware/applications
```

or

```
set APPLEPATH=/nfs/silverware/applications:/usr/bin/Apps
export APPLEPATH
```

You can launch MAE with the `-rebuild` command option to rebuild the Desktop database if the database becomes corrupted or deleted. Corruption of the Desktop database can occur after a crash or when an application is incorrectly installed. A symptom of corruption of the database is the absence of icons in the Finder windows or the presence of incorrect icons for files created by applications that are in the Desktop database. Rebuilding the Desktop database restores the correct icons.

Starting MAE with the `-rebuild` command line option while the environment variable is set as shown previously adds to the Desktop database all the application file information contained in the `applications` directory and its subdirectories. This information stays in the Desktop database until the next Desktop rebuild. The APPLEPATH variable can now be deleted.

If new applications are added later to the directory specified in APPLEPATH, they will not be automatically updated in the Desktop database. They will be added only after the user opens the folder manually or restarts MAE with the `-rebuild` option.

Note: You cannot use keyboard shortcuts to rebuild the desktop as you can on a Macintosh computer.

Building the Desktop database with the entire file system (for example, by specifying `APPLEPATH='/'`) may take a great deal of time, depending on the size of local and remotely mounted file systems.

IMPORTANT While the Desktop is rebuilding, you can press the Stop button without adverse side effects. However, it is best not to stop the Desktop rebuild until after it has finished rebuilding the `apple` directory.

Locating applications for optimal performance

To obtain the best possible performance from your Macintosh applications with MAE, you can place all applications in one central directory on a local disk. By doing this, you will decrease the time it takes to load the applications into memory because applications on remote volumes are subject to network performance. Even if you are not able to store the applications locally, placing them in a single directory enables MAE to better associate documents with their creator applications. This is performed the very first time MAE is launched, and is referred to as rebuilding the desktop.

AppleSingle and AppleDouble file formats

Files in the Macintosh hierarchical file system (HFS) can consist of two parts: a data fork and a resource fork. The data fork typically contains just text or data. The optional resource fork typically contains file attributes such as Finder information (document type, creator application, icon, and so forth), formatting information, and message strings. Other file systems usually don't support such attributes, and therefore usually don't recognize resource forks. To preserve all attributes supported by the Macintosh hierarchical file system (HFS) on such systems, Apple supports two file formats: AppleSingle and AppleDouble format. MAE can read and write both formats.

Note: This difference in file systems is the reason that MAE and UNIX often report different sizes for the same file.

By default, MAE stores files (except text files) in AppleSingle format. AppleSingle format provides a structure for storing all contents and attributes of a file in a single file. This structure minimizes the potential for file corruption (in particular, the loss of the resource fork information) when the file is stored by a foreign file system.

AppleDouble format preserves the separation of the data and resource forks by storing each in its own file. The two files share the same name except that the name of the resource fork begins with a `%` sign. For example, a file named `file1` would appear as `file1`, the data fork file, and `%file1`, the resource fork file.

Macintosh applications are likely to have a resource fork. Document files often don't have a resource fork; those that do have one often use it to store only icons and formatting information. For such files it is sometimes possible to use the information in the data fork by itself. MAE takes advantage of this fact by storing text files in AppleDouble format. Mac OS applications can use both forks of the files as they normally would, and UNIX applications can use just the data fork file and ignore the resource fork file.

Generally, the use of the two file formats is transparent. If, for example, you copy UNIX files between two copies of MAE, each MAE handles the files appropriately. However, if you were to copy UNIX files to a Macintosh computer, those files would be given a resource fork. Then those same files, subsequently copied to MAE, would be treated as Macintosh files in AppleSingle format and would likely be unusable by UNIX.

If you encounter problems with a particular application being unable to open any files, use the AppleDouble control panel to specify that application should save documents in AppleDouble format. Alternatively, you can use the utility `asdtool` (installed in the `apple/bin` directory by default) to convert between the two formats. For complete information about the `asdtool` command, see the manual page.

Troubleshooting

This section contains solutions to situations you may encounter when configuring MAE.

Insufficient memory on the SparcStation

If you start MAE on a SparcStation, you may see the following message:

```
Insufficient shared memory or swap space. Using malloc instead.
```

On a SparcStation local display, MAE attempts to enhance graphics performance by using the X Shared Memory Extension to communicate with the X server.

For MAE to succeed at this, your Solaris kernel must be configured to support several large shared-memory segments. If MAE is unable to obtain the shared memory it requires (based on the size and color depth of the MAE window), it falls back to a less efficient communications scheme.

If you see this message, try the following to ensure that sufficient Solaris kernel resources are available to support efficient execution of MAE:

- **Edit the file `/etc/system` to include the following:**

  ```
  set shmsys:shminfo_shmmax=16777216
  set shmsys:shminfo_shmseg=8
  ```

  These `shmsys` parameters should cover most screen and color-depth configurations. They will take effect after you reboot your system.

3 Structure and Function of MAE

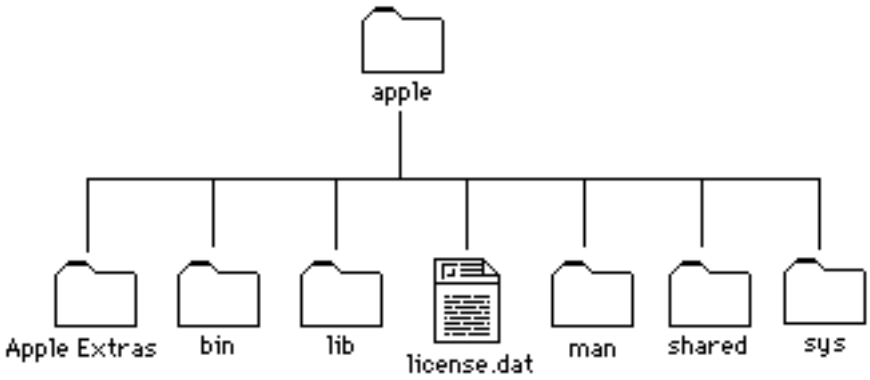
This chapter describes the following:

- the arrangement of MAE files
- the functions of MAE files
- the differences between MAE and the standard Macintosh environment

The contents of the `apple` directory

The `apple` directory contains some of the parts necessary to run MAE. The additional parts needed for each user of MAE are in the `.mac` directory and the System Folder. The first time a user launches MAE, it creates copies of these two directories in the user's home directory. The System Folder and the `.mac` directories are necessary for MAE to run. MAE may create other items in the user's home directory, depending on which MAE packages are installed and what modifications, if any, have been made to the `apple` directory. If you are running AppleTalk, you also need the AppleTalk kernel modules.

The figure below illustrates the structure of the `apple` directory.



The `apple/Apple Extras` directory

The `apple/Apple Extras` directory contains the following:

- The directory `apple/Apple Extras/AppleScript` contains the software and documentation necessary to create scripts that automate tasks within MAE. Some sample scripts are also included.
- The directory `apple/Apple Extras/docs` (if present) contains the online MAE manuals in PDF format and the Mac OS version of the Acrobat Reader application you can use to read them.
- The directory `apple/Apple Extras/demos` (if present) contains the demonstration versions of several applications.

- The directory `apple/Apple Extras/HyperCard Player` contains the HyperCard Player application and some HyperCard stacks, including one that explains what HyperCard is and how to use it.
- The directory `apple/Apple Extras/Movie Player` contains software you can use to edit and play QuickTime movies.
- The directory `apple/Apple Extras/Third_Party` contains non-Apple software for you to try.

  Note: The software in this directory is not supported by Apple Computer. Please read the disclaimer document in the folder before using any of the software. If you find an application useful, please pay the shareware fee.
- The file `apple/Apple Extras/SimpleText` is the SimpleText application that allows you to create, open, read, and modify text files.

The `apple/bin` directory

The `apple/bin` directory contains the following UNIX executable files:

- The file `apple/bin/mae` is the program that you run when starting MAE. This file, in turn, runs the file `apple/lib/engine`, a data file that contains data necessary to create the Macintosh environment.
- The file `apple/bin/macd` is a daemon that runs whenever `apple/bin/mae` runs and helps MAE interact with the UNIX environment. It also cleans up after `mae` if the `mae` process is killed.
- The file `apple/bin/lmgrd` is the license manager daemon that manages the user licenses to run MAE. These files are only present if you have installed the FLEXlm license manager package.

  Utilities associated with the license manager are as follows:
  —`lmborrow`
  —`lmcksum`
  —`lmdiag`
  —`lmdown`
  —`lmhostid`
  —`lmremove`
  —`lmreread`
  —`lmstat`
  —`lmswitchr`
  —`lmutil`
  —`lmver`
  —`lmutil`

For information on using the license manager, see the Appendix.

- The file `apple/bin/AppleMAE` is the MAE daemon that collaborates with the license manager daemon, `lmgrd`, to ensure the availability of floating licenses to users. This file is only present if you have installed the FLEXlm license manager package.
- The file `apple/bin/legal` displays the MAE license agreement.
- The file `apple/bin/appleping` is a program that allows you to test whether a host on an AppleTalk network is receiving packets.
- The file `apple/bin/appletalk` is a program that allows you to display AppleTalk status and to start or terminate AppleTalk services.
- The file `apple/bin/asdtool` is a program that allows you to convert files between the AppleSingle and AppleDouble formats.
- The file `apple/bin/atlookup` is a program that allows you to obtain information about the network-visible entities registered on an AppleTalk network.

Most of the UNIX executable files are described by UNIX-style manual pages that you can read using the `man` command.

The `apple/lib` directory

The `apple/lib` directory contains the following:

- The file `apple/lib/Default_MIV_file` contains a list of the directories to be mounted as static independent volumes.
- The file `apple/lib/AliasList` contains a list of command lines that allow you to set aliases to appear when you start MAE. For more information, see Chapter 2.
- The file `apple/lib/btree`.
- The folder `apple/lib/ccm` contains various character set mappings.
- The `apple/lib/data` file contains compressed resources used by the System Folder.
- The file `apple/lib/engine` contains the code necessary to create the Macintosh environment.
- The file `apple/lib/KeymapDepotDB` contains the keymap information for the keyboards supported by MAE.
- The folder `apple/lib/legal` contains the localized contents of the MAE license agreement.
- The folder `apple/lib/locale` is used in providing the appropriate message catalog.
- The file `MajorUpdate` is used by the installer (`INSTALL.SH`) in upgrading MAE to a new version.

The `apple/license.dat` file

The `apple/license.dat` file contains the license information required by the MAE license manager, `lmgrd`. This file may not appear in the `apple` directory, depending on how your site structures its licenses.

The `apple/man` directory

The directory `apple/man` contains the UNIX manual pages for the UNIX commands provided in MAE.

The `apple/shared` directory

The directory `apple/shared` is shared by everyone using the `apple` directory. This directory contains Apple Guide documents, applications in the Apple menu, and printer descriptions.

You cannot modify these files.

The `apple/sys` directory

The `apple/sys` directory contains the generic System Folder components used by each user.

The `apple/sys` directory also contains a `font` directory with the following fonts:

- AppleIcon Two
- AppleIcon Three
- Chicago
- Courier
- Geneva
- Helvetica
- Monaco
- N Helvetica Narrow
- New York
- Palatino
- Symbol
- Times
- Zapf Dingbats

For more information, see “Customizing System Folders” in Chapter 2 of this manual.

The contents of the `.mac` directory

The `.mac` directory, which is in the user's home directory, contains a directory named after the workstation on which it is installed.

IMPORTANT The files and directories in the `.mac` directory are critical to the correct operation of MAE. Do not move or modify these files (except for `.MIV_file` as described in Chapter 2 of this manual).

The `.mac` directory needs to be at the root of a MIV (*/MIV*/`.mac` where */MIV* represents the pathname of the MIV). By default, the `.mac` directory is in the user's home directory.

The `.mac/`*`hostname`* directory contains the following:

- a `.DMlock` file used by MAE to maintain the integrity of the `.mac/`*hostname* directory.
- a `.fs_cache` file that contains a database of Macintosh file information.
- a `.fs_info` file that contains a database of Macintosh volume information.
- a `.MIV_file` that contains definitions of the independent volumes for your system.
- a `FileIDs` file that contains a database of Macintosh file information.
- the directory `LinkAliases`, which contains an alias for each UNIX-style symbolic link.
- the directory `MIVAliases`, which contains an alias for each independent volume. See Chapter 2 for information about independent volumes.
- a Macintosh parameter RAM file (`sm.vpram`) that contains configuration information concerning the monitor, the mouse, the keyboard, and other items.
- one additional directory for each independent volume. Such a directory has a numeric name and contains Finder-related items (such as the Desktop database files and Trash) required for that volume. See Chapter 2 for information about independent volumes.

Optional directory to specify a default `.mac` directory

MAE keeps versions of these items on a per user and per host basis because they contain pathnames or configurations that may differ among machines. However, some environments use the automounter to let users log in to any workstation and see the same environment. To support this practice with MAE, you need to create a directory named `default` in your `.mac` directory.

IMPORTANT When you use a default directory, all hosts using MAE must see the same file system layout, or users will experience unpredictable results.

How MAE works

This section briefly describes the mechanics by which MAE creates the virtual Macintosh that the user sees when running MAE. From the perspective of the UNIX operating system software that controls the user's workstation, MAE is an application program. Although the look and feel of MAE is in almost every respect identical with that of Mac OS, MAE depends on the presence of the UNIX operating system to mediate all interactions with the workstation's hardware.

MAE is launched when the user executes the `mae` command contained in the `apple/bin` directory, which is created by the MAE Installer. Launching MAE creates a process called `mae`, which will be treated like any other user process that the UNIX scheduler is responsible for managing. The `mae` process also launches an auxiliary daemon process named `macd`, which provides certain services to MAE that are best handled by an independent process.

MAE contains an implementation of the Macintosh Process Manager. This software manages the loading and unloading of all the Macintosh applications that the MAE user may execute. Memory space for use by the Process Manager and the Macintosh system is made available by the `mae` process at initialization. The default size of the memory region is 16 MB, and the user can expand it either by using the `memory` command line option to the `mae` command or by using the MAE Memory control panel from within MAE. This memory size is directly analogous to the physical memory that would be present in an actual Macintosh computer and determines the maximum size and number of Macintosh applications that MAE may run at a single time.

Although MAE is capable of running multiple Macintosh applications, with the ability to switch among them arbitrarily and even to have some of them perform processing in the background, all of this occurs within the single UNIX process called `mae`. Thus, the running Macintosh applications will not appear in a list of UNIX processes. While the standard UNIX features of process memory protection and preemptive multitasking guarantee the integrity of other UNIX processes running on the same workstation as MAE, the Macintosh applications running within MAE all share a common memory region and must adhere to the cooperative multitasking paradigm that is required of all Macintosh applications.

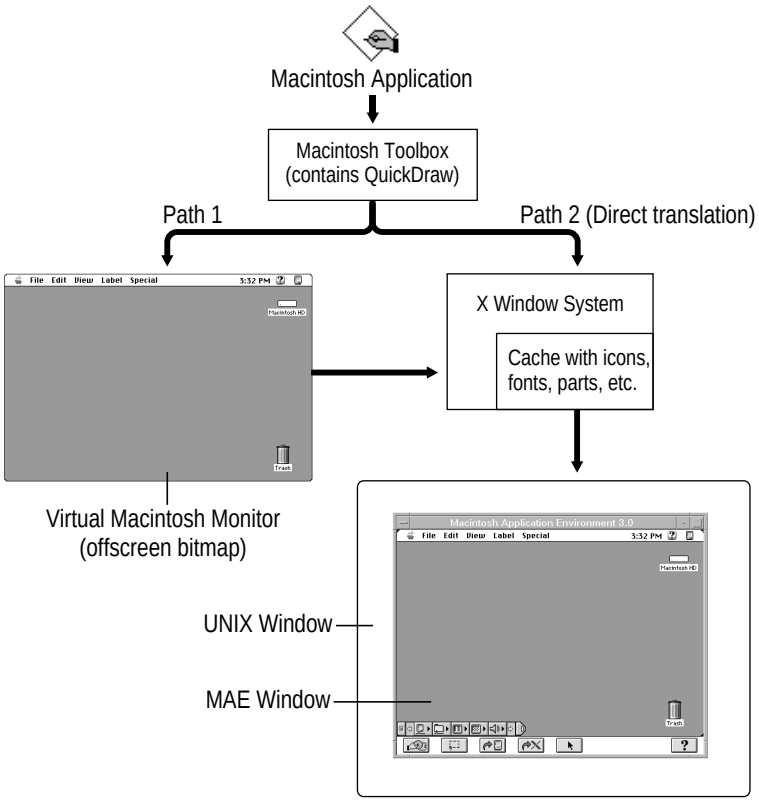
MAE can launch UNIX commands from the Finder. This differs from the launching of a Macintosh application in that a separate UNIX process is created independently of the `mae` process and will appear in a list of UNIX processes. If the UNIX command is an X window client, it creates one or more X windows that are also independent of the MAE window.

Graphics in MAE

Graphics in MAE differ from graphics on a Macintosh because MAE has no physical Macintosh-compatible video card or monitor. MAE has to convince applications that they are drawing to a Macintosh monitor, since Macintosh applications were not written to work with the X Window System.

To accomplish the hardware simulation, MAE takes advantage of the fact that virtually all Macintosh applications make calls to the Macintosh graphics standard (called *QuickDraw*) for their graphic operations. The system creates a virtual Macintosh monitor—a frame buffer—and whenever an application makes a QuickDraw call, QuickDraw updates the virtual monitor and tells MAE which part of the bitmap changed.

In previous versions of MAE, the entire frame buffer was passed to the X Window System to send the graphics to the X Server. In version 3.0, however, rather than passing the frame buffer to the X Window System, MAE converts the changes to appropriate `xlib` calls and sends these calls to the server (Path 2 in the following diagram). The frame buffer is referenced only in extraordinary cases, such as when a window is only partially visible (Path 1 in the diagram). This approach means that graphic displays in MAE version 3.0 require only ten percent of the resources required by the same function in version 2.0.



Note: A few Macintosh applications (especially games) may attempt to draw directly to video RAM on a Macintosh, usually for performance reasons. These applications might not work in MAE because the X display might not be updated consistently when using these programs.

Color flash

The term *color flash* refers to a display of unexpected colors and a screen flash that can occur when the user moves the pointer among X Window System applications with different color tables. Color flash occurs when the hardware cannot support the number of colors that must be displayed concurrently, as sometimes happens when you run an X application that uses its own color table. (A *color table* is an indexed lookup table used to display colors on a computer screen.) It is unlikely that the user will notice any significant color flash unless he or she is using a color-intensive application.

Many X window systems can have only one color table active (that is, actually installed into the hardware color table) at a time. Applications that use a color table other than the one installed display the unexpected colors characteristic of color flash.

MAE includes the “Approximate (less color flash)” option in the Mac OS Colors section of the MAE General control panel that helps reduce color flash. See “Using the MAE General Control Panel” in Chapter 9 of the *MAE User’s Manual*. When this option is on, MAE attempts to keep the color definitions in its own color table in synchronization with the *default color table* that is shared by many standard X clients.

There may be a small performance degradation when the Approximate option is on. (See “Graphics Buffer” in Chapter 4 for more information.)

Note: The default colors that MAE uses are not the standard Macintosh colors, because some of them come from the colors in the default color table. When a Macintosh application changes the colors used by MAE, you might still see color flash.

When the “Exact (more color flash)” option is selected, the default colors used by MAE are the same as those on a Macintosh.

4 Resources Required by MAE

This chapter describes the memory and disk resources that MAE uses to perform its tasks. The chapter discusses MAE's architecture and its use of the following:

- memory
- caches
- disk space

Understanding memory usage with MAE

This section describes the role various parts of MAE play in MAE's use of memory. The significant MAE components in terms of memory use are as follows:

- the MAE executable, `apple/bin/mae`, other native UNIX code, and related data
- `apple/lib/engine`, including the Mac OS ROM (Toolbox) code, the scaffolding (or glue) code, and related data
- the Mac OS system code, application code, and data
- the MAE X Window graphics buffer

The following table summarizes memory requirements for these components.

Usage	Amount (megabytes)
Native code and data	4.25
680x0 scaffolding code and data	1.5
Mac OS ROM code and data	2
Mac OS system code, application code, and data	16 (default)
Graphics buffer	0.3 (or more)
Total	24 (approximately)

Note: The 16 MB default for Mac OS system code, application code, and associated data is intended to provide reasonable performance. You can use as little as 8 MB if memory is more critical than performance.

Native code and data

The native UNIX programs consist primarily of the MAE executable (`apple/mae`), the Macintosh environment daemon (`macd`), and other daemons (`AppleMAE` and `lmgrd`, for example). Together with their associated data, these programs require approximately 4.25 MB of memory.

`apple/lib/engine`: the 680x0 scaffolding and Mac OS ROM code

The MAE engine (`apple/lib/engine`) includes the 680x0 scaffolding code (also called glue code) that serves to map Mac OS operations to UNIX system calls as appropriate, as well as the Macintosh Toolbox code normally stored in the Mac OS ROMs on Macintosh computers. This code together requires approximately 3.5 MB of memory.

The engine and its data are mapped into memory by means of the UNIX system call `mmap`(2). On platforms that support copy-on-write, the native code and data, the Toolbox code and data, and the Mac OS ROMs should mostly be shared between multiple processes running MAE. (*Copy-on-write* means that an individual UNIX process that modifies any of the memory space listed above will get a private copy of the page [usually about 4K] containing the changed memory location.)

Mac OS system and application code

The Macintosh Application Environment size value in the Memory control panel specifies the amount of memory that can be used by the Mac OS system code (including the Finder, extensions, and similar items), Mac OS application code, and associated data. The user can adjust this size in the Memory control panel or on the MAE command line with the `-memory` option. The default MAE size is 16 MB, but the minimum can be as low as 8 MB, depending on your needs. Factors to consider in setting this size include the number and size of System Folder additions (such as extensions and control panels) and the number and size of concurrently running applications.

Graphics buffer

The size of the graphics buffer depends on the size of the MAE window and the type of screen (maximum color bit depth) on which it is running. The approximate equation used to estimate the buffer size is

(*window_width* * *window_height* * *max_bit_depth*) / 8 = buffer size (in bytes)

For the default Macintosh screen (640 x 480) on an 8-bit color system, this comes to approximately 300K. Maximizing MAE window size on a 1024 x 768 screen with a bit depth of 8 yields a result of about 1 MB.

Note: Users can change the maximum bit depth used with a command argument (`-maxdepth` *number_of_bits*) or an X resource (`Mae.maximumDepth`).

The MAE code sometimes needs to process the graphics buffer. This can cause MAE to allocate a second graphics buffer of the same size in the following cases:

- The “Exact (more color flash)” option (in the Mac OS Colors section of the MAE General control panel) is set, and pixel 0 in the X display server’s default colormap (see `xcmap`(1X)) is black. To reduce color flash, MAE creates another buffer to do color mapping between the colors specified within the Mac OS environment and the colors in the current X colormap. This is caused by a restriction in the Mac OS that requires black and white to be in fixed positions.

  In this case, you can eliminate the need for a second buffer by choosing the “Approximate (less color flash)” option.

- The number of colors selected in the Monitors control panel does not match the number of colors available in an X visual supported by the underlying hardware (see `xdpyinfo`(1X)). In this case, MAE needs an extra buffer to remove or add bits to translate between what the Macintosh needs and what X supports.

  In this case, you may eliminate the need for a second graphics buffer by selecting in the Monitors control panel a bit depth that is directly supported by the underlying hardware. You may also want to set the maximum bit depth (`-maxdepth` command-line option) to this value as well.

- Direct pixels require explicit RGB values, but different machines require the RGB information in different orders (for example, BGR). If the underlying hardware does not match the Macintosh standard (RGB), MAE will allocate an extra buffer to do this mapping.

  In this case, the only way to avoid the requirement for a second buffer is to display MAE on an X server that matches the Macintosh RGB pixel order.

Swap space

Because UNIX swaps processes, it is important to have enough physical memory and swap space to handle MAE. Depending on the platform's swap-allocation scheme, be sure that each copy of MAE has swap space at least equal to or greater than (and preferably twice) the total memory requirement listed in the table at the beginning of this section.

- **To find out how much swap space is available on HP-UX, enter the following command line:**

  ```
  /etc/swapinfo
  ```

- **To find out how much swap space is available on Solaris, enter the following command line:**

  ```
  /etc/swap -l
  ```

  MAE and other applications display error messages if they cannot find sufficient virtual memory to run. Increasing the swap space may alleviate this problem.

Physical memory

There must be enough physical memory available to keep the *resident memory set* in physical memory. The resident memory set for a process is the number of pages the operating system keeps in physical memory (as opposed to swapped out) in order to run that process. UNIX usually increases the resident memory set for a process if the process is paging in a large number of pages for its time slice.

To determine how much physical memory MAE needs on an HP system:

1. **Run the `top` command.**
2. **Look at the RES column.**

To determine how much physical memory MAE needs on Sun workstations:

1. **Run the following command line:**

   ```
   /usr/ucb/ps -l
   ```

2 **Look at the RSSZ column in the output.**

MAE requires at least 9 MB of physical memory for its resident set when the process's total virtual size (as listed in the table at the beginning of this section) is the base amount (13 MB).

Check the paging activity on your system by running `vmstat`(1M) with an argument of 5 while MAE is running. The columns of interest will be

- `re` (reclaim)
- `at` (HP address translation)
- `mf` (Sun minor faults)
- `pi` (page ins)
- `po` (page outs)
- `de` (page shortfall)

Run `vmstat`(1M) with the `-s` flag to provide the `si` (swap in) and `so` (swap out) numbers. If these numbers are high, you may need more physical memory to run MAE efficiently on your machine.

Some workstations with 128 MB of memory have shown a ten percent improvement over those with 64 MB.

UNIX I/O caches

Another important determinant of how well any application will run on your system is how your operating system handles I/O. Most UNIX operating systems dedicate some fixed amount or percentage of physical memory towards caching recently used pages that have been read from or written to a file system.

If you are performing activities that require a significant amount of I/O on the same files, it is important to have enough physical memory to cache these files. To check whether you have enough memory for these tasks, run `vmstat`(1m), as described in "Physical Memory," earlier in this chapter, while running I/O-intensive applications under MAE.

Adding physical memory to your workstation may increase MAE's performance somewhat, because the additional memory can be used for I/O caching.

CPU caches

Most newer machines contain caches to reduce the latency from the CPU to main memory. When the CPU fetches an instruction to execute, or when a CPU executes an instruction that accesses memory, the cache logic first looks for that instruction datum in the cache. Finding the datum in the cache is usually faster than accessing physical memory.

However, if the instruction isn't in the cache, the cache logic must fetch it from physical memory—a slower process. The cache logic then moves the instruction or datum into the cache. Thus, the next time the instruction is needed, accessing it will be faster.

Terminology used for caches

Caches come in various sizes and types. To clarify the discussion of MAE's effect on caches, some of the terms related to caches are defined here:

- **Cache conflict** occurs when two frequently used memory locations consistently force each other out of the cache. Items in memory that are consecutive (and whose total size is smaller than the cache size) do not conflict. Two code fragments in memory have to be separated by some factor of the cache size or set size to conflict. See "Set size," later in this list.
- **Cache hit** occurs when the system finds an item in the cache. This is a fast process. Cache use is based on the assumption that programs use the same data over and over again. For best performance, cache hit rates should be high.
- **Cache line** refers to the group of memory locations that are moved into the cache when even a single element of that line is read or written. This is done to take advantage of the fact that many memory accesses are consecutive. Typical cache line sizes are 8 to 64 bytes per line.
- **Cache miss** occurs when the system does not find an item in the cache and has to go to the next level of cache or memory to locate the item. This is usually a slow process and should happen infrequently for best performance.
- **Cache tag** marks a cache line with the address, or address plus process ID, that uniquely identifies where that datum came from in memory.
- **Data cache** contains recently used copies of non-instruction memory locations accessed in a running program.
- **Directly mapped cache** uses the bottom bits of a physical memory address to create an index into the cache. If the tag at that cache location does not match the physical memory address, the cache logic retrieves the line containing the memory location requested from the next level of cache or from memory, replacing the existing cache entry (see "Cache line").
- **External cache** is not on the CPU chip, but usually is on the board containing the CPU. External caches might be slower to access than internal caches (see "Internal cache") and usually require expensive chips for speed. However, the size of external caches often can be larger than that of internal caches because there is more room on the board than on the CPU chip.
- **Instruction cache** contains copies of instructions recently executed by a running program.

- **Internal cache** is cache on the CPU chip. Internal cache is faster to access but limited in size to what can fit on the CPU chip.
- **Multilevel cache** refers to multiple caches, usually set up in some hierarchical fashion, between the CPU and memory. Multilevel caches are frequently used when the CPU chip cannot accommodate a big enough primary cache and a secondary cache is needed (see “Secondary cache”) to reduce access time between the on-chip cache and memory.
- **Primary cache** is the first cache the CPU goes to for a memory access. If the item is not found, the CPU or cache logic will try the secondary cache, if present (see “Secondary cache”). If the item is still not found, cache logic looks in memory.
- **Secondary cache** is usually located between the primary cache and memory. It is usually an external cache (see “Primary cache” and “External cache”). The secondary cache can usually be made up of slower chips than a solitary external cache (that is, where there is no internal primary cache), because the fast primary cache, not the CPU, accesses it directly.
- **Separate cache** In this architecture, instructions and data are kept in separate caches. If you run out of space in one cache, you cannot use space from the other cache. Separate caches make it easier to schedule instruction usage of the caches.
- **Set associative cache** Caches may be *single set associative* or *multiple set associative*. Single set associative caches allow lines from memory to exist in only one cache location (as determined by the virtual or physical address); multiple set associative caches partition a cache into *N* pieces and allow lines to appear in *N* places in the cache, with the location usually picked by an LRU (Least Recently Used) algorithm.
- **Set size** is the size of a particular partition in a set associative cache.
- **Unified cache** In a unified cache, both instructions and data are in the same cache and therefore compete for the same space. Unified caches provide flexibility for running an eclectic set of programs in which some of the programs may demand more instruction cache and some more data cache. It is difficult to schedule instructions with a unified cache because the data might conflict with the instructions you are executing.
- **Virtually mapped cache** uses the bottom bits of a virtual memory address to index into the cache location corresponding to that virtual memory location. If the tag does not match the virtual memory address, the cache logic retrieves the line containing the memory location requested from the next level of cache or memory.

Many machines use variations on the cache types listed here. Consult with specific vendors to find out cache configurations on specific machines and their implications for performance.

How MAE uses caches

MAE uses caches heavily and functions best on a machine with a large cache.

Cache sizes and configurations have a significant impact on the performance of all machines. A larger cache can improve performance by a factor of two or more.

MAE utilizes both instruction caches and data caches heavily, as is shown in the following lists:

Instructions:

- native code to support 680x0 scaffolding and dynamic compilation
- native code to emulate instructions
- dynamically compiled code to emulate instructions

Data:

- native data to support 680x0 scaffolding and dynamic compilation
- dispatch table that maps 680x0 opcodes to native emulation code
- 680x0 instructions that MAE emulates
- 680x0 data for Mac OS system and Toolbox code and scaffolding code
- frame buffers for graphics

One of the heaviest users of the cache is MAE's emulator dispatch table, which is indexed by the 16-bit 680x0 opcode (a 0.25 MB table). Maximum usage of opcodes is 10 to 20 percent of the total opcode space; unfortunately, this usage is sparse (that is, the opcodes used are not consecutive, nor even close together). MAE uses only one 32-bit entry per opcode, and chances are high that other opcodes nearby are not being used. On machines that have large cache line sizes (see the list of definitions in the previous section, "Terminology Used for Caches"), this sparse use causes a number of never-to-be-used memory locations to be loaded into the cache. Caches are less useful if they contain data that are never used.

MAE attempts to optimize the system's cache use. Cache use is optimized for one cache size for a particular target platform, even though cache sizes may differ between models. The goal is to pick a size (usually the smallest one reasonable) that has the best possible return on all models of that target platform.

Some of the optimization techniques are less effective on machines whose caches are multiple set associative. However, other benefits of multiple set associative caches should compensate somewhat for this limitation.

MAE cannot optimize its cache use as effectively on machines that have unified caches (with combined instruction and data) because data may conflict with instructions. However, the problems posed by a unified cache may be offset by its larger size.

On machines with multilevel caches, MAE tries to optimize its use of the primary instruction cache, which can help a secondary cache indirectly as well. However, the secondary cache is usually a unified cache and is less amenable to optimization.

The optimization of cache use also depends heavily on some repeatable and regular mapping from virtual memory pages to cache pages. This is not a problem on machines with pure virtually mapped caches. On machines with directly mapped caches, which use the physical address to determine where memory is placed in the cache, there is a question of how the operating system allocates physical pages to virtual pages. MAE knows the locations of instructions and data in virtual memory, but never knows which physical page will be allocated to a virtual page by the operating system.

Fortunately, for reproducibility reasons most operating systems use some of the virtual address to determine which physical page to allocate for the corresponding virtual page. If the operating system used a totally random choice for choosing physical pages, workstation benchmark results would not be reproducible because cache conflicts would change between runs as the virtual-to-physical mapping changed. Therefore MAE's cache optimization should still show some benefit on machines with directly mapped caches.

Components of MAE

MAE installs into three directories and optionally adds kernel modifications for networking and floppy disk support. All the nonkernel files are installed into an `apple` directory that is usually placed into the installing user's home directory (although another directory can be chosen during the installation process). MAE also creates a System Folder and `.mac` directory in the home directory of any additional users when they first run MAE. The System Folder includes all the standard Macintosh System Folder items; the `.mac` directory contains host-dependent information used by MAE.

Several users may share a single `apple` directory. Users can either set up a symbolic link (named `apple`) to the `apple` directory from their home directories or use a MAE command-line argument to specify where the `apple` directory is located:

`-appledir` *pathname*

To minimize system requirements, each copy of MAE running on a workstation uses shared resources in the `apple` directory. Consequently, each user must always have access to the `apple` directory.

Note: You can use the Get Info command to determine disk storage requirements. The Get Info command in MAE differs from the standard Mac OS version in that its totals include statistics for any invisible files in a folder.

MAE contains a number of packages. Upon installation, users can choose which packages they want to install. The following is the list of packages:

Package	Size (approximate)
AppleTalk	5 MB
Online documents	2 MB
ClarisWorks demo	6 MB
Claris Impact demo	2 MB
ClarisDraw demo	2 MB
Claris FileMaker Pro demo	8 MB

Note: The disk space required will vary from system to system, depending on the minimum disk block size. Most UNIX operating systems do not allow more than one file to share the same disk block. Since some of the packages contain many small files, these packages may take up more space on the file system.

The minimum you can install and still run MAE is the MAE package itself.

Users can share the `apple` directory but not the System Folder and `.mac` directories. The System Folder contains the following directories and files:

File or Directory	Minimum Initial Size in 1K Blocks	Description
`.SMlock`, `%.SMlock`	2	Lock MAE to prevent multiple users of a System Folder
Apple Menu Items	206	Apple Menu applications
Control Panels	978	Tools for configuring MAE (and other functions)
Control Strip Modules	41	Tools for configuring MAE (and other functions) from the desktop
Extensions	1100	Mac OS extensions, such as drivers
Extensions (Disabled)	588	Mac OS extensions that are disabled by default at startup
Finder	36	Macintosh desktop manager
Fonts	68	Macintosh fonts
MAE Enabler	19	MAE specific Mac OS code
MacTCP DNR	19	Domain name resolver for MacTCP
Preferences	69	Application-specific preferences
Scrapbook File	71	Data file for the Scrapbook desk accessory
Shutdown Check	1	Indicates whether previous MAE session was terminated normally
Shutdown Items	1	Items to run when MAE shuts down
Startup Items	1	Items to run when MAE starts up
System	538	Mac OS
Total	3171	(Grows with use)

The remainder of this chapter describes some of the larger consumers of initial peruser space listed above and some options to reduce this cost over a number of users.

The .mac directory

The .mac directory contains a directory for each host that executes MAE. This directory contains the following items:

File or Directory	Minimum Initial Size in 1K Blocks	Description
.alias	2	Ensures your AliasList file gets updated properly
.DMlock, %.DMlock	2	Prevents multiple users of the directory
.fs_cache, %.fs_cache	2	Cache/database of Mac info + btree
.fs_info, %.fs_info	2	Part of %.fs_cache
.MIV_file	1	Stores static MIV definitions
Desktop DB	26	Icons and application location database
Desktop DF	232	Part of Desktop DB
Desktop Folder	3	Items on the desktop
FileIDs	1	Maps pathnames to Mac file IDs
LinkAliases	1	Maps pathnames to Mac aliases
MIVAliases	1	Maps pathnames to independent volumes
Trash	1	Items in the Trash
sm.vpram	1	Macintosh PRAM

MAE keeps versions of these items on a per user and per host basis because they contain pathnames or configurations that may differ among machines.

Other overhead considerations

In addition to the initial disk space costs, users may incur incremental costs when they perform the following tasks:

- add fonts
- add printer description files
- drag many items to the Trash
- modify system resources
- install applications (these may add to the size of the user's System Folder)
- open folders with the Finder

The first two items are fairly straightforward: added font files and printer descriptions use disk space.

This is also true if you drag items to the Trash; MAE keeps a copy of the item in the Trash until the Trash is emptied (with the Empty Trash command).

The other items on the list are discussed in the following paragraphs.

Modifying system resources

System resources (such as monitor configuration) are stored in compressed form in the shared `apple` directory. However, if you modify one (such as changing the bit depth in the Monitors control panel), an uncompressed, unshared local copy of that resource appears in your System Folder.

Installing applications

Many application installers add items to the System Folder (for example, a file in the Preferences folder or an item in the Apple menu to launch the application). This cost in disk space varies from application to application.

Opening folders with the Finder

When users open folders with the Finder, MAE makes entries for the `.fs_cache` file in the `.mac` directory. Thus, this file grows as MAE is used. The average space cost is the number of bytes in the full pathname plus 110 bytes for each additional file or directory seen by MAE. The number of bytes is an average because internal data structures require varying amounts of overhead space, depending on the number of entries the system is currently tracking.

The information saved in the Desktop DB, Desktop DF, `.fs_cache`, and `.fs_info` files and in files used to maintain independent volumes allows MAE to access files quickly. The files also provide a place for MAE to save Macintosh-specific information for UNIX files (for example, the placement of a file icon within a folder or the default view for a folder). If the files grow too large, users can use the `-rebuild` argument on the MAE command line to clean up these files and minimize their size.

5 AppleTalk Administration

This chapter describes AppleTalk for MAE. It includes information on the following topics:

- an overview of the AppleTalk protocol suite
- installing AppleTalk on workstations running MAE
- using AppleTalk from within MAE
- using UNIX AppleTalk commands
- troubleshooting
- reference books for further information on AppleTalk

Overview

AppleTalk, a network system designed and developed by Apple Computer, provides an easy way to share network resources such as printers, file servers, and groupware services.

AppleTalk for MAE is an implementation of the AppleTalk version 56 protocol specification, based on streams, which makes AppleTalk functionality available to Macintosh applications and system software running under MAE. It is compatible with other versions of AppleTalk Phase 2.

This manual assumes familiarity with the AppleTalk networking system and streams. See the "References" section at the end of this chapter for more information.

Implementation

AppleTalk for MAE maintains the Macintosh Toolbox Application Programming Interface to AppleTalk routines, as described in *Inside Macintosh: Networking.*

AppleTalk consists of a number of protocols that correspond roughly to the layers of the OSI model. AppleTalk for MAE uses streams to configure the following AppleTalk protocols into the UNIX kernel:

- AppleTalk Address Resolution Protocol (AARP)
- AppleTalk Data Stream Protocol (ADSP)
- AppleTalk Echo Protocol (AEP)
- AppleTalk Transaction Protocol (ATP)
- Datagram Delivery Protocol (DDP)
- EtherTalk Link Access Protocol (ELAP)
- Name Binding Protocol (NBP)
- Routing Table Maintenance Protocol (RTMP)
- Zone Information Protocol (ZIP)

The remaining protocols are implemented as Macintosh-style drivers and extensions:

- AppleTalk Filing Protocol (AFP)
- AppleTalk Session Protocol (ASP)
- Printer Access Protocol (PAP)

AppleTalk for MAE implements the version 56 specification of AppleTalk, which does not include support for some extensions found in later versions, such as these:

- Apple Open Collaboration Environment (AOCE)
- AppleTalk Remote Access
- AppleTalk SNMP MIB

AppleTalk for MAE maintains compatibility with all versions of AppleTalk that conform to the AppleTalk Phase 2 specification (see *Inside AppleTalk, Second Edition)*. Workstations using AppleTalk for MAE can interact with any device that uses AppleTalk Phase 2.

AppleTalk for MAE does not support the use of AppleTalk outside the Macintosh Application Environment. There is no API available to UNIX processes for accessing AppleTalk services. Applications that rely on the availability of both Macintosh and UNIX APIs, such as AppleShare Pro, will not run on MAE.

AppleTalk for MAE cannot be used if a third-party AppleTalk stack is installed on the host workstation, and AppleTalk services will not be available to MAE running on a machine that does not have AppleTalk kernel modules for MAE installed.

Getting started with AppleTalk for MAE

AppleTalk for MAE implements AppleTalk network services on supported workstations that are equipped with appropriate interfaces and cabling. In order to provide these services, you need to install the AppleTalk Kernel Modules package with the MAE Installer. See Chapter 1 of the *MAE User's Manual* for installation instructions.

Hardware requirements

AppleTalk for MAE requires a workstation capable of running MAE, with an Ethernet interface and appropriate cabling.

Supported workstation platforms include Hewlett-Packard 9000 series 700 workstations and Sun SPARC workstations running the appropriate version of Solaris (see "Software Requirements," next).

AppleTalk for MAE utilizes the Ethernet interfaces and cabling of its host workstations. It uses standard Ethernet interfaces connected via thick, thin, or twisted-pair Ethernet cable. No additional hardware is necessary if an Ethernet network is already in place.

Other interfaces and cabling systems, such as token ring or LocalTalk, are not supported.

Software requirements

AppleTalk for MAE works with Solaris 2.5 or later or HP-UX 9.05 or 10.10.

AppleTalk requires the standard streams utilities for communications protocols. Streams must be installed before AppleTalk can be used. MAE will function without streams, but AppleTalk will be unavailable.

STREAMS is part of the standard MAE Solaris distribution. STREAMS is also included in the standard HP-UX 10.10 distribution. Users of HP-UX versions prior to 10.10 must install STREAMS/UX separately or AppleTalk for MAE will not function.

Installing AppleTalk for MAE

The AppleTalk kernel modules for MAE must be installed with the MAE Installer. You need superuser privileges to install the kernel modules.

Any preexisting AppleTalk protocol stack must be removed before installing AppleTalk for MAE. See the product documentation for instructions.

Once the MAE Installer has reconfigured the kernel, you need to reboot the machine.

What the MAE Installer does on HP-UX 10.10

The MAE Installer adds two drivers and three STREAMS/UX modules to the HP-UX kernel by installing the `libappletalk.a` library into the `/usr/conf/lib` directory on HP-UX 10.10 systems. AppleTalk functionality is added according to standard HP-UX kernel modification guidelines.

The MAE Installer alters `/usr/conf/master.d/streams` by making the additions shown below.

In the area reserved for third party drivers in the DEVICE section, the Installer adds the following lines:

```
*
*  AppleTalk streams module for MAE - libappletalk.a
*  40 is streams module, 21 is driver
*
adsp     adspinfo       40  20FC  -1  -1
at_atp   atp_info       40  20FC  -1  -1
at_sig   at_sig_info    40  20FC  -1  -1
ddp      ddpinfo        21  20FC  -1   38
elap     elap_info      21  20FC  -1   39
```

In the driver handle area of the DRIVER_LIBRARY section, the Installer adds the following:

```
ddp             libappletalk.a
elap            libappletalk.a
```

In the SUBSYSTEMS_DEFINE section, the Installer adds the following:

```
ddp
elap
```

In the LIBRARY section, the Installer adds the following:

```
libappletalk.a  0
```

The MAE Installer inserts the following lines into /stand/system:

```
adsp
at_atp
at_sig
ddp
elap
```

Then the MAE Installer creates the following device files:

```
/dev/appletalk/ddp/socket
/dev/appletalk/lap/ethertalk0/control
/dev/elap
/dev/ddp
```

Once it has modified the configuration files, the MAE Installer runs the HP-UX /usr/sbin/mk_kernel command to build a new HP-UX kernel in the /stand directory. Before the MAE Installer moves the new kernel to /stand/vmunix, it preserves the old kernel in /stand/vmunix.save.

What the MAE Installer does on HP-UX 9.0x

The MAE Installer adds two drivers and three STREAMS/UX modules to the HP-UX kernel by installing the libappletalk.a library into the /etc/conf directory. AppleTalk functionality is added according to standard HP-UX kernel modification guidelines.

The MAE Installer alters /etc/master and /etc/newconfig/master by making the additions shown next.

In the Third Party Driver area, the Installer adds the following lines:

```
*
*  AppleTalk streams module for MAE - libappletalk.a
*  40 is streams module, 21 is driver
*
adsp     adspinfo       40  20FC  -1  -1
at_atp   atp_info       40  20FC  -1  -1
at_sig   at_sig_info    40  20FC  -1  -1
ddp      ddpinfo        21  20FC  -1   135
elap     elap_info      21  20FC  -1   136
```

In the Driver Handle area, the Installer adds the following:

```
ddp             libappletalk.a
elap            libappletalk.a
```

In the Subsystem area, the Installer adds the following:

```
ddp
elap
```

In the Library area, the Installer adds the following:

```
libappletalk.a  0
```

The MAE Installer inserts the following lines into `/etc/conf/dfile` :

```
adsp
at_atp
at_sig
ddp
elap
```

Then the MAE Installer creates the following device files:

```
/dev/appletalk/ddp/socket
/dev/appletalk/lap/ethertalk0/control
/dev/elap
/dev/ddp
```

Once it has modified the configuration files, the MAE Installer runs the HP-UX `config` command and then executes `make` to build a new HP-UX kernel in the `/etc/conf` directory. Before the MAE Installer moves the new kernel to `/hp-ux`, it preserves the old kernel in `/hp-ux.save`.

What the MAE Installer does on Solaris

The MAE Installer takes advantage of the dynamic device-driver configuration on Solaris to install one driver and three streams modules into the Solaris kernel. The following five files are copied from the MAE Installation CD-ROM disk:

```
/usr/kernel/drv/atalk
/usr/kernel/drv/atalk.conf
/kernel/strmod/at_atp
/kernel/strmod/at_sig
/kernel/strmod/adsp
```

The MAE Installer adds two lines to the `/etc/devlink.tab` file:

```
type=ddi_pseudo;name=atalk;minor=elap appletalk/lap/ethertalk0/control
type=ddi_pseudo;name=atalk;minor=ddp appletalk/ddp/socket
```

The MAE Installer executes `add_drv` to add the `atalk` driver to the kernel; then it creates the following device files:

```
/dev/appletalk/ddp/socket
/devices/pseudo/atalk@0:ddp
/dev/appletalk/lap/ethertalk0/control
/devices/pseudo/atalk@0:elap
```

Finally, the MAE Installer changes the permissions of the following to 666:

```
/devices/pseudo/clone@0:le
```

The MAE Installer does not modify the Solaris kernel and does not archive a copy.

Verifying the presence of streams

To verify that streams is correctly installed on an HP-UX workstation, execute the command `strvf`, which performs a simple test and indicates success if streams is installed.

To ensure that streams is installed on a Solaris workstation, execute the `/usr/sbin/modinfo` command and use `grep` to search for `sad` to ensure that the `sad` driver is configured into the kernel.

Removing AppleTalk from the kernel

You can remove the AppleTalk Kernel Module package with the MAE Installer (see Chapter 1 of the *MAE User's Manual*). After removing the modules, reboot the workstation.

If the Installer is unable to remove the AppleTalk modules, remove them manually. To verify that the modules have been removed, do this:

- **On HP-UX workstations, enter the following:**

  ```
  /usr/bin/what/hp-ux | grep AppleTalk
  ```

  Doing so should return nothing if the AppleTalk modules have been successfully removed.

- **On Solaris workstations, enter the following:**

  ```
  /usr/sbin/modinfo | grep atalk
  ```

  Doing so should return nothing if the AppleTalk modules have been successfully removed.

Using AppleTalk for MAE

AppleTalk for MAE allows supported workstations to function as independent nodes on an AppleTalk network. All AppleTalk services function just as they do on a Macintosh computer.

Initializing the AppleTalk kernel stack

The AppleTalk kernel stack is implicitly constructed when MAE is first started on an AppleTalk-capable workstation, unless AppleTalk is explicitly disabled. Additional instances of MAE will construct the kernel stack only if it is not already constructed. This single AppleTalk kernel stack remains loaded until the final MAE process terminates.

Turning AppleTalk on or off

AppleTalk for MAE is turned on by default when MAE and the AppleTalk kernel modules are installed on a workstation and MAE is launched. If the modules are not present, AppleTalk will not be available.

AppleTalk can be turned on or off from within MAE by using the Macintosh Chooser. See the *MAE User's Manual*, Chapter 7.

AppleTalk state is maintained across invocations of MAE for each user. If a user shuts down a copy of MAE with AppleTalk inactive, AppleTalk will remain inactive until it is reactivated in the Chooser. It is possible to activate and deactivate AppleTalk within a single MAE session if AppleTalk was initially

active. If the session is begun with AppleTalk inactive, you need to activate AppleTalk in the Chooser and restart the session to access AppleTalk services.

Note: You can start AppleTalk with the `appletalk` command outside MAE. However, when you start MAE, you cannot change any of the settings in the Network control panel, including your zone and network device. Additionally, if AppleTalk is always active even when you are not running MAE, it generates unnecessary network traffic and uses system resources.

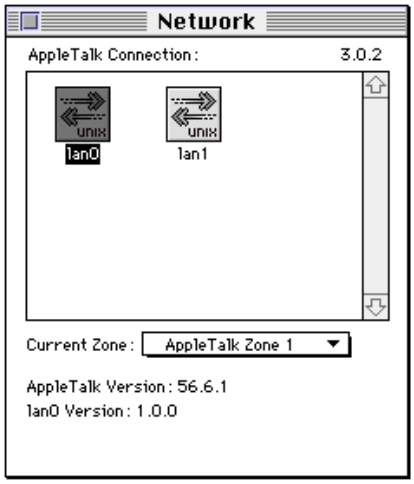
Accessing named resources

AppleTalk network resources appear to users as simple named entities. AppleTalk uses the Name Binding Protocol (NBP) to associate a triplet of object, type, and zone strings with an AppleTalk address (net, node, socket tuple). This binding comprises a Network Visible Entity (NVE).

Applications and system software dynamically register network services as NVEs. NVEs are used for access to AppleTalk resources, primarily through the Chooser. For example, selecting the AppleShare icon in the Chooser produces a list of all NVEs of type “AFP Server” in the selected zone. *Inside AppleTalk, Second Edition* explains the name-binding process in detail.

Selecting a network interface with the Network control panel

If several AppleTalk interfaces exist on the workstation, MAE will select a default interface when it is launched. To switch between two or more AppleTalk network interfaces on your MAE workstation, select the interface you want in the Network control panel.



For details, see “Changing the AppleTalk Network Interface” in Chapter 7 of the *MAE User’s Manual*.

Personal file sharing

The File Sharing extension included with MAE allows any MAE workstation with AppleTalk active to act as a personal file sharing server. Macintosh personal file sharing is an AppleTalk-based network file system that is separate and distinct from other network file systems, such as NFS. However, MAE's integration of UNIX file systems into the Macintosh Finder enables users to share both UNIX and HFS volumes and folders from their MAE workstations. These directories can be mounted by authorized AppleShare clients on the AppleTalk network.

> **WARNING** In most cases, a user *should not* share the root directory *without careful configuration of access privileges*. For example, sharing the root directory with read-only permission allows remote users to view files such as `/etc/passwd`. Users should be especially cautious when sharing folders on NFS volumes via file sharing. See the following sections for further information.

Controlling access to shared folders

Access privileges to a folder are established when the folder is shared. Personal file sharing controls access to shared folders with a database of users and groups maintained by the Users & Groups control panel. Each personal file sharing server has an individual database stored in the Preferences folder within the System Folder.

The user sharing a folder sets three different levels of remote access privileges—owner, user or group, and guests. This is done by using the Sharing dialog box that appears after you select Sharing in the File menu. The owner of the folder is the only one who can reset its permissions. Permission to move, rename, or delete the folder can also be restricted with the Sharing dialog box.

Note: It is best not to change the sharing status of a folder while remote clients are using it. If you want to change the file share permissions of a shared folder, use the File Sharing Monitor to ensure that no one is using the folder before changing its status.

Use of the folder is also restricted by the underlying UNIX permissions. Permissions established on the parent folder can be applied recursively to each folder contained within it. See Chapter 7 in the *MAE User's Manual*.

Using the /etc/mae_fileshare file to limit access

As the system administrator, you can use the file /etc/mae_fileshare to further limit access to files. For file sharing to work, this file must exist and must contain a valid form of one of the commands in the following table:

Command	Options	Description
none		Disable file sharing for the entire workstation.
owned	[-noguest]	Allow users to share only directories that they own.
any	[-noguest] [-noseedisk]	Allow users to share any directory.

The -noguest option prohibits users from allowing guest access to folders. The -noseedisk option prohibits users from allowing remote users to see the entire disk.

The default command is owned.

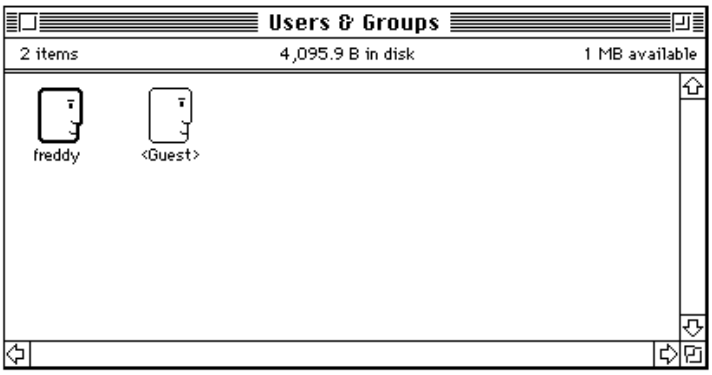
Note: To continue ensured security, you should leave the file permissions for /etc/mae_fileshare set to read-only.

Limiting the owner's access

The owner of the Macintosh environment is established by providing a name in the Sharing Setup control panel. This is the default owner of folders in that environment shared via personal file sharing.

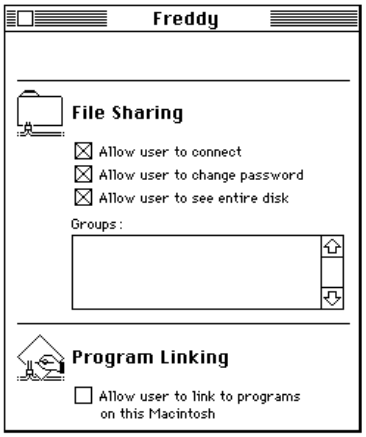
You can limit the owner's access by using the Users & Groups control panel. Do so as follows:

1 Open the Users & Groups control panel.



2 **Open the owner dialog box by double-clicking the owner icon (designated by a dark border).**

3 **Uncheck the checkbox labeled "Allow user to see entire disk."**



Determining permissions

MAE allows users to share UNIX file systems with file sharing. When a UNIX directory is shared via personal file sharing on MAE, it inherits both the UNIX and AppleShare permissions assigned to it.

The following table shows UNIX permissions compared to file sharing privileges:

If UNIX permissions are...		then file sharing privileges are		
Read	Write	Execute	See Folders and See Files	Make Changes
No	No	No	No	No
No	No	Yes	No	No
No	Yes	No	No	Yes
No	Yes	Yes	No	Yes
Yes	No	No	No	No
Yes	No	Yes	Yes	No
Yes	Yes	No	No	Yes
Yes	Yes	Yes	Yes	Yes

IMPORTANT When MAE users export part of a UNIX file system via file sharing, they can grant file sharing access only to the extent that the UNIX permissions allow them that access. For example, if a MAE user does not have write access to a file, the user cannot export that file with file sharing write access. Thus, the privileges another user has when accessing files are the most restrictive combination of the UNIX permissions and the file sharing privileges, including the restrictions imposed by the `/etc/mae_fileshare` file.

File sharing permissions, as described above, affect only shared directories mounted by a file sharing client. UNIX permissions apply to all files and directories.

Sharing UNIX directories

The seamless integration of UNIX file systems and NFS into the MAE Finder environment is extremely powerful and requires care on the part of the user when exporting folders.

> **WARNING** In most cases, a user *should not* share the root directory *without careful configuration of access privileges*. For example, sharing the root directory with read-only permission allows remote users to view files such as `/etc/passwd`. Users should be especially cautious when sharing folders on NFS volumes via file sharing.
>
> If a user shares a UNIX folder that contains an NFS mount point, the mount point is shared with effective permissions as described in "Determining Permissions," previously, *but without NFS access control*. Thus, *a remote user can mount a physical volume to which he or she has no NFS access* by mounting the volume via AppleShare once it is exported shared from MAE. Access to folders containing NFS mount points is controlled with the Users & Groups control panel and the Sharing dialog box.
>
> Although it is possible to allow access to the entire file system at the root directory, it is not recommended. You should use at least the default restrictions imposed by the `owned` command in the `/etc/mae_fileshare` file unless you have a specific reason for doing otherwise.

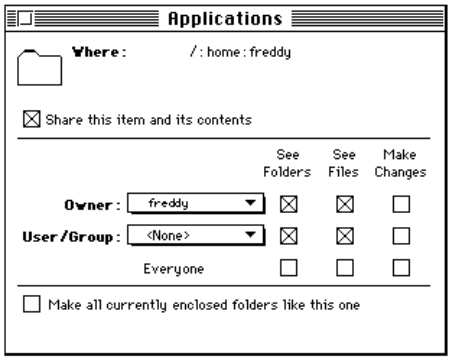
Sharing partitions with NFS mount points can degrade the performance of file sharing.

Turning off guest access

If you have modified the `/etc/mae_fileshare` file to allow guest access privileges, Macintosh personal file sharing allows the Macintosh owner to let guests use shared folders. You should turn off access for all folders not intended to be public. Do so as follows:

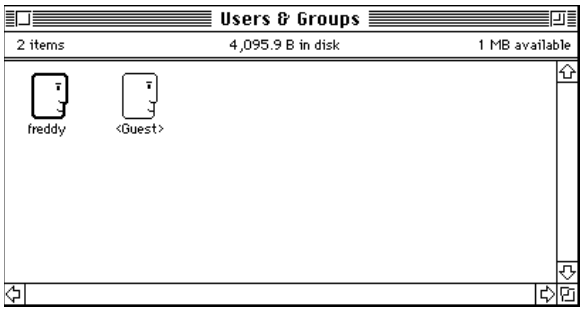
1 **Select the folder and choose Sharing from the File menu.**

2 Uncheck the Everyone checkboxes in the Sharing dialog box.

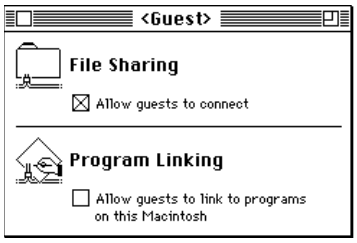


To disable Guest Access completely, you can edit the `/etc/mae_fileshare` file or perform the following steps:

1 Choose Control Panels from the Apple (🍎) menu and double-click the Users & Groups icon.



2 Double-click the <Guest> icon. The following window appears:



3 Uncheck the "Allow guests to connect" checkbox.

For further information on specifying file sharing users, groups, and privileges, see Chapter 7 of the *MAE User's Manual*.

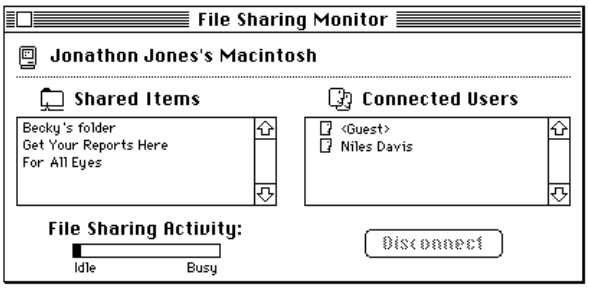
Sharing folders among several users

It is possible for several users to export the same file concurrently with file sharing.

> **WARNING** Although MAE does not automatically prevent several users from using file sharing concurrently, when several users export the same file concurrently through file sharing, users can write over other's changes.

Monitoring the use of file sharing

The File Sharing Monitor control panel displays a list of all currently connected users, a list of shared items, and a usage meter.



Use it to monitor file sharing activity and to disconnect users on demand.

Running several MAE sessions simultaneously

Users can run several MAE sessions simultaneously on a workstation. The first MAE session constructs an AppleTalk kernel stack that remains active until the final MAE session is shut down. Each session appears to the user as an independent Macintosh.

The sessions share the AppleTalk kernel stack and consequently have the same AppleTalk address and the same set of AppleTalk DDP sockets. These DDP (Datagram Delivery Protocol) sockets are allocated on a first-come first-served basis from a pool of 125 sockets. See "Implementation," earlier in this chapter, for information on DDP.

Since the AppleTalk zone and network interface are associated with the AppleTalk kernel stack, they cannot differ among concurrent MAE sessions. Also, a user cannot change the AppleTalk zone and network interface while another MAE session running AppleTalk is active.

When a single MAE session is active, you can use the Network control panel to select an AppleTalk zone and a network interface. You should set the zone and network interface before making the node available to several concurrent users.

Extensions

AppleTalk for MAE relies on the presence of several modified Macintosh extensions and control panels to provide AppleTalk network services. These extensions are different from those of Mac OS system releases and should not be replaced, modified, or upgraded. If these files are corrupted, they can be recreated by removing the System Folder and restarting MAE.

AppleTalk System Resources

Chooser. Provides access to named network resources; allows users to activate and deactivate AppleTalk.

Extensions

AppleShare. A Chooser extension that allows client-side access to AppleShare file servers.

LaserWriter8. A Chooser extension that enables Printing to AppleTalk LaserWriter Printers.

File Sharing Extension. An extension that implements file sharing for System 7.

MAE EtherTalk. A network cdev extension that implements AppleTalk over an Ethernet data link.

Network Extension. An extension that implements additional AppleTalk drivers.

Control Panels

Sharing Setup. Allows the user to set owner name, password, and machine name, and to disable or enable program linking and file sharing.

File Sharing Monitor. Monitors the use of file share volumes by remote users.

Users & Groups. Allows the Macintosh owner to create users and groups to control access to file share volumes.

Network. Allows the user to set his or her zone and select an AppleTalk interface.

UNIX AppleTalk commands

MAE's `appletalk` command is useful for determining and controlling the status of your AppleTalk environment from the UNIX command line. This command is located in the `apple` directory, which is created by the MAE installer.

The following sections describe some uses of the `appletalk` command and its options. For other options, see the `appletalk` man page.

Using `appletalk -s` to determine the status of AppleTalk

The `appletalk` command used with the `-s` option allows the user to determine the current status of AppleTalk.

If the AppleTalk kernel stack is active, the command

```
apple/bin/appletalk -s
```

returns information including the current node ID, network number, bridge address, and related information.

Using `appletalk -D` to turn off AppleTalk

The `appletalk` command used with the `-D` option allows you to shut down the AppleTalk kernel stack. This may be necessary if a MAE session is terminated improperly; for example, if a user terminates the `macd` process associated with a MAE session. An active AppleTalk kernel stack without MAE processes is not in itself a problem, but it generates minor network traffic, such as AppleTalk ARP packets.

If no MAE sessions are running but the AppleTalk kernel stack is still configured, you can shut down the AppleTalk kernel stack by running the following command:

```
apple/bin/appletalk -D
```

This command requires root access and should be used with caution; turning off AppleTalk while MAE sessions are running may cause a system crash.

If you shut down the AppleTalk kernel stack, you are prompted as follows to verify the shutdown:

```
One or more instances of Macintosh Application Environment
may be running.
Results of forcing down the AppleTalk stack are
unpredictable.
Would you like to continue [y/n]?
```

If you cancel the shutdown by entering `n`, no action is taken and the AppleTalk kernel stack continues running. If you verify the shutdown by entering `y` at the command line, then the AppleTalk stack is deconfigured. This is confirmed when the message `Continuing with shutdown` is displayed.

When the shutdown is complete, you may run `appletalk -s` to ensure that AppleTalk is no longer running.

Using `atlookup` to list the NVEs in a zone

The `atlookup` command returns a list of all NVEs (network visible entities) in a given zone. The syntax for this command is

`atlookup <`*object*`>:<`*type*`>@<`*zone*`>`

where *type* is, for example, a printer or server type, *object* is typically the hostname or other object name, and *zone* is the AppleTalk network zone. The `=` symbol is treated as a wildcard in the <*object*> and <*type*> fields and the `*` character represents the current zone. For example,

```
atlookup lilac:=@*
```

would return all NVEs registered with the name *lilac* in the current zone. See *Inside AppleTalk*, listed in the "References" section, later in this chapter.

Using `appleping` to ensure that a node is alive

The `appleping` command uses the AppleTalk Echo Protocol to ensure that a given node is alive. The format for this command is

`appleping <`*object*`>:<`*type*`>@<`*zone*`>`

These commands are located in the MAE `apple/bin` directory.

Troubleshooting

The following paragraphs provide help for common problems.

No AppleTalk Zone list in the Chooser

If the Chooser does not display a list of zones when MAE is used on a network with multiple zones, the AppleTalk kernel stack is not communicating properly with the router. Please check your router configuration and network interface.

Selecting AppleTalk Zones

Under some circumstances, a user of MAE may not be able to select a different default AppleTalk zone even though other zones are available. This can happen if two or more copies of MAE are executing on the same physical workstation; none of the MAE users will be able to change zones. This can also happen if AppleTalk has not shut down properly (it was still running after all copies of MAE were shut down). To ensure that AppleTalk is not currently running, execute the command `appletalk -D`.

Using network diagnostic tools

Standard AppleTalk diagnostic tools can be used to investigate the behavior of AppleTalk for MAE. However, most Macintosh-based tools that diagnose low-level hardware drivers cannot be run on MAE itself.

Reinstalling AppleTalk for MAE

If an AppleTalk streams module is corrupted or deleted, it is safest to shut down all MAE processes that may be executing and then remove and reinstall the AppleTalk Kernel Module package with the MAE Installer.

Restoring network extensions

Do not replace the network extensions that ship with MAE. These extensions are specially tailored for the MAE environment. (See "Extensions," earlier in this chapter, for a summary of these extensions.) If these extensions are removed, corrupted, or altered, you can restore them by removing the current System Folder and restarting MAE. This will cause the entire System Folder to be recreated. Before deleting the System Folder, be sure to save any modifications that have been made to it.

Some extensions, such as Chooser extensions and extensions installed by application installers, may be added without conflict.

An AppleTalk router comes up

If you are administering a MAE server and you change the network topology while the AppleTalk kernel stack is active, one of the following messages may appear:

```
An AppleTalk Router just came up.
This Node's address is in startup range.
Shutdown or Restart MAE to use full AppleTalk
functionality.
```

or

```
An AppleTalk Router just came up.
This Node's address is invalid for this cable.
Shutdown or Restart MAE to use full AppleTalk
functionality.
```

Quit all active MAE sessions, ensuring that the kernel stack is inactive (run `appletalk -D` and `appletalk -s`), and then restart the sessions.

(Sun only) File modes for non `-le0` interfaces

By default, device drivers used for LAN interfaces under Solaris are designated as read-only for the superuser (root). Because MAE lets any user start AppleTalk (which is normally done the first time MAE starts on each workstation), it is necessary that the file permissions be changed to readable and writeable by everyone. The MAE installation script automatically changes the permission for the device `le0`; all others need to be made readable and writeable by everyone before MAE will be able to use those devices.

(Sun only) Unloading AppleTalk stream modules on Solaris systems

On Solaris systems, the AppleTalk stream driver 'atalk' and three other AppleTalk stream modules, `at_atp`, `adsp`, and `at_sig`, are dynamically loaded into the system when MAE starts up. The driver and the modules will not be unloaded until the workstation is rebooted. If a `modunload` command is issued on any one of the stream modules, it will cause the system to crash.

(HP only) HP-UX 9.0.1 and STREAMS/UX

On HP-UX, MAE requires that STREAMS/UX be installed if you are planning to run AppleTalk. The MAE CD contains packages for both 9.0.3 and 9.0.5 of HP-UX. If you are running 9.0.1, you can use the STREAMS/UX package for 9.0.3. Please note that MAE does not support sound if used with 9.0.1.

(HP only) Booting HP-UX from the saved kernel

When you install MAE onto an HP workstation, MAE alters the kernel and saves the old HP-UX kernel in a file called `hp-ux.save`. If you need to reboot from the old kernel (for example, if the altered HP-UX kernel has become corrupted), you may do so either by copying `/hp-ux.save` to `/hp-ux`, and then rebooting, or rebooting from the `/hp-ux.save` kernel.

To reboot from the `/hp-ux.save` kernel, follow these instructions:

1 **Shut down the workstation.**

2 **Restart the workstation.**

When the power goes on, the console displays a message similar to the following:

```
Search for potential boot devices.
To terminate search, press and hold the ESCAPE key.
```

3 **Press and hold the (Esc) key until you see the following message:**

```
Selection process stopped.
Searching for potential boot devices.
```

After a moment, the console displays a list of SCSI device options from which to boot.

4 **Choose the appropriate boot device.**

For example, if SCSI device 6 is the appropriate option, enter the following:

```
b scsi.6.0 ipl
```

This will cause an ISL prompt (`ISL>`) to be displayed.

5 **At the ISL prompt, enter the following:**

```
hpux boot disc(;0)/hp-ux.save
```

This will result in booting the alternate kernel in `/hp-ux.save`.

References

System 7 Macintosh Networking Reference

This is a document published by Apple Computer, Inc. It provides a user guide to the file sharing features of AppleTalk that are discussed in this chapter and in Chapter 7 of the *MAE User's Manual.*

Part number 030-3936-A

Inside Macintosh: Networking

This book describes key concepts of networking the Macintosh with other computers. It describes in detail the components and organization of AppleTalk, how to select an AppleTalk protocol, and how to write software that uses AppleTalk networking protocols.

ISBN 0-201-62269-6

Inside Macintosh: Interapplication Communication

This book provides a complete technical reference to the Apple Event Manager, the AppleScript component, the Program-to-Program Communications Toolbox, and the Data Access Manager.

ISBN 0-201-62200-9

Inside AppleTalk, Second Edition

This book provides an in-depth discussion of the protocol architecture of the AppleTalk network system. The Second Edition features completely updated, detailed descriptions of the AppleTalk protocols, including the enhancements of AppleTalk Phase 2, and is the essential developer tool and programmer's compendium of AppleTalk protocols.

ISBN 0-201-55021-0

Planning and Managing AppleTalk Networks

Topics covered in this book include the role of the network administrator, an overview of the AppleTalk network system, network services such as file sharing and electronic mail, AppleTalk network connections to industry-standard environments such as Ethernet and token ring, network media and topologies, guidelines for designing a network, factors affecting installation, security considerations, network maintenance guidelines, and troubleshooting tools, strategies, and scenarios.

ISBN 0-201-52345-0

6 The Sound Manager

This chapter describes the implementation of Macintosh sound capabilities on MAE. In addition, it offers solutions and workarounds to problems that might arise on Sun and Hewlett-Packard workstations.

Using the MAE Sound Manager

The Sound Manager in MAE acts transparently, much like the Sound Manager on the Macintosh. However, there are some setup issues and concerns for the different platforms. In addition, MAE is constrained by the host resources, which add performance issues.

MAE sets the default sampling rate to 44.1kHz (the same rate as audio CDs). The MAE Sound control panel allows you to set the sampling rate to whatever the hardware allows. For best sound quality, use the default sampling rate of 44.1kHz.

Note: On low-performance workstations, you may need to lower the sampling rate if you experience pauses or gaps in sound output.

When executing multiple instances of MAE on a single workstation, only the first MAE instance will be able to use sound.

Note: MAE does not support stereo or 16-bit sound.

Hewlett-Packard workstations

The Hewlett-Packard sound implementation uses the Audio Server that is available on all sound-enabled HP platforms. On HP-UX 9.*x* systems, you need to update the Audio Server with HP patch PHSS_4341 to make sound work correctly. You do not need a patch for HP-UX 10.*x* systems. See the Read Me file (`README.TXT`) on the MAE CD-ROM for the latest information on patches.

Sound is not supported in HP-UX releases earlier than 9.03.

To determine if the patch has been installed on your HP-UX 9.*x* system, check for the file `/etc/filesets/PHSS_4341`. If this file is present, it is likely that the patch has been installed.

On both HP-UX 9.*x* and 10.*x* systems, follow the instructions in the `aserver`(1M) and `audio`(5) man pages for setting up and testing your Audio Server. Once you have set up your Audio Server, sound should work correctly.

HP audio will work only locally on the host Audio Server, by default. To enable a remote Audio Server, set the `AUDIO` environment variable to the Audio Server to which you want to send sound output.

There are a number of useful environment variables for sound.

- `AUDIO` specifies the Audio Server to send data to; for example, set the `AUDIO` environment variable to <*hostname*>`:0`.
- `SPEAKER` specifies destination for the sound. Set the `SPEAKER` variable to `INTERNAL` (the default) to use the internal speaker. Set it to `EXTERNAL` to send the sound to an output jack.

Troubleshooting for HP workstations

A list of potential problems and their solutions follows.

- **Problem:** Audio Server has been updated but no sound is produced.

  Solution 1: Audio security may be enabled. Refer to the man page for the `asecure` command to enable sound output. The default configuration for the audio security is to disable sound output for all users. The `asecure` program enables access for any or all users of the target machine.

  If the user is not properly authorized to use the Audio Server, MAE issues an error message during startup. Although you cannot use sound, MAE functions normally. You can instruct MAE to skip the check for the Audio Server. To do this, set the `TB_SOUND_OVERRIDE_CHECK` environment variable to 1. This allows MAE to produce sound (although error messages are still produced during startup). The environment variable does not override the security features of the server, but it does allow MAE to produce sound. For more information, consult your HP documentation for `asecure` and the Audio Server.

  Solution 2: (Only works for HP-UX 9.*x* systems) Increase the system volume. Use the program `/usr/audio/bin/acontrol` and increase the playback volume.

- **Problem:** MAE is running remotely, but no sound is produced.

  Solution: If the display server is not equipped with an Audio Server (that is, it is not an HP workstation), you cannot get sound on it. If it is, before launching MAE set the `AUDIO` environment variable to the name of your display (that is, export `AUDIO=`<*hostname*>`:0`) on the machine that will run the MAE process.

- **Problem:** MAE sound hesitates; there are intermittent silent portions in the sound output.

  Solution: Run MAE on a faster machine or turn down the sampling rate in the Sound control panel. The higher the resolution of the sound, the more resources are necessary to produce the sound. Reduce the resources necessary by lowering the sampling rate.

Sun workstations

If the Sun workstation is sound capable, MAE sound will work with it. If the user who is running MAE is different from the one logged into the console, you must change the permissions of the audio devices. To do so, first list the audio device files with the following command:

```
ls -la /dev/audio
```

Then follow the symbolic links (there are three or four levels) to find the actual device. As superuser, you need to change the permissions of the audio device to read/write access in order to enable sound.

Note: You must change the permissions for both `/dev/audio` and `/dev/audioctl` for sound to operate.

Troubleshooting for Sun workstations

- **Problem:** MAE does not produce sound.

  Solution 1: Make sure you are logged into the console or have the permissions on `/dev/audio` and `/dev/audioctl` set correctly for your user. See "Sun Workstations," above, for details about permissions on added devices.

  Solution 2: Press the mute button on the speaker box (if available). If the speaker box is muted, no sound is available.

  Solution 3: Turn up the volume on the speaker box or in the Sound control panel.

- **Problem:** MAE sound hesitates.

  Solution: Run MAE on a faster machine or turn down the sampling rate in the Sound control panel. The higher the resolution of the sound, the more resources are necessary to produce the sound. Reduce the resources necessary by lowering the sampling rate.

- **Problem:** The speaker box mute button on older Sun workstations has no effect.

  Solution: Use the MAE Sound control panel to adjust the volume.

- **Problem:** The speaker box volume control on older Sun workstations has no effect.

  Solution: Use the MAE Sound control panel to adjust the volume.

- **Problem:** Other programs cannot generate sound using `/dev/audio`.

  Solution: Start MAE with the `-nosound` option.

7 General Troubleshooting

This chapter lists situations that you might encounter when using MAE 3.0 and describes how to respond. This chapter contains hints and suggestions for the following topics:

- the MAE desktop
- file systems
- display
- applications
- MAE independent volumes
- installing applications
- keyboard
- performance

Other chapters in the MAE documentation contain troubleshooting information specific to the chapter. You can find other troubleshooting sections in the following locations:

- Chapter 2, “Starting and Quitting MAE,” in the *MAE User Manual*.
- Chapter 3, “Installing Macintosh Applications,” in the *MAE User Manual*.
- Chapter 4, “Printing in MAE,” in the *MAE User Manual*.
- Chapter 6, “Using CD-ROM Discs and Floppy Disks,” in the *MAE User Manual*.
- Chapter 5, “AppleTalk Administration,” in this manual.

Desktop

This section describes situations you might encounter when in the Desktop environment.

Can’t empty Trash

If you drag a folder into the Trash that contains two items with the same name but different capitalization (for example `File1` and `file1`), MAE cannot empty it. This is because MAE is case insensitive and cannot see items with the same name but different case. Because MAE cannot see all the contents, it can’t empty the Trash.

To empty the trash, open the Trash (double-click the Trash icon) and select the folder. With the folder selected, choose Put Away from the File menu. This returns the item to its original location. Then, use UNIX commands to delete the files.

You cannot move “file name” to the Trash, because a disk error occurred

If you encounter this error, the Trash needs more space to complete the operation. To solve the problem, do one of the following:

- Place your `.macdir` in a UNIX file system with more free space.
- Free some space on the UNIX file system where the `.mac` directory resides.
- Drag a smaller amount of data to the trash.

Generic icons appear for files of installed applications

If you install an application on a volume that is not mounted at startup, the documents for these application may have a generic icon. (This occurs often if you use a remove NFS server with a set of common applications.) To ensure the correct icons appear, include an entry for the remote file system in the `.MIV_file`. See Chapter 2 to learn how to create this entry.

Display

This section describes situations you might encounter with the display and graphics.

Adjusting mouse movements

If your mouse pointer does not move smoothly, you can use the X command `xset` to fine-tune mouse action. With the `xset` command, set the mouse acceleration anywhere from 15 to 20 (by default, the acceleration is 4). See the `xset` man page for more information about the command.

Full-screen MAE displays

To arrange for full-screen MAE display without a window manager running, use the `-geometry` option specifying the screen size. Under a window manager, this may not produce the desired effect because of borders, title bars, and such. For this case, it is best to not specify a geometry and let the window manager set the geometry by using the Maximize button or window menu option. To get the full Macintosh experience under HP VUE, set the X resource `Vuewm*mae*clientDecoration` to none and start MAE with the geometry set to the full screen size.

Resizing the Finder and application windows

A popular feature of MAE is the ability to dynamically resize the X11 window (and thus, the Macintosh desktop) while it is active. When resizing occurs, MAE sends an Apple event to all active applications indicating that the desktop size has changed. Some older applications do not support this Apple event and will continue to behave as though the desktop were the same size as when the application was launched.

Flashing menu bar after exporting to X

If text is selected from an open application, such as Note Pad, and exported to X (using the Export to X key from the MAE Toolbar), the Finder menu bar may flash or disappear when returning focus to MAE. If this happens, clicking the Macintosh desktop within MAE will restore the menu bar.

Cursor disappears

If you are running MAE in a monochromatic environment, the cursor may sometimes disappear. For example, this happens when a black cursor moves over a black area of the screen.

If your X server supports color cursors, you can set the foreground and background colors of the MAE cursor with the TBFGCURSORCOLOR and TBBGCURSORCOLOR environment variables, as in the following examples.

```
setenv TBFGCURSORCOLOR red
setenv TBBGCURSORCOLOR red
```

Screen fades to black

Some applications take over the entire colormap and set everything to black to perform a fade. To prevent this problem, start MAE with the `-installcmap` option.

Changing the X modifier map

If you make any changes to the X modifier mapping while MAE is running, you lose those changes when you switch to MAE. To make any changes to the X modifier map, shut down MAE, make the changes, and then start MAE again. MAE only saves external X map changes during startup.

Modifier map sharing between MAE and X applications

Every keystroke that occurs in an X environment is inspected by X so that it can catch the keystrokes that invoke X operations. To accomplish this, the X environment uses a *modifier map*—a keyboard translation table.

While the X window containing MAE is the active window, MAE modifies this keyboard map, so that keystrokes meaningful to MAE are passed to MAE by the X environment for processing, even if they would normally have meaning to the X environment itself. This means that while MAE is active, keystrokes are processed as specified in the MAE Keyboard control panel. For example, if the [META] key is mapped to the MAE [Command] key, the X environment will pass on to MAE those [META] keystrokes that occur while MAE has focus. Typing Meta-Q in the MAE environment will therefore quit a Macintosh application even though Meta-Q would normally be interpreted by the X environment as a CLIENT_KILL (which would terminate MAE).

As soon as MAE is no longer the active window, the original modifier map is restored and all keystrokes function as defined for the X environment. This restoration, however, is not reliable under all circumstances. For example, pushing a MAE session over the network to a server running X11R6 can result in unpredictable results.

One solution to this problem is to disable MAE's ability to change the modifier keys. This can be done either by setting an environment variable

```
set TBNOMODMAP=1
```

or by starting MAE with the `-nomodmap` option.

IMPORTANT Be careful if you choose to disable the modifier mapping; keystrokes intended for MAE may be trapped by the X Window Manager with unexpected results.

MAE independent volumes

This section describes common problems and solutions for MAE independent volumes.

Problems with icons and windows with newly mounted MIV volumes

If icons in a newly mounted MIV window disappear, close the window and reopen it. If this does not restore the icons, restart MAE.

This occurs when a folder is in transition from a folder to a MIV alias.

If you open a folder that has just been mounted and before MAE has completed converting it into a MIV alias, you will see the contents of the window until the folder is converted. Then, the window closes (because the status of the folder changes). To see the contents, double-click the newly created MIV icon for the folder.

Exceeded maximum number of mounted volumes (MIVs)

You can have up to 50 mounted volumes. If you attempt to mount another the following message appears:

```
You have exceeded the maximum number of mounted volumes.
The mount for 'volumename' has failed.
Unmount unneeded volumes by dragging their icons to the trash.
```

To get around this limit, unmount unneeded volumes by dragging them to the Trash and attempt to mount the MIV again or shutdown MAE and remove some static MIV definitions from the `.MIV_file`.

"The disk *name* cannot be used, because the folder cannot be found" message

If you have several MIVs mounted and you turn on file sharing, you may see the message

```
The disk name cannot be used, because the folder cannot be found.
```

You get this message if the `.mac` folder is not at the root of a MIV (*/MIV*/`.mac` where */MIV* represents the pathname of the MIV). (By default, the `.mac` directory is in the user's home directory.) To correct this, move the `.mac` folder to the root of a MIV.

"One or more items could not be shared" error message

If you attempt to share a volume and you see the message:

```
One or more items could not be shared, because not all
items are available for file sharing.
```

Wait a minute and retry the operation. If you continue to see the error message, add the volume to the `.MIV_file`.

File systems

This section describes situations you might encounter when using file systems.

"Too many open files" error

HP systems allocate file descriptors on a per workstation basis, instead of per user. Therefore, if more than one user is running MAE from the system, they might use all of the available file descriptors. If this happens you see the following message:

```
You have exceeded the maximum number of open files on
this workstation.

This is a serious error.

MAE may not be able to function if it cannot open
critical system files.

Shut down MAE immediately.
```

To increase the default number of file descriptors, log in as the root user and start the System Administration Manager (`sam`). Select Kernel Configuration and then select Configurable Parameters. In this section, you can increase the total number of available file descriptors by changing the value of the NFILE variable. (Each user needs about 375 file descriptors.)

Network file system performance

When a user's System Folder or Apple directory (`appledir`) resides on a network file system, performance will be affected by the configuration and by the loads on the file servers involved. In some cases, this affects MAE behavior. For example, to avoid waiting forever because of a communication failure with an NFS host, MAE periodically reviews ongoing file operations and cancels any file system operation that loses communication for a specified amount of time (called the "File System Timeout"). By default, this timeout is ten seconds. This means that if MAE waits for a response from an NFS-mounted file system for more than the timeout, MAE gives up on the access

and returns failure. If the operation is non-critical (such as a directory listing), MAE treats that file system as inaccessible. If the access is critical to MAE's operation (such as the Finder), then MAE hangs.

In configurations with heavy network traffic (resulting in slow network response), MAE's default timeout setting may be too short. If so, you may see an error such as "… because the disk '/' cannot be found." Using the `-fstimeout` option at startup increases the amount of time that MAE waits before giving up on a remotely mounted file system and helps to prevent MAE from hanging. For example, you can start MAE with the command `mae -fstimeout 30`, which sets the timeout to 30 seconds. This value should be sufficiently high for networks where response time is variable due to heavy loads. If `-fstimeout` is set to 0, the heartbeat function is disabled and MAE waits indefinitely for a remote file system.

If network response time is still affecting the performance of MAE after you increase the value of `-fstimeout`, consider putting the System Folder or Apple directory on a local volume.

Accessing files and directories with a space

You can reference files and directories with a space by using a backslash (\) before the space. For example,

```
ls -l System\ Folder
```

Alternatively, you can enclose the entire name in quotes (either single quotes or double quotes as long as they match). For example,

```
ls -l "System Folder"
```

Permission problems

If you are having problems accessing files after the permissions have been modified to grant you access, you can either remove the System Folder and install a new one or rebuild the desktop by starting MAE with the `-rebuild` option.

Using automounter (HP only)

If you use automount to access another user's home directory, the directory might not appear where you expect. In UNIX, the mounted home directory would appear under `/home/`username. However, the automounter mounts the directory at `/tmp_mnt/home/`username. MAE only sees the directory at `/tmp_mnt`.

Similarly, if you are using automounter and want to install MAE in an automounted directory, you need to specify the beginning of the path in `/tmp_mnt`. For example, specify the path `/tmp_mnt/net/homer/apple` instead of `/net/homer/apple`.

Sharing `apple` directories across systems

Although you can share Macintosh files and applications between Sun and HP workstations, you cannot share `apple` directories.

Optional directory to specify a default `.mac` directory

MAE keeps versions of these items on a per user and per host basis because they contain pathnames or configurations that may differ among machines. However, some environments use the automounter to let users log in to any workstation and see the same environment. To support this practice with MAE, you need to create a directory named `default` in your `.mac` directory.

IMPORTANT When you use a default directory, all hosts using MAE must see the same file system layout, or users will experience unpredictable results.

Using Fetch

Using Fetch to "get" and "put" non-text Macintosh files from or to UNIX file systems requires special care to ensure the files are transferred in a usable state by MAE (preserving their AppleSingle format). When doing a "get" operation of a non-text Macintosh file, always select binary transfer mode. When doing a "put" operation of a non-text Macintosh file, always select AppleSingle as the file format when prompted by the dialog box. Macintosh and UNIX text files should always be transferred by selecting text (ASCII) transfer mode.

For complete instructions about using Fetch, read the online help for Fetch.

Note: Apple Computer has a license to use Fetch with MAE. You do not need to pay the shareware fee.

Problems with aliases

Aliases not created during initial startup

If MAE crashes during initial startup while MAE is creating (or updating) a new System Folder, important aliases might not be created that are created at the end of the installation process. Many aliases are put in the System Folder that point to applications in the `apple` directory, and you will experience performance problems.

If this happens, remove the newly created System Folder and restart MAE. MAE creates a new System Folder for you. If you are updating MAE, you can use your backup System Folder.

Using CellNames with MAE

If you use an AFS file system, you can add remote cells to your `/afs` folder with the CellNamesMAE file. You can choose any of three locations for this file:

- `apple/CellNamesMAE`. This is the default location. If you use this location, any system that shares this `apple` directory can access the cells.
- `/usr/vice/etc/CellNameMAE`. Use this location to use cells on your local system.
- `.mac/`*hostname*`/CellNameMAE`. This location lets the CellNamesMAE file be specific to a user, regardless of their location.

MAE hangs after an NFS server shuts down

If a server for a UNIX hard mount point shuts down, MAE hangs. To solve this problem, restart the server.

Space for `/users` on HP Series 700 workstations

You might need to increase the capacity of the `/users` file system to provide space for a MAE System Folder for each user. If you need to create new file system to increase the space, the new file system must support long file names.

The process below copies the contents of `/users` to a new MAE-compatible file system:

1 **Log on as superuser.**

2 **Perform one of the following steps:**

If `/users` is a mounted file system, create a new mount point `/users.temp` and remount `/users` on `/users.temp`.

If `/users` is in the root file system, change the name of `/users` to `/users.temp` by using the command

```
mv /users /users.temp
```

3 **To create a new file system, start SAM (System Administration Manager) by typing the command**

```
sam &
```

Use SAM to perform these steps:

4 **Select the Disks & File Systems topic and then CD-ROM, Floppy and Hard Disks.**

5 **Select the new SCSI Drive, highlight Action, and select "Add a Hard Disk."**

6 **Select Modify Defaults and select Use Long File Names.**

Note: The Use Long File Names option results in a file system that supports file names longer than 14 characters. The SAM default supports file names only up to 14 characters and prevents MAE from properly creating long Macintosh file names on that file system.

7 **Select OK and mount the new File system to `/users`.**

8 **Exit SAM.**

9 **Copy the contents of `/users.temp` to the new file system.**

To do so, use the following commands:

```
cd /users.temp
find . -print -depth | cpio -pdumv /users
```

10 **Verify that the contents of `/users` are the same as the contents of `/users.temp`.**

11 **If `/users.temp` is a mounted file system, unmount it. If `/users.temp` is in the root file system, remove it with the command**

```
rm -rf /users.temp
```

Installing applications

If the installation program does not complete, copy the contents of the installation floppy disks to your hard drive and run the installation from your hard disk.

IMPORTANT If you copy the files to your system, use Mac OS operations to transfer the files. If you use UNIX commands, the file modification dates may not be preserved. Some installers require these dates to be unmodified before they will install an application.

Alternatively, you can copy the files from a Macintosh computer using floppy disks or a networked file server.

Note: MAE requires applications to be 32-bit clean.

If you have difficulty installing or using an application, try turning off dynamic recompilation. To turn off dynamic recompilation, start MAE with the `-mx` command option.

Installation of incompatible system extensions

MAE relies on the presence of several modified Macintosh extensions and control panels. These extensions are different from those of Mac OS system releases and are installed differently in MAE to minimize disk space consumption. These extensions should not be replaced, modified, or upgraded except by the MAE installer. If these files are corrupted, they can be recreated by removing the System Folder and restarting MAE. In general, you should not use any system software installers or system software enablers that are not provided with MAE. For further information, see “Extensions” in Chapter 5 of this manual.

The installation programs for some applications add extensions to the System Folder that may be incompatible with MAE. For example, HyperCard 2.3 installs the Speech Manager extension.

If you experience problems after installing an application, start MAE with the `-noextensions` option and remove any incompatible extensions or control panels.

Installing PPDs

If you are installing software that installs a PPD, the installation will fail. PPDs are stored in the `apple/shared` folder (with an alias in the `sys/extensions` folder). This folder is read-only so you will not be able to install new PPDs unless you have write permission for the apple directory.

Keyboard

This section describes problems you may encounter when using the keyboard and recommended actions.

Shift-Option key sequence on IBM keyboards

To perform a Shift-Option key sequence, press the Option key (Ctrl/Act) before the Shift key. If you press the Shift key first, the results may be incorrect.

To correct this situation, use the MAE Keyboard control panel to set the Option key to a key other than Ctrl/Act.

Performance

Starting up MAE takes a long time

If MAE takes close to five minutes between the loading of the extensions to the launching of Finder but then behaves correctly, the `appledir` might have been moved to another directory with the `mv` command.

Once the `appledir` has been moved, the aliases inside the System Folder, such as MAE Guide, cannot be resolved.

To solve this problem, remove or replace all aliases that point to items in your apple directory or remove your System Folder and have MAE build a new one for you.

Appendix The License Manager

MAE supports floating (network) licensing, which allows all the authorized users at your site to share licenses up to the number you have purchased. This licensing approach allows your end users to make most efficient use of the licenses your company has purchased, while easing your license administration tasks. This chapter describes the components of the license manager and discusses the following aspects of administering MAE licenses:

- setting up and configuring license management
- selecting server nodes
- obtaining license files
- determining where to put license management components
- combining MAE license management with existing FLEXlm license management

Most of the information in this chapter is for those administrators who want to take advantage of multiple redundant servers or who use the FLEXlm license manager software and want to consolidate license management tasks. The section “License Manager Quick Start” provides instructions for those administrators who prefer to start using their MAE licenses immediately.

License manager quick start

If your main goal is to start using MAE immediately, you do not need most of this chapter. To enable license management, perform the following steps:

1 **Copy your license file, `license.dat`, into the `apple` directory on each workstation on which you will run MAE.**

2 **Start the license manager daemon, `lmgrd`, with a command such as the following:**

```
/opt/apple/bin/lmgrd -c /opt/apple/license.dat
```

(The `-c` option must be used to specify the location of the license file.)

This quick start approach is not the most efficient approach to license management for workgroups. At a minimum, you should eventually consider using a single network-mounted copy of the license file, and you should modify your `rc` files to start the license manager daemon automatically.

Overview

MAE uses a client/server model to manage licenses in a way that maximizes flexibility in the way you use your licenses and minimizes downtime due to equipment or network problems. The MAE process and the daemon processes (which together compose the license server) can run on separate nodes on your network.

Note: The MAE license manager uses FLEXlm, a popular license manager used to control the use of UNIX-based software products. If you are familiar with FLEXlm administration practices, and your site is using revision 4.1 or later, you can easily incorporate MAE licenses into your existing license management practices.

License management components

The MAE process works with three main license management components to provide flexible, efficient licensing:

- the MAE daemon, `AppleMAE`
- the license manager daemon, `lmgrd`
- the license file, `license.dat`

This section describes these three components and how they work together with the MAE process.

The AppleMAE Daemon

The AppleMAE daemon (also known as the vendor daemon) keeps track of how many licenses are checked out and who is using them. It communicates with the MAE process and the license manager daemon to ensure maximum availability of unused licenses. If the AppleMAE daemon terminates for any reason, all users lose their licenses, regaining them automatically when AppleMAE restarts.

The License Manager Daemon (`lmgrd`)

The license manager daemon (`lmgrd`) supervises the floating licenses for MAE on your network. It starts (and, if necessary, restarts) the AppleMAE daemon. It also supervises connections between the MAE process and the AppleMAE daemon.

You can have multiple redundant license manager daemons, each on a different server node, ensuring that licenses are available if any two out of the three server nodes are running. With redundant servers, you don't have to split up your licenses among multiple servers, or rely on any one machine.

The License File

License data is stored in a license file, `license.dat`. You must have a current, valid license file to run MAE. (See your MAE product packaging or the Read Me file `README.TXT` for instructions on obtaining your license file.) The license file is a text file that contains information about the server nodes and the AppleMAE daemon as well as some encrypted data that makes your licenses valid for your site.

In some environments, the licensing information for MAE can be combined with that of other applications into a single license file. See "Combining License Files," later in this chapter.

The license request process

When a MAE process starts on any workstation, the following steps occur:

1. **MAE locates the license file and uses it to determine the host name of the license server node and the port number of the license manager daemon, `lmgrd`.**
2. **MAE establishes a connection with `lmgrd` and requests communication with an AppleMAE daemon.**
3. **`lmgrd` determines the appropriate AppleMAE daemon and sends its host name and port information back to MAE.**
4. **MAE establishes a connection with the AppleMAE daemon and sends its request for a license.**

5 **The AppleMAE daemon determines whether any licenses are available and returns a message granting or denying use of a license.**

6 **If a license is available, MAE completes its startup process. If a license is not currently available, MAE displays a message to that effect and terminates.**

Configuring license management

As the administrator, you can specify the following aspects of MAE's license management:

- the location of the license file
- the location of all executable programs and daemons
- the location of all log files
- the TCP/IP port number used by the license manager process, `lmgrd`
- assigning licenses to users, nodes, and groups

Setting up license management

As the administrator, you are responsible for setting up licensing on your system or network. If you do not have multiple licenses for MAE, or you don't want to share licenses, you may want to use the quick start procedure at the beginning of this chapter.

If you have multiple licenses that you want users to be able to share, you must set up at least one license server to manage those licenses. If you use a license server, you will have to configure the MAE license management system for your site by performing the following general steps, each described in greater detail later in this chapter:

1 **Select your license server nodes.**

2 **Determine the host IDs of your license servers.**

3 **Obtain a license file for your site.**

You will need to provide the hostid for each server.

4 **If you want to integrate MAE licenses with your existing FLEXlm 4.1 or later license management system, you'll need to incorporate the license file information appropriately.**

See "Combining License Files," later in this chapter.

5 **Determine if you want an options file and, if so, set it up.**

6 **Determine the best location for the license management files and put them there.**

7 **Start `lmgrd`.**

Selecting license server nodes

Before running MAE using floating licenses, you will need to set up your license server node (or nodes). You must first select which node or nodes to run your license servers on. In selecting your server nodes, consider the following factors:

- resources required by each server
- costs and risks of using diskless nodes and remotely mounted disks
- whether you need to run multiple (possibly redundant) servers

Each of these factors is detailed in this section.

Note: The license server software (the `lmgrd` and `AppleMAE` daemons) provided with MAE can run only on a computer running Solaris, HP-UX 9, or HP-UX 10.

Resources required by the server

When you select a server node, consider the system limits on resources required by the server. This section presents some of those requirements. In a site with fewer than 100 licenses, few of these requirements should pose a problem on any supported workstation.

Processes

When you run `lmgrd`, it automatically starts up one copy of the AppleMAE daemon and any other vendor daemon specified in a DAEMON line in the license file. Each time a Transmission Control Protocol (TCP) client such as MAE connects to the server, it uses a process file descriptor. If the maximum number of file descriptors is exceeded, additional copies of AppleMAE (or other vendor daemons) will be started to accommodate the overflow requests. In practice, this is a rare occurrence even on large networks. See your operating system's documentation to determine what the maximum number of file descriptors is for your server computer.

Sockets

When using TCP, a single vendor daemon can support as many users as the system limit for file descriptors—for example, 256 (on Sun OS 4.x) to 4000 (on OSF/1). When files are exhausted, additional vendor daemons are spawned to allow for extra file descriptors. When using UDP, there is no limit to the number of end users per vendor daemon process because they can share a single socket in the vendor daemon.

Each MAE client connected to a license server uses one socket. The total number of sockets used by the license server programs is slightly larger than the total number of simultaneous clients.

If you have a large number of licenses (two hundred or more), you should ensure that the system limit on the number of sockets and file descriptors is adequate to handle all of the licenses.

CPU time

License management in MAE uses minimal server time. The MAE client only communicates with the server once during startup to obtain a license, once during shutdown to return the license, and once every five minutes to inform the server that it is still active.

Disk space

The only output file created by the license server is the log file. The log file contains one line for each license checkout or checkin transaction. If you have a lot of license activity, the log file can grow very large. You need to consider where to put this file and how often to delete or shorten it. If disk space is at a premium, you can choose to overwrite the log file each time the license manager is started (using the + parameter with the REPORTLOG option), or to turn off logging of these transactions using the NOLOG option.

Alternatively, you can trade out the log file at certain intervals while the daemons are running, by piping the output of `lmgrd` to a shell script that appends each line to the file, as follows:

Start `lmgrd` with this command:

```
% lmgrd | sh -c 'while read line; do echo "$line" >> logfile
; done'
```

You can then rename the file *logfile* at the desired interval, and a new log file will automatically be created. Another approach is to make *logfile* a symbolic link and change the value of the link to trade the log file.

Note: The log file should be a local file on the server machine.

Memory

The license daemons use little memory. Typically, `lmgrd` uses approximately 100K and the vendor daemons use approximately 120K each.

Network bandwidth

The license management system sends relatively small amounts of data across the network. Each transaction message, such as for a checkout or checkin, typically generates a data transfer of less than one kilobyte. For small numbers of clients, licensing can be effectively run over slow networks, even with dial-up SLIP lines.

For sites with hundreds of client applications that communicate frequently with their vendor daemon, the network bandwidth use may start to become significant. In this case you should run the client and server on the same local area network or on a high-speed wide area network.

Note: In high-traffic networks you may want to place a copy of the license file in a file system local to the application rather than reading the license file data from `lmgrd` by setting `LM_LICENSE_FILE` to a *port@host* address.

Remote mounted disks and diskless nodes

Keeping `lmgrd`, the vendor daemons (such as `AppleMAE`), the license file `license.dat`, and the log file all on locally mounted disks ensures that licenses will be available as long as the server is running. Storing any of these files on a remotely mounted disk doubles the points of failure that could lead to a loss of all of your licenses; the loss of either the license server machine or the file server machine will make the licenses unavailable.

Using a diskless node as a server is an obvious example of this problem. In addition, some licensed applications make security checks that fail on a diskless node. (If you find that you are having problems with a locked file, one possibility is that you are attempting to run on a diskless node.)

Multiple servers

You can split your licenses among multiple independent servers. This approach allows licenses administered by each server to be available as long as that server is running.

Alternatively, you can set up redundant license servers to operate as a single logical license server. This approach requires more administration, but helps to ensure the availability of all your licenses in the event that a server becomes unavailable for any reason.

To set up redundant servers, you must first use the host IDs for the three server workstations to obtain a license file with three SERVER lines. (See your MAE package or the Read Me file `README.TXT` for details.) You then need to make sure that each license server has `lmgrd`, `AppleMAE` (and any other vendor daemons), and the license file `license.dat` on a local file system. You then start `lmgrd` on each license server.

Note: In a redundant server configuration, a majority of the servers listed in the license file must be available before any licenses can be used. In other words, if only one server in a three server configuration is running, then no licenses will be available.

To simplify administration, it is best to use one FLEXlm server (optionally in redundant mode using peer servers) per site. This setup provides the most flexibility as you only have to allocate keys to a floating site.

Only use multiple FLEXlm servers per site if you have completely separate administrative domains. In this case, you can treat the domains like separate sites.

Determining license server hostids

To obtain a valid license file you must provide the hostids of those machines that you intend to use as license servers. You can get the hostid of the server machine by running the utility `lmhostid` on that machine. If you haven't yet installed MAE, you can get the host ID of your machine by using the appropriate command for your system from the following table. MAE supports HP and Sun workstations as license servers; commands for other systems are provided for your convenience.

Hardware Platform	Host ID	Type this command on the license server	Example
Apollo	20-bit ID	`lcnode -me`	BE70
Convex	32-bit hostid	"getsysinfo"	
Decstation	ethernet address	`pfstat` or `arp hostname`	080020005532
DG aviion	32-bit hostid	`/usr/sbin/systemid`	0C020972
HP	32-bit hostid	`echo 'uname -i' 16o p | dc`	778DA450
	ethernet address	`lanscan` (use station address without leading "0x")	0000F0050185
	"id module"	read ID typed on module, remove 'A', and convert remainder to hex	118CD098
MIPS	32-bit hostid	`hwconf`	170a3472
Motorola 88K	32-bit hostid	`lmhostid`	170a3472
NeXT	32-bit hostid	`hostid`	170a3472
RS/6000	32-bit hostid	`uname -m` (returns 000276513100), then remove last 2 digits, and use remaining last 8 digits	02765131
SCO (FLEXlm v.3.0)	Hostid String	`uname -x` (returns SCO00354), then prefix with `"ID_STRING="`	ID_STRING=SCO00354
SGI	32-bit hostid	`echo '/etc/sysinfo -s' 16o p | dc`	69022F72
SUN	32-bit hostid	`hostid`	170a3472
VAX/VMS Alpha/OpenVMS	ethernet address	`ncp`	0800200055327
VAX	ethernet address	`pfstat`	0800200055327

Obtaining license files

When you have determined the hostid of each workstation you want to configure as a license server, you can obtain your license file (or files). This process involves contacting an authorized Apple license dealer, providing your hostids, and receiving the necessary license files. See your MAE product package or the Read Me file (`README.TXT`) for instructions.

IMPORTANT The license file contains encrypted "license keys" that must be unmodified for MAE to run. Do not change a license file unless you are familiar with the file structure and the consequences of any changes you make.

Determining if you need an options file

An options file is not required for MAE, but you may want one if you need to be able to control such licensing options as

- granting or denying the use of licenses
- reserving licenses for specific users, groups, or computers
- limiting the information kept in the log file

If you decide you want to create an options file to tailor the use of MAE licenses for your site, see the section "The Options File," later in this chapter.

Determining where to put license management components

The license manager daemon, `lmgrd`, the `AppleMAE` vendor daemon, and the `license.dat` file are in the `apple/bin` directory by default. If you are running one or more separate license servers, you should move these files and the log file to the appropriate server or servers.

The license manager daemon, `lmgrd`

The license manager daemon, `lmgrd`, must be run on the server node specified in the SERVER line of the license file.

The vendor daemon, `AppleMAE`

The vendor daemon, `AppleMAE`, can be run anywhere on the network, as long as its pathname is specified correctly in the DAEMON line of the `license.dat` file. However, you should put it on the same server node as the `lmgrd` daemon to minimize possible points of failure in the license management system.

The license file, `license.dat`

If you need to move the license file from its default location, you can identify its new location by using the `-c` option on the `lmgrd` command or by setting the environment variable `LM_LICENSE_FILE`.

To set `LM_LICENSE_FILE` to the location of the license file, use the following commands:

In the C shell:

```
% setenv LM_LICENSE_FILE license.dat-pathname
```

In the Korn and Bourne shells:

```
# LM_LICENSE_FILE=license.dat-pathname
# export LM_LICENSE_FILE
```

Replace *license.dat-pathname* with the full pathname to the license file.

Using the `-c` option of the `lmgrd` command overrides the setting of the `LM_LICENSE_FILE` environment variable.

Note: Some applications may not recognize the `LM_LICENSE_FILE` environment variable.

If you are running MAE on multiple nodes, you have three options for making your license file available to all nodes:

- Place the license file in a partition that is available (through NFS) to all nodes in the network that need the license file. This approach is recommended because it allows you to administer a single copy of the license file.
- Read the license file data from `lmgrd` by setting `LM_LICENSE_FILE` to *port@host*, where *host* is the server hostname and *port* is the port number from the SERVER line in the license file. This approach is easy to administer, but it adds to network traffic, especially if the license file is large.
- Copy the license file to all of the nodes where it is needed. This approach is probably easiest to set up, but it requires you to administer multiple copies of the license file.

Since the `AppleMAE` vendor daemon keeps track of license usage, and since the license file contains encrypted data to protect it against modification, you can move and copy the license file as much as necessary.

Note: If you are running redundant servers, you should have one copy of the license file (as well as `lmgrd` and all vendor daemons) on each server node. If you don't, you lose all the advantages of having redundant servers, since the node holding these files becomes a single point of failure.

License management utilities

MAE includes the following utilities to help you administer license management:

- `lmcksum`—performs a checksum to help verify data entry in a license file
- `lmdiag`—helps you diagnose license checkout problems
- `lmdown`—allows for the graceful shutdown of all license daemons
- `lmhostid`—reports the hostid of a system
- `lmremove`—allows you to remove a single user's license for a specified feature
- `lmreread`—causes the license daemon to reread the license file and start any new vendor daemons
- `lmstat`—gives the status of all network licensing activities
- `lmswitchr`—switches to a new report writer log file
- `lmver`—reports the version of a library or binary file

These utilities are also available as a group through the command `lmutil`. Each license management utility is linked to `lmutil`.

By default, these utilities are also in the `apple/bin` directory. There is little reason for an end user who is not an administrator to use these utilities, except for `lmstat` to find out who is using licenses. You might, therefore, want to move them or limit users' access to these utilities so that a user couldn't, for example, use the `lmdown` utility to shut down license management. For information about each command, see the man page for that command. The man pages for these commands are installed with the other MAE man pages in `/opt/apple/man`.

The log file

The log file can be kept on the network, as long as its pathname is provided with the `lmgrd` command or is specified correctly in the REPORTLOG option in the options file. However, it is best to put it on the same server node as the `lmgrd` daemon to minimize possible points of failure in the license management system.

Starting the license manager, `lmgrd`

Once you have completed the setup steps outlined earlier, the license manager daemon, `lmgrd`, can be run.

To start the license manager daemon, execute an `lmgrd` command similar to the following

In the C shell:

`%` *lmgrd-pathname* `-c` *license.dat-pathname* `>&` *log-pathname* `&`

In the Korn or Bourne shell:

`$ nohup` *lmgrd-pathname* `-c` *license.dat-pathname* > *log-pathname* 2>&1 &

Replace *lmgrd-pathname* with the full pathname of the `lmgrd` daemon, *license.dat-pathname* with the full pathname to the license file, and *log-pathname* with the full pathname to the license log file.

To start the `lmgrd` daemon automatically every time you reboot the license server, add to the appropriate startup file a line similar to the following:

`nohup` *lmgrd-pathname* `-c` *license.dat-pathname* `>` *log-pathname* `2>&1 &`

Startup files vary from system to system. See the documentation for your operating system for specific information about modifying startup files.

Note: Adding this line does not start the daemon until you reboot your license server.

The format of the license file

The license file `license.dat` contains all site-specific information required by the license management system. This information includes

- server names and host IDs
- vendor names and paths to vendor daemon executables
- information about applications and features of applications

Note: Some sites that already use FLEXlm license management will want to combine the MAE license file with existing license files. This section includes some information that is not relevant to MAE but will be of use to those sites. Because MAE license management requires FLEXlm version 4.1, you must use 4.1 to be able to merge license files. Consequently, no information about older versions is included in this section.

License files usually begin with a SERVER line (or three lines for redundant servers) followed by one or more DAEMON lines, followed by one or more FEATURE lines.

You can modify four types of parameters in the license file:

- node names on the SERVER line
- port numbers on the SERVER line
- pathnames on the DAEMON line
- options file pathnames on the DAEMON line

Nearly everything else in the license file is used to compute the encryption code and should be kept exactly as supplied by Apple. All parameters that you can modify are explicitly identified in this section. All data in the license file is case sensitive, unless otherwise indicated.

SERVER lines

The SERVER line specifies the node name and hostid of the license server, and the port number of the license manager daemon (`lmgrd`). Normally a license file has one SERVER line. Three SERVER lines means that you are using redundant servers. The absence of a SERVER line means every FEATURE or INCREMENT line in the license file is governed by an uncounted license. You cannot delete SERVER lines from a license file because the hostids from the SERVER lines are encrypted into the passwords on every FEATURE and INCREMENT line.

The format of the SERVER line is

`SERVER` *`nodename`* `id [`*`port-number`*`]`

where	**is the**
`nodename`	string returned by the UNIX `hostname` or on some systems, `uname -n` command. Modifiable.
`id`	the string returned by the `lmhostid` command
`port-number`	TCP port number to use. Optional, but if not specified, then the FLEXlm TCP service must be present in the network services database. The port number is usually specified. Modifiable.

Example:

```
SERVER enterprise 0122345 1701
```

DAEMON lines

The DAEMON line specifies the daemon name and path. The format of the DAEMON line is shown below.

```
DAEMON daemon-name daemon_path [options_file_path]
```

where	is the
daemon-name	name of the vendor daemon, such as AppleMAE, used to serve some feature in the file
daemon_path	pathname to the executable file for this daemon. Generally you can install the daemon anywhere. (It is recommended however, that it be installed on the same license server node as the `lmgrd`.) Modifiable.
options_file_path	full pathname of the optional options file for this daemon. Modifiable.

Example:

```
DAEMON sampled /usr/local/sampled \
/usr/local/options/options.dat
```

Note: The backslash (\) line continuation character is supported.

FEATURE or INCREMENT lines

A FEATURE line describes the license to use a product such as MAE. An INCREMENT line can be used in place of a FEATURE line, as well as to "incrementally" add licenses to a prior FEATURE or INCREMENT line in the license file.

The format of the FEATURE or INCREMENT line is as follows:

```
FEATURE|INCREMENT name daemon version exp_date #lic code \
[HOSTID=hostid][VENDOR_STRING="vendor-string"] \
[vendor_info="..."] [dist_info="..."] [user_info="..."] \
[asset_info="..."] [ISSUER="..."] [NOTICE="..."] [ck=nnn] \
[OVERDRAFT=nnn] [DUP_GROUP=NONE|SITE|[UHDV]]
```

where	**is the**
name	name given to the feature by the vendor
daemon	name of the vendor daemon (for example, `AppleMAE`); also found in the DAEMON line. The specified daemon serves this feature.
version	version of this feature that is supported by this license
exp_date	expiration date (for example, 7-may-1993). If the year is 0, then the license never expires.
#lic	number of concurrent licenses for this feature If the number of users is set to 0, the licenses for this feature are uncounted. In this case, no `lmgrd` is required, but a hostid on the FEATURE line is required.
code	license key for this FEATURE line. Two otherwise identical FEATURE or INCREMENT lines may have different license keys if the start dates are different.

The following fields are all optional:

"*vendor_string*"	Vendor-defined string, enclosed in double quotes. This string can contain any 64 characters except a quotation mark. White space is ignored for the purpose of the encryption algorithm, but may be significant to the software vendor.
hostid	String returned by `lmhostid`. Used only if the feature is to be bound to a particular host, whether its use is counted or not. (Numeric hostids are case insensitive.)
`vendor_info="…"`	Additional information provided by the software vendor. Modifiable
`dist_info="..."`	Additional information provided by the software distributor. Modifiable
`user_info="..."`	Additional information that you can provide. Modifiable
`asset_info="..."`	Additional information that you can provide for asset management. Modifiable.
`ISSUER="..."`	Issuer of the license.
`NOTICE="..."`	A field for intellectual property notices.
`ck=`*nnn*	A checksum, useful with the `lmcksum` utility, which will verify that the license has been entered correctly by the end user. Not encrypted.
`OVERDRAFT=`*nnn*	The overdraft policy allows your vendor to specify a number of additional licenses that users will be allowed to use beyond the number of licenses you have purchased. This allows your users to not be denied service when in a "temporary overdraft" state.

`DUP_GROUP=...` Duplicate Grouping parameter can be specified in the license to allow additional use of a feature that is not counted against the number of licenses. The syntax is

`DUP_GROUP=NONE|SITE|[UHDV]`

where

`U = DUP_USER`

`H = DUP_HOST`

`D = DUP_DISPLAY`

`V = DUP_VENDOR_DEF`

Any combination of `UHDV` is allowed, and the `DUP_MASK` is the OR of the combination. For example "`DUP_GROUP=UHD`" means the duplicate grouping is (`DUP_USER|DUP_HOST|DUP_DISPLAY`), allowing a user on the same host and display to have additional uses of a feature.

Examples:

```
FEATURE MAE AppleMAE 3.000 31-dec-1997 20 1234567890 “xyz”
INCREMENT f1 demo 1.000 1-jan-0 5 12345678901234567890 \
  HOSTID=INTERNET=195.186.*.* NOTICE=”Licensed to XYZ corp”
```

Only the first FEATURE line for a given feature will be processed by the vendor daemon. If you want to have additional copies of the same feature (for example, to have multiple node-locked counted features), then you must use multiple INCREMENT lines. INCREMENT lines form license groups based on the feature name, version, and node-lock hostid. If the feature name, version, and node-lock hostid parameters (and the vendor string, if the vendor specified one) match those of a prior INCREMENT or FEATURE line, the new number of licenses is added to the old number. If any of these parameters do not match, a new group of licenses is created in the vendor daemon, and this group is counted independently from others with the same feature name. A FEATURE line does not give an additional number of licenses, whereas an INCREMENT line ALWAYS gives an additional number of licenses.

Note: The `ls_use_all_feature_lines` option causes FEATURE lines to function as INCREMENT lines. This option is rarely used.

FEATURESET lines

The FEATURESET line is used to prevent FEATURE lines from being added to or removed from the license file. The format of the FEATURESET line is as follows:

```
FEATURESET daemon-name code
```

where	is the
daemon-name	Name of the vendor daemon used to serve some features in the file
code	Encryption code for this FEATURESET line. This code encrypts the codes of all features this daemon supports, so no FEATURE lines can be removed, added to, or rearranged in this license file.

Example:

```
FEATURESET AppleMAE 12345678
```

PACKAGE lines

The PACKAGE line supports two different licensing needs:

- to license a set of products as a suite
- to allow efficient distribution of a license file with many features that largely share the same FEATURE line arguments

Note: Although MAE license management supports PACKAGE lines, MAE itself consists of only one feature.

A PACKAGE line, by itself, does not license anything—it requires a matching FEATURE/INCREMENT line to license the whole package. A PACKAGE line can be included in a product by the software vendor, independent of any licenses. Later, when you purchase a license for that package, one or more corresponding FEATURE/INCREMENT licenses will enable the package.

Example:

```
PACKAGE pkg_name vendor pkg_version pkg_key \
  COMPONENTS=pkg_list \ [ OPTIONS=pkg_options ]
```

where	**is the**
pkg_name	Name of the package. The corresponding FEATURE/INCREMENT line must have the same name.
vendor	Name of the vendor daemon that supports this package
pkg_version	Version of the package. The enabling FEATURE/INCREMENT line must have the same version.
pkg_key	20-character encryption code

pkg_list	List of components, in the following format:
	feature[:*version*[:*count*]]
	PACKAGE lines must include at least one COMPONENT parameter. *Version* and *count* are optional; if left out, their values come from the corresponding FEATURE/INCREMENT line.
	Count is legal only if OPTIONS=SUITE is not set—if not set, the number of licenses will be the count on the COMPONENTS line multiplied by the number of licenses in the FEATURE/INCREMENT line.
	Examples:
	`COMPONENTS=″comp1 comp2 comp3 comp4″`
	`COMPONENTS=″comp1:1.5 comp2 comp3:2.0:4″`

pkg_options

Currently the only supported option is SUITE. This is what distinguishes a suite package from a package used to ease distribution.

With OPTIONS=SUITE, the FEATURE corresponding to the package is checked out in addition to the component feature.

If OPTIONS=SUITE is not set, then the corresponding FEATURE is removed once the package is enabled and it is not checked out when a component feature is checked out.

```
PACKAGE suite demo 1.0 \
  20CHARCODEXXXXXXXXXX COMPONENTS=″comp1 \
  comp2″ OPTIONS=SUITE

FEATURE suite demo 1.0 1-jan-0 5 \
20CHARCODEXXXXXXXXXX
```

In this example, two features, comp1 and comp2 (both version 1.0), have five non-expiring licenses available. When comp1 or comp2 is checked out, “suite” will also be checked out.

```
PACKAGE suite demo 1.0 \
  COMPONENTS=″comp1:1.5:2 comp2:3.0:4 \
  comp3″

FEATURE suite demo 1.0 1-jan-95 3 \
20CHARCODEXXXXXXXXXX\
  ISSUER=distrib1
```

In this example, the component versions override the feature versions, and the number of licenses available for any component is the product of the three licenses for suite and the number of licenses for that component. The result is equivalent to:

```
FEATURE comp1 demo 1.5 1-jan-95 6 \
  20CHARCODEXXXXXXXXXX\
  ISSUER=distrib1

FEATURE comp2 demo 3.0 1-jan-95 12 \
  20CHARCODEXXXXXXXXXX\
  ISSUER=distrib1

FEATURE comp3 demo 1.0 1-jan-95 3 \
  20CHARCODEXXXXXXXXXX\
  ISSUER=distrib1
```

UPGRADE lines

The format of the UPGRADE line is as follows:

```
UPGRADE name daemon fromversion version exp_date #lic code \
  "string" \ [hostid] ck=nnn
```

This format is the same as for a FEATURE or INCREMENT line, with the addition of the *fromversion* parameter. An UPGRADE line removes up to the number of licenses specified by *#lic* from any old version of the feature that is equal to or greater than the *fromversion* value and creates a new version with that same number of licenses.

For example, the two lines

```
FEATURE f1 demo 1.000 1-jan-94 5 9BFAC03164EDB7BC0462 ""
UPGRADE f1 demo 1.000 2.000 1-jan-94 2 \
  1B9A30316207EC8CC0F7 ""
```

would reassign two of the five existing licenses for version 1.0 of feature f1 to version 2.0 of feature f1.

UPGRADE will operate on the closest preceding FEATURE or INCREMENT line with a version number that is >= *fromversion*, and < *version*.

Note that UPGRADE does not work for node-locked, uncounted licenses because the license count is irrelevant.

A sample license file

The following example is a license file for a single vendor with two features.

```
SERVER  excellent_server  17007ea8  1700
DAEMON  AppleMAE  /apple/bin
FEATURE  MAE1  AppleMAE  1.000  01-jan-1993 10 \
  1EF890030EABF324 ""
FEATURE  MAE2  AppleMAE  1.000  01-jan-1993  10 \
  0784561FE98BA073 ""
```

The license file above would allow the license server excellent_server with the hostid 17007ea8 to serve 10 floating licenses for both feature MAE1 and feature MAE2 to any user on the network.

Limits on modifiable parameters

This section lists the limits on modifiable license file parameters:

host names	64 characters
user names	20 characters
license file lines	2048 characters, with "\" as a continuation character

Types of licenses

License files are created by the software vendor. License files can specify any of the following policies:

- floating (concurrent) usage
- node-locked
- floating usage on a limited set of hosts
- mixed node-locked and floating usage

Floating (concurrent) licenses

A floating license allows anyone on the network to use the licensed software, up to the limit specified in the license file. (This approach is also referred to as concurrent usage or network licensing.) Floating licenses have no hostids on the individual FEATURE lines. This configuration requires an `lmgrd` daemon because the daemon tracks the concurrent usage of the licenses.

Note: MAE licenses are issued as floating licenses. If you are using a demonstration copy of MAE whose license has expired, see the Read Me file (README.TXT) for information on obtaining a valid license.

An example of a license file that provides floating licenses is

```
SERVER lulu 17001234 1700
DAEMON xyzd /etc/xyzd
FEATURE f1 xyzd 1.00 1-jan-95 2 code1 ""
FEATURE f2 xyzd 1.00 1-jan-95 6 code2 ""
FEATURE f3 xyzd 1.00 1-jan-95 1 code3 ""
```

This license file specifies that two licenses for feature f1, six licenses for feature f2, and one license for feature f3 are available for any node on the network that can access the license server lulu.

Node-locked licenses

Node-locking restricts the use of the licensed software to one specific node (workstation). A node-locked license has a hostid on any FEATURE line that is node-locked to a particular host. There are two types of node-locked licenses: uncounted and counted.

If the number of licenses is set to 0, then the licenses are uncounted and unlimited use is permitted on the specified node. This configuration does not require an `lmgrd` daemon because there is no need to track the concurrent usage of the features.

The following license file allows unlimited usage of feature “f1” on the nodes with host IDs of 12001234 and 1700ab12:

```
FEATURE f1 xyzd 1.000 1-jan-95 0 code1 "" 12001234
FEATURE f1 xyzd 1.000 1-jan-95 0 code2 "" 1700ab12
```

If these were the only FEATURE lines in this license file, no `lmgrd` daemon would be necessary (and you should not start one).

The following license file allows three licenses for feature f1, but only on the node with host ID 1300ab43. In this case, the daemons should be run on the same node as the feature; there is no reason to run the daemons on another node.

```
SERVER lulu 1300ab43 1700
DAEMON xyzd /etc/xyzy
FEATURE f1 zyzd 1.00 1-jan-95 3 code "" 1300ab43
```

Floating licenses with limited availability

Another solution that some vendors choose is a combination of node-locked licenses and floating licenses. This approach provides a fixed number of floating licenses for use only on a set of servers that you define.

In this configuration, the license file has one node-locked FEATURE line for each node that is to be able to run the software. The same number of licenses is generally specified on each FEATURE line. This configuration does require an `lmgrd` daemon to track the concurrent license usage.

For example, the following license file allows a total of two simultaneous licenses of a feature called f1 on any of the three nodes specified, with the daemons running on the license server lulu:

```
SERVER lulu 17001234 1700
DAEMON xyzd /etc/xyzd
FEATURE f1 xyzd 1.00 1-jan-93 2 code1 "" 17001234
FEATURE f1 xyzd 1.00 1-jan-93 2 code2 "" 17002222
FEATURE f1 xyzd 1.00 1-jan-93 2 code3 "" 17003333
```

Mixed node-locked and floating licenses

Uncounted node-locked and floating licenses can be mixed in the same license file.

The following license file allows unlimited use of feature f1 to be used on nodes 17001111 and 17002222, while allowing two licenses for the same feature f1 to be used anywhere else on the network:

```
SERVER lulu 17001234 1700
DAEMON xyzd /etc/xyzd
FEATURE f1 xyzd 1.00 1-jan-93 0 code1 "" 17001111
FEATURE f1 xyzd 1.00 1-jan-93 0 code2 "" 17002222
FEATURE f1 xyzd 1.00 1-jan-93 2 code3 ""
```

This configuration requires an `lmgrd` daemon because of the two floating licenses on the third FEATURE line.

The license finder

The license finder provides license file path information to licensed applications so that you do not have to configure the `LM_LICENSE_FILE` environment variables for all your users. The license finder provides a single license file path to all clients.

MAE and other licensed applications determine the name of the license file using the following algorithm:

1. **Use the pathname specified by `LM_LICENSE_FILE`, if it is set.**
2. **If step 1 fails, attempt to connect to port "FLEXLM" from the services map on flexlm_license_finder, flexlm_license_finder2, flexlm_license_finder3, in that order. If port "FLEXLM" is not available, the application attempts to connect to port 744 (the "official" FLEXlm port). The connection timeout interval is fixed at 2 seconds.**

   The connection attempt is made using TCP/IP. No broadcasts are involved. (For debugging, the connect timeout can be modified by setting FLEXLM_FINDER_TIMEOUT to an integer). If the lookup of flexlm_license_finder fails, this step terminates.

   If the connection to any license finder is successful, the application gets the license file path from the finder.
3. **If steps 1 and 2 fail, use the default license file path.**

Starting the license finder

To start the license finder, execute the following command on the node flexlm_license_finder:

```
lmgrd -f finder_file_pathname
```

Using the `-f` option with this command causes `lmgrd` to operate exclusively as a license finder.

IMPORTANT In license finder mode, `lmgrd` will only process license finder request messages and send the license file path back to the client application. It will not start any vendor daemons or process any other client request. This mode is provided in `lmgrd` so that you do not need a separate executable.

Finder file format

The "finder file" format is as follows:

```
DATA lfpath license_file_path
```

For example:

```
DATA lfpath \
  /usr/local/flexlm/licenses/license.dat:1234@host2
```

Note: The license finder is not available on VMS systems.

Preventing conflicts with other licensed applications

If you are running other licensed products using FLEXlm license management, you may have to change your license server configuration to prevent licensing conflicts. You can use one of three configurations to accomplish this:

- Multiple license server nodes, each running one `lmgrd` daemon and one license file
- One license server node running multiple `lmgrd` daemons and multiple license files
- One license server node running one `lmgrd` daemon and one license file

Note that each `lmgrd` can control only a single license file. While running separate `lmgrd` daemons and license files simplifies initial installation in some ways, it would probably increase the day-to-day administration of your licenses. With configuration 1, you would have more license servers to monitor; with configuration 2, you would have a single server node, but multiple license manager daemons to administer.

A license server node is specified by its hostname and hostid in a SERVER line in a license file. License files are considered to be compatible if they contain the same number of SERVER lines and the hostid value of each SERVER line in one file exactly matches the hostid value of the corresponding SERVER line in the other file. Compatible license files can be combined to create configuration 3, in which one `lmgrd` daemon and one license file on the same server can manage the licenses for multiple applications.

Combining license files

If your license files are compatible (that is, they use the same server nodes), you can combine them with any text editor by writing all of their contents into one file, then deleting duplicate SERVER lines so that only one set of SERVER lines remains. Save the resultant data to a file named `license.dat`. If you don't put the file in the default location, then you also need to set the `LM_LICENSE_FILE` variable to the pathname you choose for the file.

Note that you are not required to combine compatible license files; you always have the option of running separate copies of `lmgrd` as described in the next section.

Note: For MAE you must run at least version 4.1 of `lmgrd`. This is true regardless of the version you are using with your other applications. The 4.1 version can process the data from older license files that you might want to combine with the MAE license file. You cannot, however, use an older version of `lmgrd` with the newer license files and vendor daemons.

Using separate license files on the same server node

If the license files for two products are not compatible, then you must keep the license files separate and run separate copies of `lmgrd` for each license file. You can run multiple copies of `lmgrd` on a single server node.

Reasons why license files may not be compatible include the following:

- They are set up to run on different server nodes, so hostids are different.
- One file is set up for single server (has only one SERVER line), the other is set up for redundant servers (has multiple SERVER lines).
- One vendor uses a custom hostid algorithm, so the hostids on the SERVER lines are different even though the files are for the same machine.

When you run multiple copies of `lmgrd` separately for each license file, there are two details to remember:

- The port number on the SERVER line of each license file must be unique. You can use a standard text editor to change the port number in each license file so that they are all different.
- You must make sure that you are using a compatible version of `lmgrd` when you start it up for a particular license file. This can be done by either using an explicit path to `lmgrd` , or by renaming `lmgrd` to something specific for that product (some vendors do this to make it easier to identify the `lmgrd` they ship with their product).

When running client programs (such as a licensed application), you can set the `LM_LICENSE_FILE` environment variable to point to multiple license files. For example, you may have a license file from vendor ABC and a license file from vendor XYZ with incompatible servers. You can place the license file from vendor ABC into the directory

`/usr/flexlm/abc.dat`

and the license file from vendor XYZ into the directory

`/usr/flexlm/xyz.dat`

then set the `LM_LICENSE_FILE` environment variable to point to both of them. Multiple pathnames in `LM_LICENSE_FILE` must be separated by a colon ":".
For example, in the C shell:

```
% setenv LM_LICENSE_FILE \
  /usr/flexlm/abc.dat:/usr/flexlm/xyz.dat
```

In the Korn and Bourne shells:

```
# LM_LICENSE_FILE=/usr/flexlm/abc.dat:/usr/flexlm/xyz.dat
# export LM_LICENSE_FILE
```

Using a different server node

If applications use different license server nodes, each set of license servers requires separate license files. The resulting (multiple) license files can be installed in convenient locations, and you can set the `LM_LICENSE_FILE` environment variable as described in the previous section.

Each application will query each license file in the order in which it is listed in the `LM_LICENSE_FILE` path. If the license server serving a license file listed in the environment variable is unreachable, perhaps due to an NFS problem, changing the `LM_LICENSE_FILE` would allow a user to obtain a license from another server.

The options file

An options file allows you to control such licensing options as

- granting or denying the use of licenses
- reserving licenses for specific users, groups, or computers
- limiting the information kept in the log file

An options file is not required for MAE, but you may decide you want to create one to tailor the use of MAE licenses for your site.

When `lmgrd` starts a vendor daemon, it passes it the location of the options file. There can only be one options file per vendor daemon and each vendor needs a separate options file.

You can use a text editor to create an options file following the guidelines in this section. To use the options file, you must then add its pathname to the appropriate license file in the fourth field on the DAEMON line for the application's vendor daemon. For example,

```
DAEMON AppleMAE /etc/xyzd /usr/local/MAE_opts
```

would enable the `AppleMAE` daemon to use the options file `MAE_opts`.

Each line of the file controls one option. The options are as follows:

EXCLUDE	Deny a user access to a feature.
EXCLUDEALL	Deny a user access to all features served by this vendor daemon.
GROUP	Define a group of users for use with any options. USER_GROUP is an alias for GROUP.
HOST_GROUP	Define a group of hosts for use with any options.
INCLUDE	Allow a user to use a feature.
INCLUDEALL	Allow a user to use all features served by this vendor daemon.
LINGER	Cause the vendor daemon to hold licenses for a period after the application checks them in or exits.
NOLOG	Turn off message logging for certain items.
REPORTLOG	Specify that a log file be written.
RESERVE	Reserve licenses for a user.
TIMEOUT	Specify idle timeout for a feature, releasing it for use by another user.

You can use the backslash (\) character as a continuation character in options file lines. You can include comments by starting each comment line with a pound sign (#). Everything in an options file is case sensitive.

The following sections detail the options that you can specify in the options file.

The EXCLUDE option

`EXCLUDE` *`featurename type name`*

Excludes a user, host, display, or group from the list of those allowed to use the feature. Excluded users will not be allowed to use the feature.

- *featurename* – name of the feature or application (MAE, for example)
- *type* – one of USER, HOST, DISPLAY, or GROUP, as described later in this section
- *name* – name of the user or group to exclude

To exclude the user hank from the list of users able to use feature f1:

EXCLUDE f1 USER hank

The EXCLUDEALL option

`EXCLUDEALL` *`type name`*

Excludes a user, host, display, or group from using all features served by this vendor daemon.

- *type* – one of USER, HOST, DISPLAY, GROUP or HOST_GROUP
- *name* – name of the user or group to exclude

To exclude any user on the server chaos from using all features served by this vendor daemon:

```
EXCLUDEALL HOST chaos
```

The GROUP option

`GROUP` *`groupname usernamelist`*

`USER_GROUP` *`groupname usernamelist`*

Defines a group of users for use in INCLUDE, INCLUDEALL, EXCLUDE, EXCLUDEALL, and RESERVE option lines.

- *groupname* – name of the group being defined
- *usernamelist* – list of user names in that group

To define the group Hackers consisting of bob, howard, and james:

```
GROUP Hackers bob howard james
```

USER_GROUP is an alias for GROUP.

The HOST_GROUP option

`HOST_GROUP` *`groupname hostnamelist`*

Defines a group of hosts for use in INCLUDE, INCLUDEALL, EXCLUDE, EXCLUDEALL, and RESERVE option lines.

- *groupname* – name of the group being defined
- *hostnamelist* – list of host names in that group

To define the group Hackers consisting of bob, howard, and james:

```
GROUP Hackers bob howard james
```

The INCLUDE option

`INCLUDE` *`featurename type name`*

Includes a user, host, display, or group in the list of those allowed to use the feature. Anyone not in an INCLUDE statement will not be allowed to use that feature.

- *featurename* – name of the feature being affected
- *type* – one of USER, HOST, DISPLAY, GROUP, or HOST_GROUP
- *name* – name of the user or group to include

To include user bob in the list of users able to use feature f1:

```
INCLUDE f1 USER bob
```

The INCLUDEALL option

`INCLUDEALL` *`type name`*

Includes a user, host, display, or group in the list of those allowed to use all features served by this vendor daemon.

- *type* – one of USER, HOST, DISPLAY, GROUP, or HOST_GROUP
- *name* – name of the user or group to include

To allow the user jane to use all features served by this vendor daemon:

```
INCLUDEALL USER jane
```

The LINGER option

LINGER *featurename interval*

This causes the daemon to "hold on" to the license for *featurename* for *interval* seconds after the application checks in the license or exits. This option can be useful for short-duration programs that will be used many times in a row by the same user, to ensure that the user will be able to re-acquire the license repeatedly. This is only useful if the application uses duplicate grouping. Otherwise, LINGER will cause you to use extra licenses.

The NOLOG option

`NOLOG` *`what`*

Turns off logging of specific events by license management daemons. You can use this option to reduce the size of the lmgrd or "debug" log file.

- *what* – what to turn off; one of IN, OUT, DENIED, or QUEUED

Each type of event requires its own NOLOG option line. For example, to turn off logging of checkins:

```
NOLOG IN
```

To turn off logging of checkouts and queued requests, two separate NOLOG lines are required:

```
NOLOG DENIED
NOLOG QUEUED
```

Note: The log file is an ASCII text file.

The REPORTLOG option

`REPORTLOG` *`filename`*

REPORTLOG specifies the file that will contain the report-writer log for a vendor daemon. If *filename* begins with a + character, the file will be appended to; otherwise the file will be overwritten each time the daemon is started.

The RESERVE option

RESERVE *numlic featurename type name*

Reserves licenses for a specific user. Any licenses reserved for a user are dedicated to that user. Even when that user is not actively using the license, it will be unavailable to other users. However, a reserved license not actually in use will not cause an overdraft to be reported by lmgrd .

- numlic—number of licenses to reserve
- featurename—name of feature to reserve the license for
- type—one of USER, HOST, DISPLAY, GROUP, or HOST_GROUP
- name—name of the user or group to reserve licenses for

To reserve one license of feature f1 for user mel:

```
RESERVE 1 f1 USER mel
```

The TIMEOUT option

TIMEOUT *featurename seconds*

Set the time after which an inactive license is reclaimed by the vendor daemon.

- featurename—name of the feature (or application)
- seconds—number of seconds after which inactive license is reclaimed

To set the timeout for feature f1 to one hour (3600 seconds):

```
TIMEOUT f1 3600
```

The license manager releases the license for a feature if another user needs the license and that feature (or application) has been idle for a period longer than the specified timeout period. The feature must explicitly declare itself idle for this to work, or the application must be stopped (for example, with Control-Z). That is, unless the application explicitly supports this feature, it will not work while the feature is running. The daemon declares a feature idle when it has not heard from the feature (the client sends heartbeats).

The application can set a minimum value for the timeout. In this case, if you specify a timeout value smaller than the minimum, the minimum is used. The default minimum value is 9000 seconds (15 minutes). Alternatively, the application can disable the timeout feature, in which case the TIMEOUT option has no effect.

If you do not specify a timeout value in your options file, then there will be no timeout for that feature. Licenses are only freed by TIMEOUT when a new request for a license is made.

Rules of precedence in options files

Before you can use options to administer licenses effectively, you must understand the options file precedence. INCLUDE and EXCLUDE statements can be combined in the same options file and control access to the same features. When doing so, keep in mind the following:

- If there is only an EXCLUDE list, everyone who is not on the list will be allowed to use the feature.
- If there is only an INCLUDE list, only those users on the list will be allowed to use the feature.
- If neither list exists, then everyone is allowed to use the feature.
- The EXCLUDE list is checked before the INCLUDE list; so someone who is on both lists will not be allowed to use the feature.

Once you create an INCLUDE or EXCLUDE list, everyone else is implicitly not part of the group. This feature allows you, as an administrator, to avoid having to explicitly list each user that you wish to allow or deny access to.

Simple options file example

The following is an example of a simple options file.

```
RESERVE 1 compile USER robert
RESERVE 3 compile HOST mainline
EXCLUDE compile USER lori
NOLOG QUEUED
```

This options file has the following effects:

- Reserve one license for the feature compile for the user robert.
- Reserve three licenses for the feature compile for anyone on a computer with the hostname mainline.
- Prevent the user lori from using the compile feature on any node on the network.
- Cause QUEUED messages to be omitted from the log file.

The sum total of the licenses reserved must be less than or equal to the number of licenses specified in the FEATURE line in the license file. In this example, there must be a minimum of four licenses on the compile FEATURE line. If fewer licenses are available, only the first set of reservations (up to the license limit) are used.

If this data were in file `/usr/local/flexlm/options/local.options`, then you would modify the license file DAEMON line as follows:

```
DAEMON xyzd /usr/local/xyzd \
  /usr/local/flexlm/options/local.options
```

Limiting access for multiple users

Each INCLUDE, INCLUDEALL, EXCLUDE, EXCLUDEALL, and RESERVE line must have a single user name or a single group listed. For example, to exclude bob, howard, and james from using the feature called toothbrush you could create the following options file:

```
EXCLUDE toothbrush USER bob
EXCLUDE toothbrush USER howard
EXCLUDE toothbrush USER james
```

Alternatively, you could create a group of users and use a single Group option line to exclude them all from using the feature. Like the previous example, the following options file would exclude bob, howard and james from using the feature called toothbrush:

```
# First define the group Hackers
  GROUP Hackers bob howard james
# Then exclude the group
  EXCLUDE toothbrush GROUP Hackers
```

You can use HOST_GROUP to allow, deny or reserve licenses for multiple hosts. For example, to exclude all users logged in on the hosts fred and barney from using a feature called f1, add these lines to your options file:

```
HOST_GROUP writers fred barney
EXCLUDE f1 HOST_GROUP writers
```

EXCLUDE example

The following options file provides an example of ways to use the EXCLUDE option.

```
#First Define the group “painters”
GROUP painters picasso mondrian klee
EXCLUDE spell GROUP painters
EXCLUDE spell USER bob
EXCLUDE spell HOST bigbrush
```

This options file would effect the following restrictions:

- Prevent the users picasso, mondrian, and klee from using the feature spell on any machine on the network.
- Prevent the user bob from using the feature spell on any machine on the network.
- Prevent any user logged into the host bigbrush from using the feature spell.
- Allow all users, as long as they are not on bigbrush, and they are not in painters and they are not bob to use feature spell (by implication).

In this case, the two EXCLUDE statements concatenate to create a list of the four users bob, picasso, mondrian, and klee. Note that bob could be added to the group painters if painters is not used for some other license.

INCLUDE example

The following options file provides an example of ways to use the INCLUDE option.

```
INCLUDE paint USER picasso
INCLUDE paint USER mondrian
INCLUDE paint HOST bigbrush
```

This options file would

- Allow the user picasso to use the feature paint on any machine on the network.
- Allow the user mondrian to use the feature paint on any machine on the network.
- Allow any user, as long as they are on the host bigbrush, to use feature paint.
- Deny access to the feature paint to anyone except picasso, mondrian or anyone from the host bigbrush (by implication).

The daemon log file

The license daemons all generate log files with messages in the following format:

mm/dd hh:*mm* (*daemon_name*) *message-text*

#lic	Number of concurrent licenses for this feature
where	**is the**
mm/dd hh:m	Time that the message was logged
daemon_name	Either `lmgrd` or the vendor DAEMON name. In the case where a single copy of the daemon cannot handle all of the requested licenses, an optional "_" followed by a number indicates that this message comes from a forked daemon.
message-text	The text of the message

The log files can be used to

- Inform you when you need to purchase additional software licenses
- Diagnose configuration problems
- Diagnose daemon software errors

Informational messages

`Connected to` *node-name*

This daemon is connected to its peer on node *node-name.*

`CONNECTED, master is` *name*

The license daemons log this message when a quorum is up and everyone has selected a master.

`DEMO mode supports only one SERVER host!`

An attempt was made to configure a demo version of the software for more than one server host.

`DENIED:` *n feature* `to` *user* `(mm/dd/yy hh:mm)`

user was denied access to *n* licenses for *feature*.

`EXITING DUE TO SIGNAL nnn`

`EXITING with code nnn`

All daemons list the reason that the daemon has exited.

`EXPIRED:` *feature*

feature has passed its expiration date.

`IN: feature by user (n licenses) (used: d:hh:mm:ss)`

user has checked back in *n* licenses of *feature* at *mm/dd/yy hh:mm*.

`IN server died: feature by user (n licenses) (used: d:hh:mm:ss)`

user has checked in *n* licenses by virtue of the fact that his or her server stopped running.

`License Manager server started`

The license daemon was started.

`Lost connection to host`

A daemon can no longer communicate with its peer on node *host*, which can cause the clients to have to reconnect, or cause the number of daemons to go below the minimum number, in which case clients may start exiting. If the license daemons lose the connection to the master, they will kill all the vendor daemons; vendor daemons will shut themselves down.

`Lost quorum`

Two of three redundant servers are down, so the daemon will process only connection requests from other daemons.

`MASTER SERVER died due to signal nnn`

The license daemon received fatal signal *nnn*.

`MULTIPLE xxx servers running.`

`Please kill, and restart license daemon`

The license daemon has detected that multiple licenses for vendor daemon *xxx* are running. You should kill all *xxx* daemon processes and restart the license daemon.

`OUT: feature by user (N licenses) (mm/dd/yy hh:mm)`

user has checked out N licenses of *feature* at *mm/dd/yy hh:mm*

`Removing clients of children`

The top-level daemon logs this message when one of the child daemons dies.

`RESERVE feature for HOST name`

`RESERVE feature for USER name`

A license for *feature* is reserved for either user *name* or host *name*.

`RESTarted xxx (internet port nnn)`

Vendor daemon *xxx* was restarted at internet port *nnn*.

`Retrying socket bind (address in use)`

The license servers try to bind their sockets for approximately six minutes if they detect “address in use” errors.

Host ID problems

Symptom: *When you run the license manager on your machine, it tells you it is the wrong hostid.*

Cause: The vendor daemon checks the hostid listed on the SERVER line in the license file; if it does not match the hostid of the machine it is running on, this message will be printed. Possible causes are (1) you are trying to run the license server on a different machine from the machine the file was made for; (2) the hostid of the machine you are running on changed (for example, the HP ID module was moved, or the CPU board was replaced); (3) the hostid in the license file was modified.

Solution: Verify that the hostid of the machine on which the vendor daemon (or node-locked client program) is being run matches the hostid specified in the license file (on the SERVER line for the vendor, or on the FEATURE line for a node-locked client). You can run the `lmhostid` program to see what the hostid is according to the license manager. You may not modify the hostid in the license file. If the hostid of your server machine changes, you will have to get a new license file from your software vendor.

Connection problems

Symptom: *The application program (or `lmstat`) can't connect to the server to check out a license.*

Cause: The licensing routines in the application are unable to make a TCP connection to the server and port specified in the license file. Possible reasons for this are: (1) the wrong license file is being referenced by the application program; (2) the server machine specified in the license file is down; (3) the vendor daemon specified in the license file is not running; (4) the hostname in the license file is not recognized by the system; (5) the network between the client machine and the server machine is down; (6) You are mixing FLEXlm v1.5 (or earlier) and v2.1 (or later) vendor daemons in one license file, and have run `lmgrd` without the `-b` command line option; (7) TCP is not running on your machine.

Solution: Verify that the application is using the proper license file. Verify that the specified server machine is up and reachable by executing another command that uses TCP, such as `rsh` or `rlogin`, from the client to the server. Verify that the vendor daemon is running (you can use `ps` on the server to look for it). Examine the license log file to see if any problems are reported, particularly messages indicating that the vendor daemon has quit. Run `lmstat -a` from the server machine to verify that the vendor daemon is alive. Run `lmstat -a` from the client machine to verify the connection from client to vendor daemon across the network. Try using `telnet <hostname> <portnum>` where `hostname` and `portnum` are the same as on the SERVER line in your license file. If you have the FLEXlm v4.0 utilities, try `lmdiag`.

Symptom: *The application program is able to connect from some machines but not from others.*

Cause: You are running a v1.5 license server on a machine with two Ethernet boards.

Solution: There is a bug in FLEXlm v1.5 in which the license server only listens to the Ethernet address associated with the primary hostname on the system. The simplest solution is to get an upgraded license server from your software vendor (you can run a new license server and still keep running the old application software). Alternatively, you can ask your software vendor for a different license file and run your license daemon on a different machine with a single Ethernet board. If this is not possible, you may be able to modify your host tables to direct all FLEXlm requests to the active port on the machine.

Symptom: *The application program (or* `lmstat`*) gets the error* `can't read data` *when attempting to connect to the license server.*

Cause: The program is able to find the server, but it is not getting the expected data. The most likely cause for this is that you are running a newer application program than the vendor daemon.

Solution: Check the version of the application program. Verify that it is using the expected license file. Check the version of the vendor daemon program referred to by the license file (look at the log file for that daemon).

Other client problems

Symptom: *When you run your application program (or vendor daemon), you get the error* `bad code` or `inconsistent encryption code`.

Cause: Possible causes for this are (1) the license file was modified (either the hostid on a SERVER line or anything on the FEATURE line was changed); (2) the vendor used the wrong version of his license creation program to generate your license file (or there is a bug in that process).

Solution: You may not modify the license file (except for specific fields; see "The License File," earlier in this Appendix). If you need to change something in your license file, you must get a new license file from your software vendor.

Symptom: *When the second user tries to check out a license, the vendor daemon prints an error concerning* `Parameter mismatch` *in the log file and refuses the license.*

Cause: The most likely cause of this problem is that you are simultaneously trying to run two different versions of the application program, and the software vendor has not specifically set up the new version for this kind of compatibility. Check the license server log file for a `comm version mismatch` warning message; this indicates that someone is running an older client than the license server daemon, `lmgrd`.

Solution: Run only the new version of the application (or only the old version).

Other server problems

Symptom: *When you start up* `lmgrd`, *it says* `execl failed` *on your vendor daemon.*

Cause: `lmgrd` uses `execl` to start each vendor daemon running. If there is a problem starting the vendor daemon, this message is output to the log file. This error is typically caused by one of the following: (1) there is no executable at the location referred to by the license file (and printed out in the log file); (2) the executable does not have the proper permissions to be run (the file does not have the "x" bit set, or one of the directories in the path is not readable); (3) there was an error building the executable and it cannot be run; (4) the executable is for a different machine architecture.

Solution: Verify that the path to the vendor daemon is absolute (starts with a slash character, "/") and that it points to the executable program itself, not the containing directory (for FLEXlm v1.5). Ensure that the file exists by doing an `ls -l` of the vendor daemon filename specified in the log file. Make sure you do this as the same user that started `lmgrd`. Verify that the file is executable. Note that if you are running as root and using an NFS-mounted file system, the relevant protection bits are the "other" bits (*not* the "user" bits), even if the file is owned by root. Do a `whatis` on the file (if your system has that program). `whatis` should tell you the file is an executable for the machine you are trying to run it on. Run the vendor daemon directly from the command line. If the vendor daemon is properly linked, it will tell you that it must be run from `lmgrd`; if it crashes or fails to execute, then it is not properly linked.

Symptom: *The license server keeps reporting "lost lock" errors in the log file and exiting.*

Cause: The lockfile (normally placed in `/usr/tmp`) is being removed by someone else. There could be another daemon running, or the license administrator (or a script he or she set up) could have deleted the file.

Solution: Check to see if there is more than one copy of the daemon running: use a command like `ps -aux | grep vendorname` to search for it. Check for more than one `lmgrd` running as well, since it will restart your vendor daemon when it is killed. If more than one `lmgrd` is running, kill them all (using the `kill` command, not `kill -9`). Then kill any remaining vendor daemons (try a simple kill; if that fails then try `kill -9`) and start one fresh copy of `lmgrd`. Check to see if there is a shell script running that cleans out `/tmp` (or `/usr/tmp`). If so, try modifying it so that it does not delete zero length files.

Index

A

acontrol command 97
.alias 71
aliases,
  creating aliases that appear at startup 38–40
  troubleshooting 106
AliasList file 39, 52
apple/bin directory 51
apple/bin/engine file 51
apple/bin/lmgrd file 51
apple/bin/macd file 51
apple/bin/mae file 51
apple directories,
  sharing across platforms 106
apple directory,
  contents 50
  shared by several users 69
  specifying location 25
AppleDouble,
  command option 25
AppleDouble format 46
Apple Extras directory,
  contents 50
apple/lib directory 52
apple/lib/AliasList file 39
apple/lib/btree file 52
apple/lib/ccm file 52
apple/lib/data file 52
apple/lib/engine 60
apple/lib/engine file 52
apple/lib/KeymapDepotDB file 52
apple/license.dat file 53
APPLEPATH,
  specifying 25
APPLEPATH environment variable 45–46
appleping command 90
appleping program 52
AppleMAE daemon 52, 113
apple/man directory 53
Apple (🍎) Menu Items 70
AppleScript directory 50
AppleShare extension 88
apple/shared directory 53
AppleSingle format 46
apple/sys,
  if corrupted 34
apple/sys directory 53
AppleTalk,
  determining status with appletalk -s 89
  diagnostic tools 91
  enabling and disabling 80
  ensuring that a node is alive 90
  for MAE 74
  hardware requirements 75
  implementation in MAE 74–75
  initializing the kernel stack 80
  installation 76
  installation on HP-UX 76–77, 78
  installation on Solaris 79
  listing network visible entities (NVEs) in a zone 90
  Name Binding Protocol (NBP) 81
  network resources 81
  Network Visible Entity (NVE) 81
  reference information 94
  reinstalling if streams is corrupted 91
  removing 80
  restoring network extensions 91
  shutting down the stack with appletalk-D 89
  software requirements 76
  system resources 88
  troubleshooting 91
  unsupported extensions 75
AppleTalk address,
  during multiple concurrent MAE sessions 87
AppleTalk ARP packets 89
appletalk command 89
AppleTalk DDP sockets,
  during multiple concurrent MAE sessions 87
AppleTalk kernel modules,
  installing 30
appletalk program 52

AppleTalk protocols,
AppleTalk Address Resolution Protocol (AARP) 74
AppleTalk Data Stream Protocol ADSP 74
AppleTalk Echo Protocol (AEP) 74
AppleTalk Filing Protocol (AFP) 74
AppleTalk Session Protocol (ASP) 74
AppleTalk Transaction Protocol (ATP) 74
Datagram Delivery Protocol (DDP) 74
EtherTalk Link Access Protocol (ELAP) 74
Name Binding Protocol (NBP) 74
Printer Access Protocol (PAP) 74
Routing Table Maintenance Protocol (RTMP) 74
Zone Information Protocol (ZIP) 74
applications,
and processor performance 38
optimal locations 46
asdtool command 47
asdtool program 52
asecure command 97
atlookup command 90
atlookup program 52
AUDIO environment variable 96, 97
Audio Server 96
automounter troubleshooting 105

B

balloon help 16

C

caches,
cache conflict 65
cache hit 65
cache line 65
cache miss 65
cache tag 65
CPU caches 64
data cache 65
directly mapped cache 65
emulator dispatch table's use of 67
external cache 65, 66
MAE's use of 67
multilevel cache 66
multiple set associative cache (when optimizing cache use) 68
primary cache 66
secondary cache 66
secondary cache (when optimizing cache use) 68
separate cache 66
set associative cache 66
set size 66
unified cache 66
unified cache (when optimizing cache use) 68
UNIX I/O caches 64
virtually mapped cache 66
cache use,
optimization by MAE 67–68
CD-ROM,
mounting on a server 20
CellNamesMAE file 107
color flash 57, 62
Color Flash Reduction option 58
color table 57
command options 24–25
commands,
acontrol 97
appleping 90
appletalk 89
asecure 97
atlookup 90
config 77, 78
mae 55
modinfo 79
strvf 79
swapinfo 63
top (HP) 63
vmstat 64
xcmap 62
xdpyinfo 62
config command 77, 78
control panels 70
File Sharing Monitor 87
MAE General 32, 36
Memory 36
Network 81
Sharing Setup 83
Sound 96
Users & Groups 82, 83, 85
copy on write 61
CPU caches 64
customizing easy install 30

D

DDP sockets. *See* AppleTalk DDP sockets
Default_MIV_file 52
demos directory 50
Desktop database 44
disk space,
application installers 72
Finder 72
MAE's use of 69–72
major consumers 71
system resources 72
display,
setting 25

display, troubleshooting 100
DISPLAY variable 25
.DMlock 71
.DMlock file 54
docs directory 50
documentation. *See* MAE documentation

E

easy install,
customizing 30
emulator dispatch table,
use of caches 67
environment variables,
in X resources 28
/etc/mae_fileshare file 83
/etc/system file 48
Ethernet 75
extensions 70, 88
distributing 35

F

features,
new in MAE version 3.0 12
Fetch 106
file formats,
AppleSingle and AppleDouble 46
.FileIDs file 54
file sharing,
potential security risks 85
File Sharing control panel 88
File Sharing extension 88
File Sharing Monitor control panel. *See* control panels
file systems, troubleshooting 104
Finder Window Update (in MAE General control panel) 36
floating licenses 133
font directory 53
fonts,
default fonts 53
sharing global fonts 35
.fs_cache 54, 71
.fs_info file 54

G

graphics buffer,
allocation of a second buffer 62
determining size 61
graphics in MAE 56

H

help,
toolbar 17
HP-UX,
booting from the saved kernel 93
HP-UX.save file,
booting HPs from 93
HyperCard player 51

I, J

iconifying MAE 38
iconify option 25
icons, troubleshooting 100
installation,
AppleTalk 76
list of package installed 69
packages 31
setting up MAE on a server 19–21
installation files,
copying to a server 21
installation of MAE,
using command-line options 30
installing application software,
with a TIV 32
installing MAE,
options 30

K

keyboard operations 15

L

$LANG,
X environment variable 28
LaserWriter8 extension 88
libappletalk.a library 76, 77
license agreement file 52
license.dat 113
license.dat file 21, 53
license file 113
format 123
placement options 121
license files,
combining 137
obtaining 120
license finder 135
license management,
preventing conflicts 136
troubleshooting 153

license manager 22
components 112
configuration problem messages 149
installation package 31
installing 22
placing components 120
quick start 112
setting up 114
setting up multiple servers 118
starting 22
utilities 51
license manager daemon (lmgrd) 113
license requests 113
placement options 121
licenses,
types 133
LinkAliases 71
LinkAliases directory 54
lmgrd 22
LM_LICENSE_FILE variable 21, 121, 135

M

macd daemon process 55
.mac directory 69
contents 54, 71
specifying a default 5 4
specifying location 25
Macintosh Process Manager 55
.MIV_file 54, 71
default 52
Mac OS ROM 60
MacTCP DNR 70
MAE,
components 69
customizing 2 4–48
disk space 69
functional description 55
graphics 56
iconifying 25
improving performance 36–37, 38
running multiple concurrent sessions 87
sharing through NFS 22
MAE architecture 50–55
Mae.appleDirectory. *See* X resources
mae command 55
options 24–25
MAE command options 24–25
MAE documentation 16
MAE Enabler 70
MAE EtherTalk extension 88
MAE General control panel 32
Mae.geometry. *See* X resources
MAE Guide help 17
Mae.iconic. *See* X resources
MAE independent volumes. *See* MIV
MAE information on the Web 17
Mae.installColormap. See X resources
Mae.maximumDepth. See X resources
mae process 55
mae program file 51
Mae.system.folder. See X resources
Mae.title. See X resources
MAE Toolbar help 17
MajorUpdate file 52
man pages,
directory 53
memory 60
68K scaffolding 60
adjusting with the -memory option 61
application code 61
graphics buffer 61
insufficient memory message 47
Mac OS system 61
native UNIX programs 60
physical 63
requirements by components 60
resident memory set 63
swap space 63
swap space availability 63
usage 60
Memory control panel. *See* control panels, Memory
memory size,
specifying for applications within MAE 25
menu bar, flashing 101
MIV 40–43
defining 42
maximum number 103
renaming 43
troubleshooting 103
unmounting 43
MIVAliases 71
MIVAliases directory 54
MIV_file,
creating 42
deleting 43
moving 43
mmap system call 60
modifier keys 14
modinfo command 79
mouse buttons 14
mouse movements, adjusting 100
mouse operations 15
Movie Player directory 51
multiple concurrent MAE sessions 87

N

Network control panel 81, 88
Network extension 88
network file system performance 105
network interfaces
  switching 81
Network Visible Entity 81
NFS mounts
  configuring 29

O

online manuals,
  installation package 31
opcodes 67
OpenDoc, installation package 31
options file 139

P

packages for installation 31
paging activity,
  checking with vmstat 64
permission problems 105
permissions,
  UNIX and FileShare compared 84
permissions with file sharing and UNIX
  potential problems 84–85
personal file sharing 82–87
  controlling access to folders 82
  limiting owner's access 83
  permissions, determining 84
  sharing folders concurrently 87
physical memory 63
preferences,
  distibuting common preference files 35
processor performance 38
Processor Slow Down feature 37

Q

QuickDraw 56

R

Read Me file 16
README.TXT 16
rebuilding the desktop 44
  command option 25
  using shortcut keys 44
Reduce Color Flash 62
remote cells 107
remote installation 19
resident memory set
  definition 63

S

.SMlock 70
SANE compatibility 25
screen size, troubleshooting 101
security risks with file sharing 85
setting up a server for license management 115
shared memory 37
Sharing dialog box 85
sharing MAE through NFS 22
Sharing Setup control panel 83, 88
shmsys parameters 48
Shutdown Items 70
SimpleText application 51
si (swap in) number 64
sm.vpram 71
sm.vpram file 54
so (swap out) number 64
sound,
  AUDIO environment variable 96, 97
  Audio Server (on HP) 96
  HP implementation 96
  MAE Sound Manager 96
  mute button 98
  sampling rate 97, 98
  sampling rate setting 96
  SPEAKER environment variable 96
  troubleshooting on HP 97
  troubleshooting on Solaris 98
  volume control 98
sound command option 25
Sound control panel 96
SPEAKER environment variable 96
streams,
  for running AppleTalk 76
  verifying 79
strvf command 79
swap space 63
swapinfo command (HP-UX) 63
symbolic links 43
System Folder,
  characteristics unique to MAE 34
  contents 70
  created from apple/sys 34
  customizing 34
  forcing creation of a new one 34
  specifying with command option 25
System Folders,
  customizing new System Folders 35
system resources 72

T

TB_SOUND_OVERRIDE_CHECK 97
Temporary Installer Volume. *See* TIV
terminology 14
third-party software directory 51
timeout,
    specifying 25
TIV 33
    creating 32
    installing application software 32
    making persistent 33
    using 32
TIV_pref file 33
too many open files error 104
top command (HP) 63
Trash, can't empty 100
troubleshooting 99–110
    AppleTalk, Chooser's Zone list missing 91
    AppleTalk router comes up 92
    booting HP-UX from the saved kernel 93
    insufficient memory 47
    licenses 153
    reinstalling if streams is corrupted 91
    restoring network extensions 91
    using network diagnostic tools 91

U

UNIX directories,
    potential security problem with file sharing 85
UNIX files,
    space character 105
Users & Groups control panel 82, 88

V

vmstat command 64

W

Web page for MAE 17
windows, resizing 101

X, Y, Z

$XAPPLRESDIR,
    X environment variable 28
$XAPPLRESLANGPATH,
    X environment variable 28
xcmap command 62
.Xdefaults file 29
xdpyinfo command 62
X modifier map 102
X resources 26
    environment variables 28
    for command-line arguments 26
    loading order 28
    Mae.appleDirectory 2 6, 27
    Mae.geometry 26
    Mae.iconic 26
    Mae.installColormap 26
    Mae.maximumDepth 27
    Mae.systemFolder 27
    Mae.title 27