CHAPTER 4

The OPEN LOOK File Manager

The OPEN LOOK GUI is unique among X11 Graphical User Interfaces in specifying the detailed behavior of important client programs such as the File Manager. With MIT X11, there is no comparable program. With Motif, there may or may not be a version of one of several competing file managers or “desktop managers” like *X.Desktop* or *Looking Glass*. But with any implementation of a complete OPEN LOOK environment, there should be a File Manager program, and it should behave as described in this chapter.

The File Manager is an important tool for users accustomed to icon-based systems such as the Macintosh and Microsoft Windows, as well as to those unaccustomed to command-line oriented systems with hierarchical directories, such as UNIX. The File Manager is also a good illustration of how to use the standard OPEN LOOK “drag and drop” facilities.

Sun documents the File Manager as part of the DeskSet suite of tools, and by this logic it should be described as part of Chapter 7, *The OpenWindows DeskSet Clients*. However, we feel that a program of this sort is of sufficient import to justify its own chapter, so here it is.

4.1 What is a File Manager?

The File Manager is a program that makes it easy to manage your files. When you start the File Manager, files and directories are represented pictorially by small images called *icons*. You can work on these files and directories by manipulating the icons that represent them. You can think of the File Manager as an iconic directory browser, or as a visually-oriented command interpreter (or shell), or as a tool for making your OPEN LOOK system look more user-friendly. You can even think of it as a tool for turning a $5,000 workstation into a $1,000 MacIntosh™ (but this will not impress your financial advisors.)

In the Sun environment, it is also an important part of The X Window System environment in that it enhances some of the other DescSet tools, which therefore don’t need, and don’t have, very fancy file/directory browsing capability; you can drag mail messages to directories in the File Manager, or drag files from it to PrintTool, and so on. Also, under Solaris 2, File Manager interacts quite well the the Volume Manager, so that you need only insert a floppy or CD-ROM (and if it's a floppy run the *volcheck* command or the *File->Check For Floppy* menu item, and you will see a new directory view of the floppy. You can also format or relabel floppies, unmount and eject them, etc.

The File Manager is all these, and more. Since File Manager is a visual program, let's start with a picture. Figure 4-1 shows a screen with a File Manager running, and a row of icons.

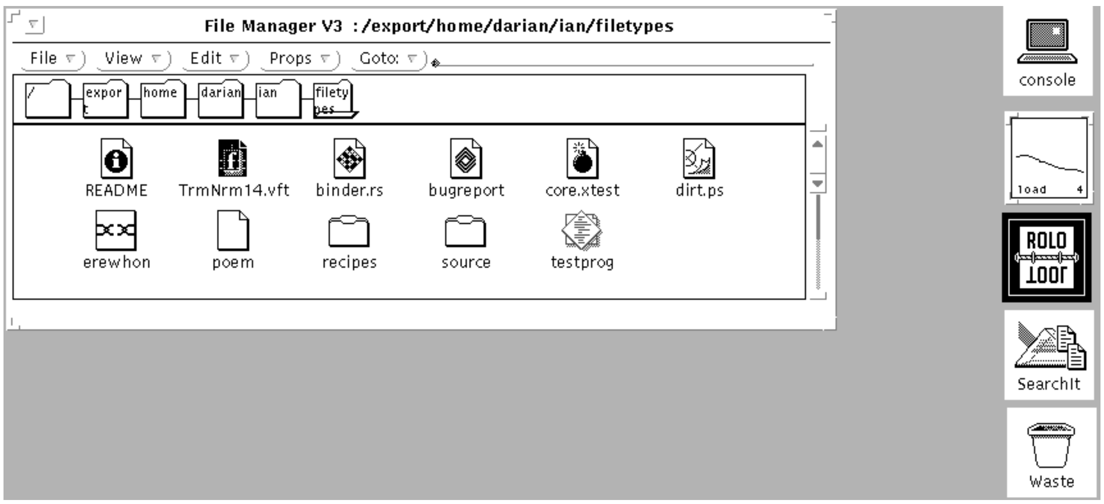

Figure 4-1. The File Manager

This gives an overview of some features the File Manager offers: a menu bar of pull-down menus to control its operation, a map from the "root" of the filesystem ("/") to the current directory, and a display pane showing the name and an iconic picture for each file object in that directory.

The icons *inside* the File Manager window represent *files*, some of which may be executable programs waiting to be run. The icons *outside* the File Manager, that is, on the Root Window or Workspace, represent *running programs*. This distinction is important. Unlike the MacIntosh Operating System, neither UNIX nor the OPEN LOOK GUI have the notion of a single current application; each known type of file will automatically be opened by the correct application when you double click on it.

In this example directory, Table 4-1 lists what is in each of those files.:

Table 4-1. What's in a Name?

Filename	Data type
README	English text
TrmNrm14.vft:	font definition
binder.rs:	rasterfile, 64x64x1 standard format image

Table 4-1. What's in a Name?

bugreport	Frame Maker document
core.xtest	core file from xtest
dirt.ps:	PostScript document
erewhon:	broken symbolic link to badsym
poem:	english text
recipes:	directory
source:	directory
testprog:	executable shell script

In short, the File Manager shows a pictorial representation of all the files. Note that all file icons have the same overall shape — that of a page with the corner folded down — but those whose contents are identifiable have a more detailed picture inside. One exception is executable programs which are shown with a special icon (see above *testprog* in Figure 4-1).† And all directories are shown by pictures of a file folder. File Manager uses this analogy throughout; directories are called "folders" in the File Manager screens and documentation.

But there is much more to the File Manager than is shown here, including the ability to drag-and-drop a file to another application. File Manager provides a visual metaphor for such tasks as navigating the directory tree, moving or copying files from one directory to another, deleting files, and many other operations.

If you are an old-time UNIX diehard, you may dislike such visual imagery, and prefer the command-line mode of using systems. You'll probably find some familiarity with the File Manager useful for drag and drop occasionally, but you may just want to skim this chapter.

4.2 Common Operations

If you see an icon representing a file and you want to do "the obvious" operation on it, then all you have to do is double-click SELECT on its icon, and the appropriate action will happen. For example, the file *poem* in this directory can be edited just by moving the pointer onto the picture of a file labelled "poem," and double-clicking the SELECT button. An editor will be started with *poem* as the input file. Let's look at some of these mouse operations in more detail.

† In OpenWindows 2.0 they were shown as a tiny OPEN LOOK window since they will run in a window. (IBM-PC and Macintosh users may choose to think that executable programs have been drawn as a picture of a floppy disk).

4.2.1 Running programs

Double-clicking SELECT on just about any file's icon will cause a program to be run. If you click on an icon that represents an *executable* object — either a shell script or a compiled program — the File Manager will try to run it for you. On the other hand, clicking on a file whose type is *bound* to some application will cause that application to be run. *Binding* a file to an application means that all files of a particular type will be run through that application when selected from within the File Manager. This *binding* between files and actions is just a name for what happens when you click on a file's icon. For example, files of type "worksheet" might be *bound* to the Lotus 1-2-3 application. This means that when you double-click on the icon for a "worksheet" file, the File Manager will automatically invoke Lotus 1-2-3 on that file.

Many standard types are bound for you automatically, including many of the standard commercial applications available on the Sun platform. We'll see some of these in the following paragraphs, and how to change the bindings (or even add your own) in *Section 4.4, "Customizing the File Manager,"* later in this chapter.

First, let's look at running an application that is in our directory. The file *testprog* shows up as an executable program. Let's double-click on it. It gets marked as selected or busy (the icon is shown in reverse video), then the footer of the File Manager window displays this message:

```
Opening testprog; errors may display on console or cmdtool.
```

If you requested a window-based application, you will see that program's main window appear. If your application is not window-based, but is a simple command-line program, you won't necessarily see any further indication that your program is running: File Manager won't give you any; beyond the notification that it has started your program. If the program writes to the UNIX *standard output*, its output will normally appear in your Console window. In either case, File Manager is available for other work as soon as it has *started* your application; you don't have to wait for the application to finish before using other File Manager functions.

What about starting up a program by clicking on a file whose type is bound to an application? In our sample directory, the first two files, *TrmNrm14.vft*, and *binder.rs*, show icons that are associated with graphical utilities; the former is a binary font file in a form that can be edited using Sun's *fontedit* program (see Chapter 9, *Graphics Clients*). The latter is a raster file that can be displayed with the *snapshot* program, also described in Chapter 8. The file *dirt.ps* is a PostScript file, and clicking on it (or any file that is recognized as containing PostScript) will start the OpenWindows PostScript previewer, *pageview*, to view this file (see Chapter 9, *Graphics Clients*). The file "poem" is a text file, so it is considered to be bound to an X-based text editor, normally *textedit*, (see Chapter 5, *The cmdtool/shelltool Terminal Emulator*). Double clicking on any of these icons will try to launch the corresponding application for you.

4.2.2 Change Directory

Eventually you will get tired of looking at the files in just one directory. If a directory icon (picture of a file folder) is shown in your File Manager window (as are *recipes* and *source* in our sample directory), you can change directory to it just by double-clicking SELECT on it. A short form for this is to click SELECT once and hit the RETURN key. The File Manager will do a change directory operation internally, and will then read the contents of the new directory and display it, using the same iconic representation as before. Note that this change of directory does *not* affect any other shells or programs you may have running in other windows; on UNIX there is no way for one process to change another's working directory. It does, however, set the working directory of any new programs or applications that you start from within this instance of the File Manager.

Let's say you double click on the folder (directory) *recipes*. The icon for that directory will show busy for a second, then File Manager will display the new directory, and your new screen might look like Figure 4-2.

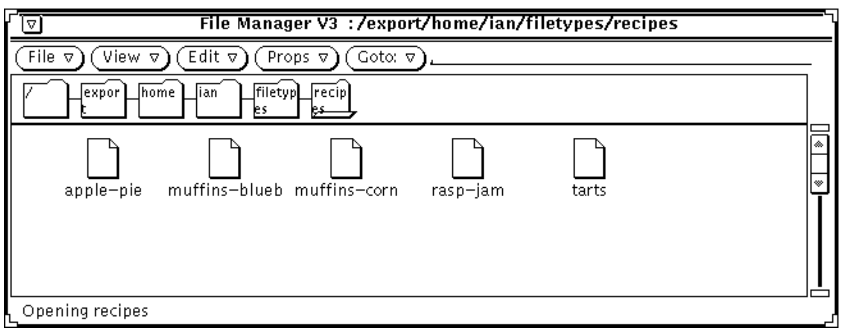

Figure 4-2. File Manager in recipes directory.

Then you might wish to edit the "tarts" recipe. Just double-click on the picture of the file labelled *tarts.* Your File Manager titlebar will show "busy" for a moment, as in Figure 4-3.

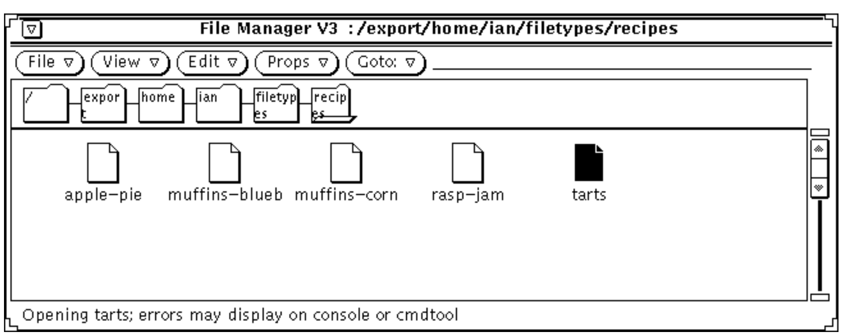

Figure 4-3. File Manager starting editor on "tarts"

Figure 4-4 shows the edit window that will appear in a few seconds.

On OpenWindows, the editor used by default is *textedit*, described in Chapter 5, *The cmdtool/shelltool Terminal Emulator* (To change this default, see *"Customizing the File*

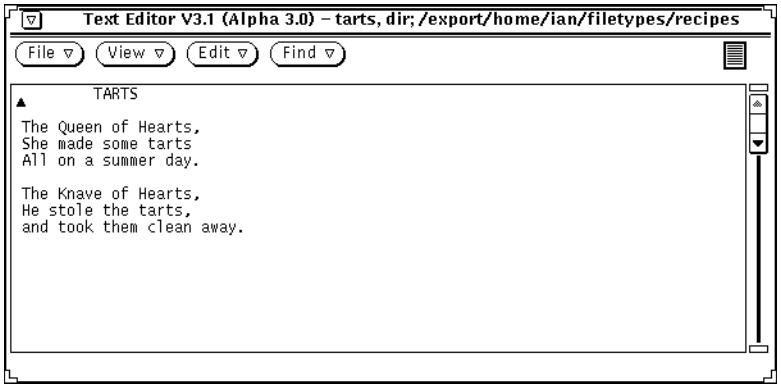

Figure 4-4. Edit window for "tarts"

Manager" below). On AT&T-OL, the *vi* editor is used. The File Manager window is ready for action again as soon as it resets its titlebar to non-busy state; it doesn't make you wait to finish editing before you can undertake new operations.

4.2.3 Moving or Copying Files

When you wish to move or copy a file, you can use the File Manager with OPEN LOOK's "drag and drop" operation. Drag-and-drop is a fancy term for "slide", and is familiar to us from a variety of similar operations. In the games of Chess and Checkers, for example, we select a player, and either lift or slide it to a new location, then release it. Computer drag-and-drop is similar: briefly, you select the icon(s) of the file(s) you want to move or copy, drag the icon into a window that represents a directory, and release the mouse button to "drop" the icon(s) into that directory.†

As an example, let's move the file *tarts*, to our home directory. First, to select a single file, *tarts,* just click **and hold**, SELECT with the pointer over the file's icon. Now if you move the mouse, you will see (Figure 4-5) that an outline of the file's icon follows the pointer:

Now that you have the icon "in tow", you can move it to any directory icon to drop it. At this point you don't have a second File Manager running to drop the file into. However, the File Manager that you are running shows icons for all the directories between the root directory ("/") and your current working directory (which in this example is */home/darian/ian/book/ol/filemgr/recipes*). So we'll move the *tarts*). file to the icon for

† This is one of OPEN LOOK's significant extensions beyond what the underlying X Window System provides. Hence it will not work correctly if for some reason you are not using the OPEN LOOK Window Manager; it will not work with *twm, mwm*, or *uwm.*

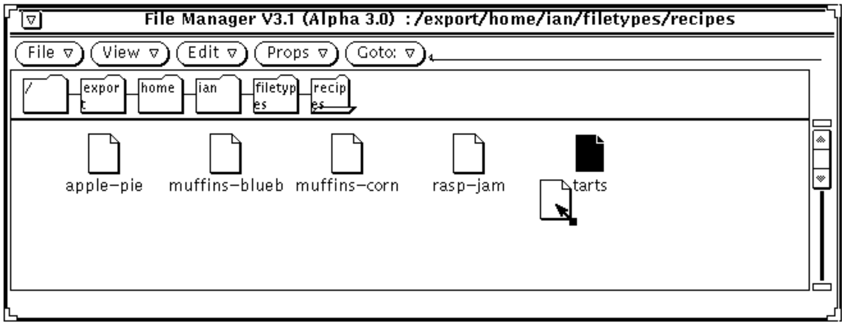

Figure 4-5. Icon outline for drag-and-drop

/home/darian/ian, my home directory, and drop the *tarts*, icon on top of my home directory's icon:

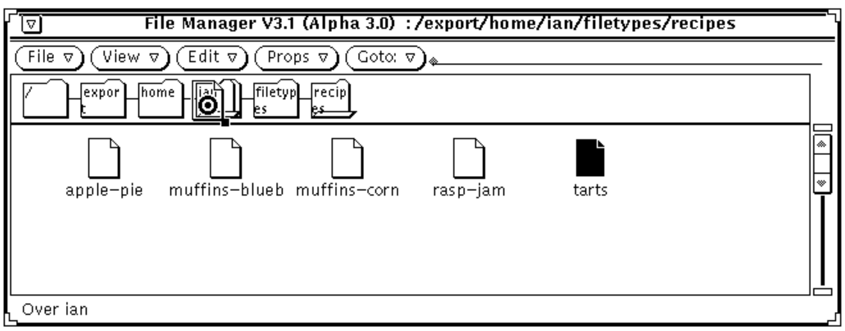

Figure 4-6. Moving the icon to drop it on a directory

Finally the File Manager screen will be updated to show that the file *tarts* is gone, and the window footer will contain the confirming message `Move to '/home/darian/ian' succeeded`, as shown in Figure 4-7:

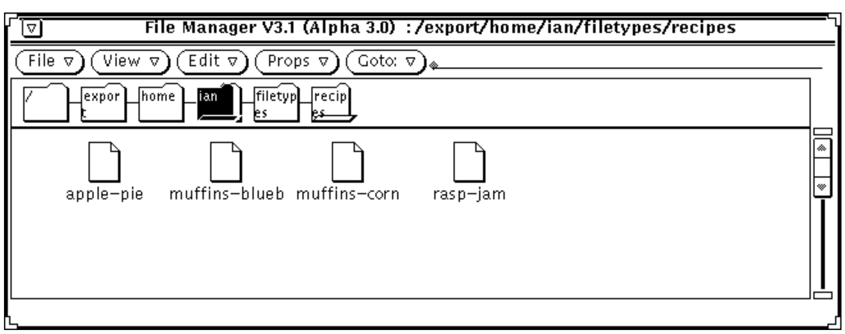

Figure 4-7. Double Confirmation: Icon gone, message appears

The move operation has been completed successfully.

To copy a file instead of moving it, all you do is hold down the DUPLICATE key (normally the Control or CONTROL key) while you click and hold the SELECT mouse button. The cursor shows slightly differently (see Appendix D, *Standard Cursors - X11 and OPEN LOOK*) to let you know that you are copying, rather than moving. Everything else is the same.

If you try to move or copy a file into a directory that is "read only" for some reason (for example, because you lack write permission on the directory, or the filesystem is mounted read-only), its icon will display a padlock icon while you hold the icon of the file you are moving over it.

On modern versions of UNIX (that is, SunOS, other BSD derivatives, and System V Release 4), you can move a directory as easily as a file, provided only that you can't move a directory from one "filesystem" or "slice" to another. On System V Releases 2 and 3, and System V Release 4 with the System V Filesystem Type, you cannot move a directory; you can only rename it within the same parent directory.

If you need to move several files, you may want to start another copy of the File Manager, and change it to the target directory. This will let you see what files exist in both directories before and after each move. And in this case you can just drop the icon anywhere in the icon window for the target directory.

Another possibility is to select multiple icons. As before, click SELECT (and release it) to select the first, and click ADJUST to select each additional icon. Then, click *and hold* SELECT, and you will get a "stack of icons" outline, which you can move to the target directory and release.

4.2.4 Deleting Files

Deleting files is just a special case of moving files. When you start the File Manager, it creates a separate window labelled *WasteBasket*.† The window is iconified, but you can still "drop" files into it.

If you select a file, drag it into the WasteBasket, and drop it, it is removed from the directory it was in. What could be easier?

But unlike the normal UNIX command for removing a file, this operation does not physically remove the file until one of the following occurs:

- You exit the File Manager, in which case you are asked if it's OK to remove the file(s) in the WasteBasket, and you click on "Yes."
- You change your mind about removing the file, double-click on the WasteBasket folder to open it, and move the file back where it was!

† On Sun OpenWindows Version 2, the WasteBasket icon sometimes appears with no icon label or image when it first comes up, so it will appear as a blank icon. But it still functions correctly.

4.2.5 Renaming Files

You can change the name of any file or directory in one of two ways. The quickest way is as follows:

- Click SELECT on the filename; the File Manager will underline the name.
- Treat the underlined filename as a text field; click on it, and then you can type a new name. Or, you can clear part of the name by clicking SELECT and clearing from there to the beginning with Control-U and type a new name, or delete or insert characters at will.†
- Then just hit RETURN.

This method is also used to assign names to objects created using *Create Document*; or *Create Folder*. The longer way, which you'd only use if you wanted to change other properties at the same time, is by selecting *NewFolder* and using the *File Properties* menu described in the next section.

For example, if we wish to create a folder or directory named *mideast*. in our example directory, the File Manager window will look like this after we create the folder:

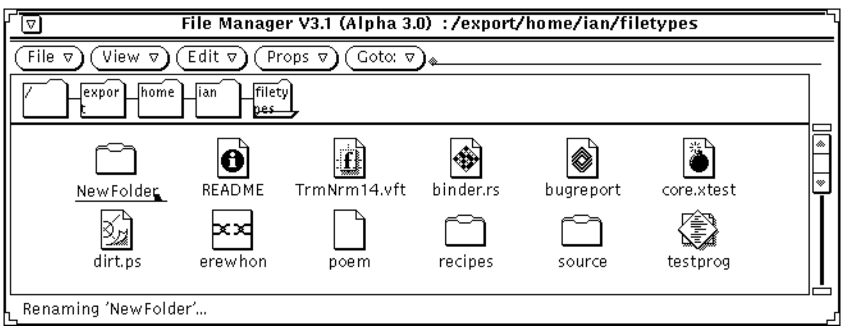

Figure 4-8. File Manager screen after Creating NewFolder

Now we need only click SELECT at the point we wish to change. Most commonly we want to erase the characters *NewFolder* completely and type a new name. So we click SELECT

† For a list of other editing keystrokes, see the discussion of *textedit* in Chapter 5, *The cmdtool/shelltool Terminal Emulator*.

at the right side of the existing name, type Control-U, and type the new name as shown in Figure 4-9:

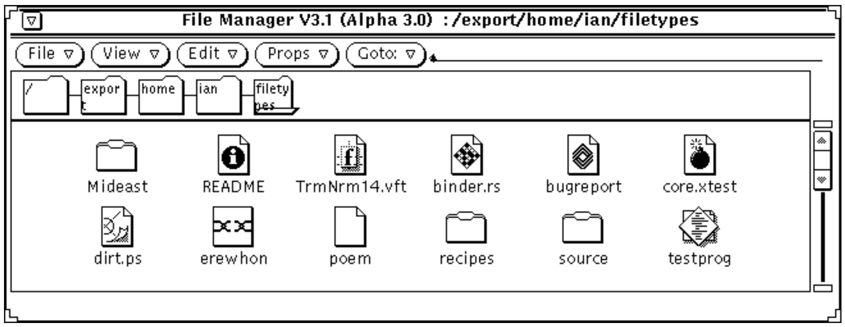

Figure 4-9. Directly renaming NewFolder to mideast

The rename will take effect as soon as you either hit RETURN or click SELECT elsewhere to take the focus away from the filename's text field.

Note that you can rename any object, not just a newly-created one, using this method. The trick is that you must first select the icon, *then select the filename* underneath the icon to mark it for renaming, then type the new name you want for it.

4.2.6 Displaying and Changing File Attributes (Properties)

This section demonstrates one of the clever ways in which the designers of OL worked to make it easy to use. To change any of the attributes of a file, you just call up its properties sheet. This replaces a whole grab-bag of operating-system-dependent utilities[†] with one simple, consistent menu. To display all the information about a file, simply select it, and

† On UNIX at least *ls -l, mv, chown, chgrp,* and the complexity of *chmod*.

click on the *Props* menu button (or drag it down to *File Properties...*). Figure 4-10 shows the properties sheet for a simple file.

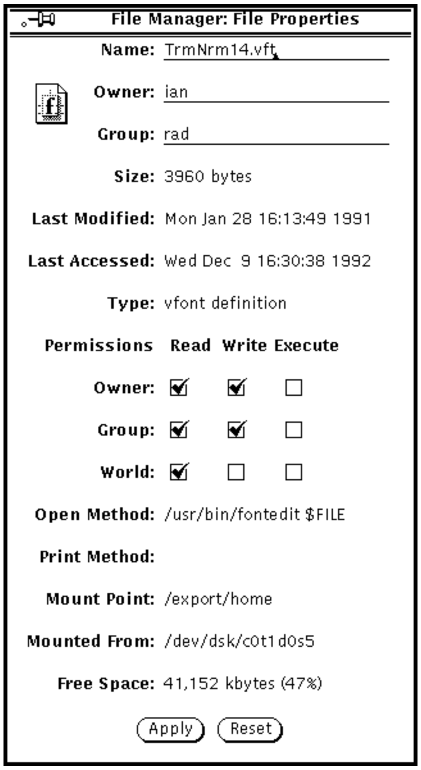

Figure 4-10. File Properties Sheet

If all you wanted to do was examine the file information, it's there. Read it, then with the pointer in this window press the Iconify button (L7), or select *Dismiss* from its window menu. If you want to change the attributes of the file, change them and click on the *Apply* button. To change the name of the file (rename it), you just type a new name in the "File name" text field, and click SELECT on the *Apply* button. To change the owner or group, type the name of a new group in the appropriate text field, and click APPLY. Of course to do this you must have sufficient privileges (own the files or be *root)*. To change permissions, click SELECT on the Checkboxes for read, write and execute for yourself, people in your group, and others on the system. At any rate, once you've specified any or all changes, you just click SELECT on the APPLY button to make them to take effect. If you have selected multiple files, then any properties that don't apply to all of the selected files are grayed out.

4.3 Menu Bar Operations

The Menu Bar, or row of buttons at the top of the File Manager window, gives you access to a large variety of operations. These buttons are named *File, View, Edit, Props,* and *Goto.* In this section we'll discuss those items that cause actions to be done for you: File, Edit, File Props, and Home/Goto. We'll discuss *View* and *Tool Props* later, under the heading of *Customizing the File Manager.*

4.3.1 File Menu Operations

The *File* menu pulls down to offer you a choice of *Open*, *Print File*, *Create Folder*, *Create Document*, *Remote Copy...*, or *Your Commands*.

4.3.1.1 Open

The *Open* item is only active if a file is selected in one of the windows. This selection in turn pulls down to *File, With Goto arguments,* or *In Document Editor*. These let you open a file in various ways. *File*, which is the default if you just select *Open*, opens the file with whatever program is appropriate. A "worksheet" will be opened in Lotus 1-2-3; a Sun Raster file will be opened with *snapshot* program (described in Chapter 9, *Graphics Clients*), and so on. If you wish to pass command-line arguments to the given program, you can enter then in the *Goto* textfield and then use *Open* with *With Goto arguments* to pass the *Goto* field's contents as command line arguments. Lastly, if you wish to open the program using the default document editor (normally *textedit*), you can *Open* it *In Document Editor*, which overrides any program bindings for the file.

✗ Note that some types of files contain "binary data" and cannot be successfully edited with a text editor; if you get "garbage" characters in a file, it usually indicates that the file is binary, and you should not save it with a text editor or the file may be corrupted.

4.3.1.2 Print File

The *Print File* selection does just what it claims to: sends a copy of the selected file to your system's printer. The printer is normally your system default printer, but this can be overridden by use of the Property sheet, described below under the section *Customizing the File Manager*. Like *Open*, this selection is not available unless a file has been selected. In particular, you cannot select this item if a directory has been selected instead of a file!

4.3.1.3 Create Folder

Create Folder is like the UNIX or MS-DOS *mkdir* command in that it creates a new directory or folder. However, it does not prompt you for a name; it creates the folder and gives it the name *NewFolder*, to let you defer choosing a real name for it until you are ready. You then rename it in a manner similar to that described above in the section *Renaming Files*; for convenience, the filename text field is already selected, so you need only select the insertion point, type Control-U, and the new name.

On the other hand, if we want to change the permissions (or some other attribute) of the newly created directory at the same time as we rename it, we can use the *File Properties*; dialog, as follows:

- Click SELECT on *NewFolder*'s icon
- Pull down *Props* and select *File Properties*, or press the *Props* (L3) key. This will cause the *File Properties* dialog to appear.
- Change the *Name* text field from *NewFolder* to *mideast*.
- Change the permissions using the check boxes.

- The *File Properties* screen should now look something like this:

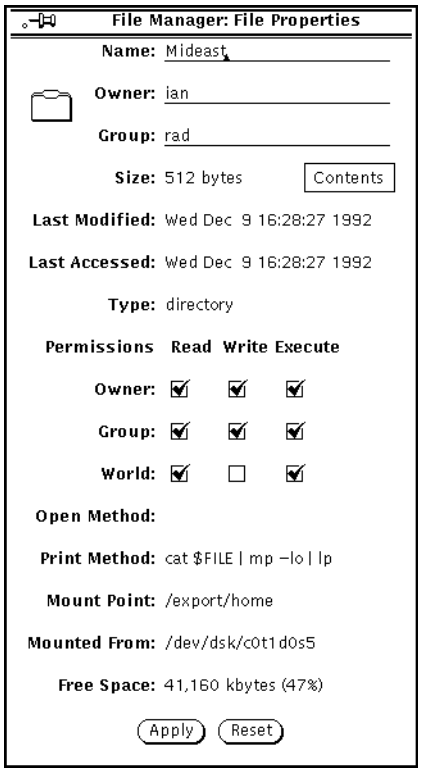

Figure 4-11. File Properties for renaming NewFolder to mideast

Click SELECT on the APPLY button; the change will be done and the *File Properties* dialog will pop down.

4.3.1.4 Create Document

How do you create a new document? Just select *Create Document*. The process is almost identical to the *Create Folder* operation just described. The new document is created as *NewDocument*, and you can edit it by clicking on its filename to set the text caret and using any text field editing, usually Control-U and type a new name. Once it's created, it behaves like any other file, except that it's created empty, so it has no contents or type. The most common operation next is to edit the new file to make it a text file.

4.3.1.5 Find

The *Find...* menu item pops up a dialog (see Figure 4-12). It can find files either by certain criteria (filename, owner, modification date, file type) or by contents.†

Let's say you wanted to look for all recently-modified text files containing the word "compendiate" Here is how you might set up the dialog, assuming that you used the extension *.txt*, for your text files:

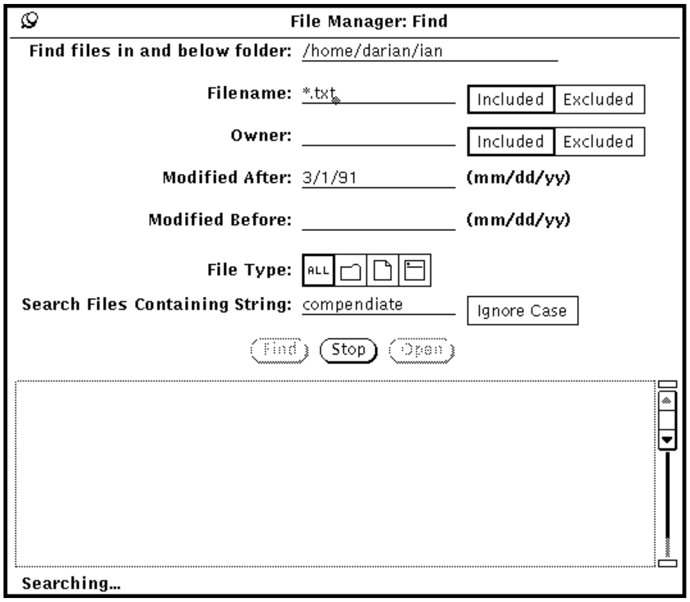

Figure 4-12. Looking for "compendiate"

Then click on *Find* to start the search. If any files are matched, they will be presented in the scrolling list in the bottom part of the window; these files can be clicked on, dragged, etc., just like other files in File Manager's display. This is shown in Figure 4-13.

4.3.1.6 Remote Copy

From time to time, you need to access a file that isn't on any machine you have a terminal emulator window on and that isn't exported for use as a network-mounted file. Then you must copy the file from the remote machine to your local system's disk. In command line mode, you would use the *rcp* utility. But in File Manager, you need only call up the *Remote Copy* selection. The *Remote Copy* operation allows you to copy file(s) to and from any machine that is reachable by the TCP/IP networking software and on which you have an account. You specify the source host and pathname, and the destination host and pathname. You can use either relative pathnames (path name does not begin with a '/' character) or absolute pathnames (path given does begin with a '/'). Relative pathnames are interpreted relative to your home directory on the remote machine, but relative to the current directory on the local machine (the machine that File Manager is running on). Let's copy a file from

† Experienced UNIX users will recognize this as similar to the *find* command in UNIX. If you need to perform a search that has more complexity than this *Find...*, dialog, you can use the standard *find* command. This function is in fact implemented using *find*. For example, when using the find function of File Manager to search in /usr/include for files containing the string TIOCCONS, we did a */usr/ucb/ps -a* in another window and saw this:

748 pts/0 S 0:00 find /usr/include -exec egrep -l TIOCCONS { } ; -follow

This way of finding strings in files is rather inefficient compared to "full text searching" as is done by SunSoft's *SearchIt* package or the contributed software package *LQ-Text*, among others.

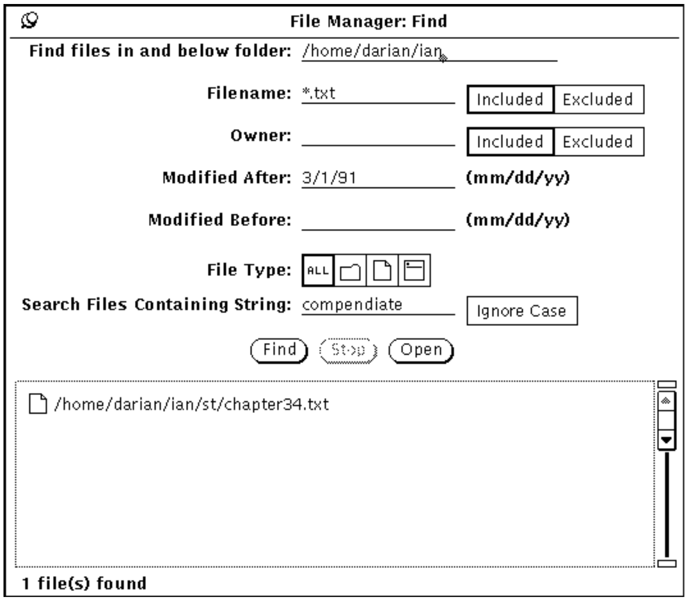

Figure 4-13. Found one file

machine *timbuktu*, to the machine we are on. Click SELECT on *Remote copy...*, and you get the pop-up window shown in Figure 4-14.

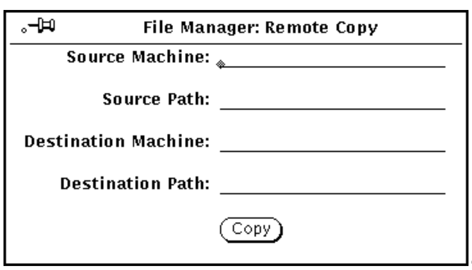

Figure 4-14. Remote copy pop-up window

The popup is pinnable, and since we might want to copy multiple files, let's pin it up. Then type the remote machine name, the filename, and a place to put it in the local directory, and select on the COPY button. It becomes busy:

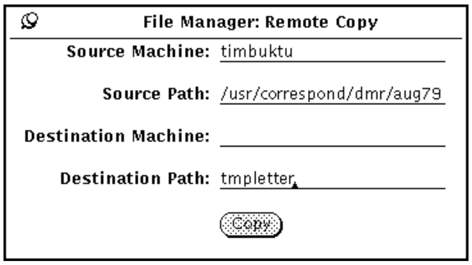

Figure 4-15. Remote copy window during copy

To copy out a file, just SELECT it in the File Manager window; the hostname and full path appear in the *Source Machine* and *Source Path* fields for you.

If there are any problems, such as "Permission denied", they will be printed in the left footer of the File Manager window, so watch that window for any error messages. If all goes well, the file you are copying to the local machine will appear in your File Manager view window after a short delay, assuming you're copying it to the same directory that the File Manager is in.

4.3.1.7 Your own commands

The *Custom Commands* menu lets you save a series of common commands, and execute them at the click of a button. This may appear to be redundant with the *.openwin-menu* of the OPEN LOOK Window Manager, but it is certainly easier to update. And it may make it easier to have different commands set up for different machines in a networked environment. To add a command to the list, just pull down *Add Command*.

You get a small dialog window with a text field in which to type a name for the command and the actual UNIX command line. When you've typed it the way you want it (see Figure 4-16), select *Apply*, and the new command will be added to your personal list. Next time you pull down the *Your Commands* menu, you will see the new command.

The other entry under this selection, *Shell*, starts up a terminal emulator window with an interactive shell running. This is rarely used, since you normally start shell windows from the *Programs*, menu of the *Workspace*, menu. However, it might be a good way to restart your window manager if it failed when you didn't have any shell windows running. Just select *Shell*, and when the shell window comes up, give the command

```
olwm &
```

and the window manager should reappear after a moment — you will see the titlebars reappear on your windows.

4.3.2 Edit Menu Operations

In most OPEN LOOK applications, the Edit menu is used to edit text. In File Manager, by contrast, it requests editing operations on whole files.

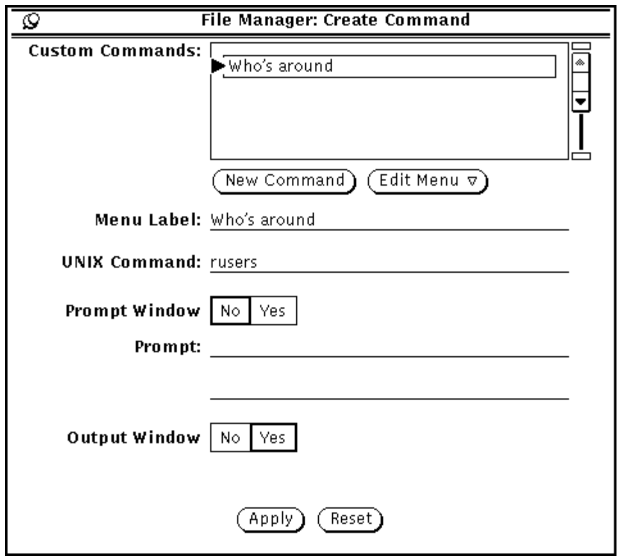

Figure 4-16. Create Commands Window

4.3.2.1 Select All

Select All, for example, selects (highlights) all the objects in the display window, so you can use them all as the target of some operation like print or move.

4.3.2.2 Link

The *Link*, selection allows you to create a *link*, to a file. A link is an additional name for the file; once you have created the link, you can refer to the file by either name. On Sun's OpenWindows, the new link is named 'filename.1' (or .2 if .1 already exists, and so on).

If your system supports "symbolic links", note that File Manager will create a regular link if the source and target directories are on the same filesystem, otherwise it will create a symbolic link. If you are on System V Release 2 or 3, you can ignore this distinction.

Symbolic links behave a little differently than regular links. When you add or remove a regular link, the file's link count is incremented or decremented. If the link count reaches zero, the file's data blocks are discarded by UNIX and the file ceases to exist. Symbolic or "soft" links are actually files that contain the real *name* of another file; when you "open" (to read or write) a file that is a symbolic link, the UNIX kernel reads the symbolic link and then tries to open the other file by name. A "symlink" can therefore do two things that no ordinary link can do:

- it can cross disk partition or "slice" boundaries.
- it can be used to point at directories.

The drawback of symlinks, however, is that if the file they point to is renamed or removed, the symlink still points to it; it will show up in an *ls* listing, for example, but you won't be able to read it or write it, which is confusing. Such misleading links are called "broken" or "dangling" symbolic links, and are shown by File Manager with the "broken chain" icon used to represent the file "erewhon" in Figure 4-1.

4.3.2.3 Copy

Copy works just link *Link*, except that it copies the file instead of linking it. That is, after a *Link*, you have only one copy of the file on disk, accessible under two different names. But after a *Copy*, you have two different files (though they initially have the same contents), each with its own name.

4.3.2.4 Cut and Paste

Paste and *Cut* work together to move files into and out of the directory. Select a file and *cut* it, and Poof! It's gone! Where did it go? By default, it goes into the WasteBasket folder, just as if you had dragged it there and dropped it. But *Cut* can be *pulled right* (hold the MENU button and drag the pointer to the right to pop down a new menu) to produce a menu asking if you want to *Cut to Wastebasket*, (the default), *Move to Clipboard*, or *Really Delete*. The third item tells the operating system to unlink or remove or delete the file, while the middle item moves it to a special folder called "the Wastebasket," from which the file can be retrieved later by moving it into a (possibly different) directory. The *Show Clipboard* item shows what has been cut into the File Manager Wastebasket.

4.3.3 Goto Menu Operations

To help you move around quickly by directory name, the *Goto* menu lists your home directory and several of the other directories that you have visited most recently. Sliding the pointer down this menu and releasing on any one of these will make it the current directory. Initially, the default is your home directory, so you can get to your home directory just by clicking SELECT on the *Goto* selection item.

*But n*otice that there is a text field to the right of the button. If you type a path name in this text field, then click SELECT on the *Goto* button, that directory will be made your current directory. If you click MENU on the button, you will see two items. One is the default, the contents of the text field (or as much of it as will fit on the display).

This field has another use, too. You can use it to specify a range of files to operate on. This has the effect of "wildcards" in the UNIX shell environment. For example, to select all the PostScript files in a directory, just type the string **.ps* in the *Goto* field and then press Return or Enter, and all the files that match will be marked selected. Then if you want to delete them all, just hit the CUT key (L10). Or, drag any one of the selected icons onto the trash can (or any other drop target, such as print!), and *all* the files will be moved or copied as appropriate.

In fact, you don't even need to use this field. Just move the pointer into the file icon display pane and type your pattern there. The pattern will appear in the footer as you type it, and as each character is typed, any files that match will be selected. This is faster, but more exacting, as you don't have the textfield's editing abilities at your disposal.

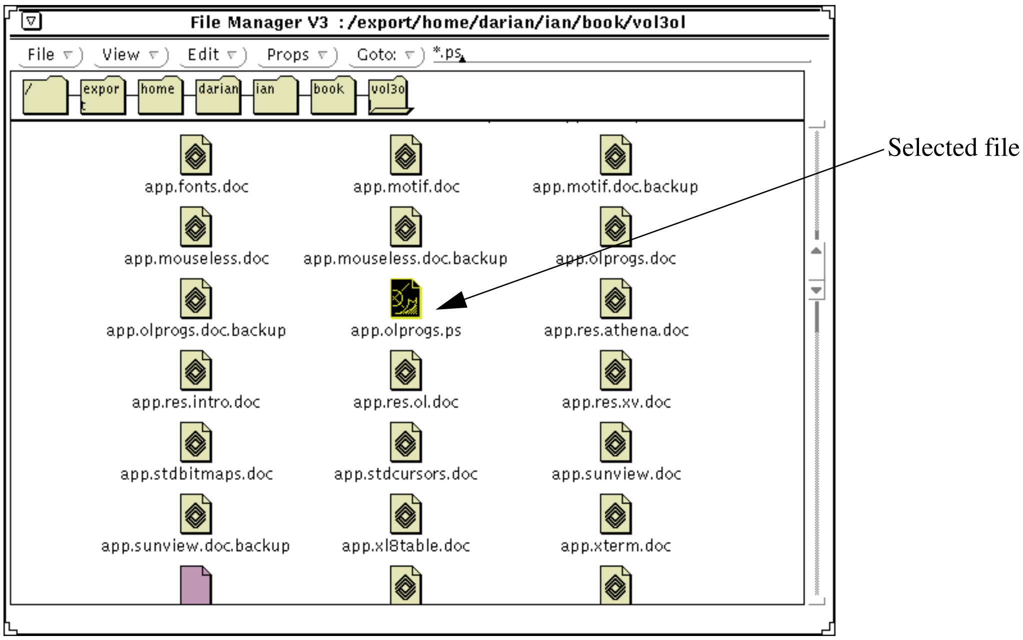

Figure 4-17. Wildcard selection in File Manager

4.3.4 MENU button in the display

As a convenience, there is a menu attached to the MENU button while in the File Manager window that contains some of the most common items of the ones we've seen. The list includes *Select All*, *Cut*, *Copy*, *Paste*, *Delete, Print*, and one called simply *Up*. Except for the last one, these all do the same thing as they do when activated from the top-level buttons in the menu bar. The last one does what you might expect from its name; it moves you up one directory. It is inactive when you are in the root directory ("/"), since there is nothing above that directory.

4.3.5 MENU button in the Wastebasket folder

If you open the *Wastebasket* icon, you see a folder display for items that have been moved or cut into the wastebasket. There are no menu buttons, but the MENU pointer button has a menu similar to the menu just described. It has one important addition; the first item is *Empty Wastebasket*, which actually deletes the files in the wastebasket—files so removed are *really gone!* This is useful when your hard disk is running low on space

4.4 Customizing the File Manager

The overall behavior of the File Manager is fixed. But you can change the details of its behavior at several levels. First are the *View*. controls on the top row. Second is the Properties sheet. The third level of customization is in configuration files.

4.4.1 View Menu Button

As we've seen, the menu bar items are *File*, *View*, *Edit*, *Props*, and *Goto*. Most of these were discussed above. The File Manager *View* menu items control what you see, or your view of the files. Unlike many of the other buttons, changes you make here are saved in a file for later File Manager sessions, that is, they stay in effect until you change them again.

The first item is either *Path* or *Tree*, initially *Path*. When the setting is *Path*, the "parent directories" window shows a single row of icons for the path to the current directory. When the setting is *Tree*, however, you get a visual presentation of the parent directories and all the parallel directories, as shown in Figure 4-18

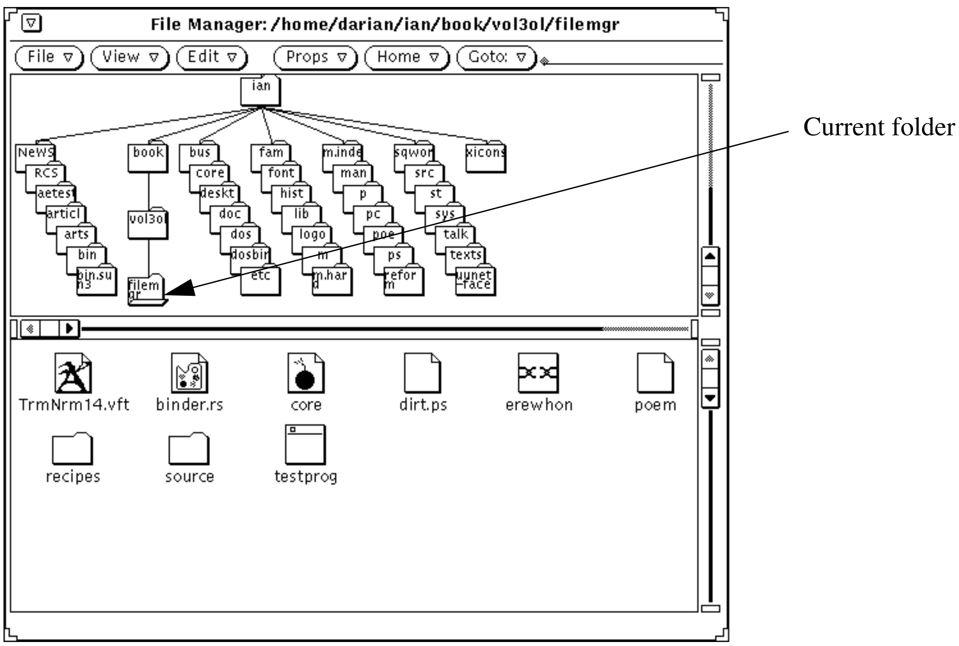

Figure 4-18. View set to Tree shows the directory tree

Now it may happen that the directory hierarchy that is shown on the screen gets large and complicated, if you have lots of subdirectories. To help you remember where you are, the currently-open folder is drawn with an "open folder" icon (see Figure 4-18) to remind you of your place in the overall directory hierarchy.

Each icon in this window is "active." That is, you can double-click on it, drop a selection into it, etc., in any way that makes sense. The next few items, *Hide Subfolders*, *Show All Subfolders*, *Begin Tree Here*, and *Add Tree's Parent*, are used to control the tree display, but only if the mode is *Tree* **and** you have selected at least one directory with subdirectories (folder with subfolders). *Hide Subfolders* and *Show All Subfolders* control whether subdirectories of the current (or selected) folder will be shown. Note that selecting *Show All Subfolders* is a recursive operation. That is, it shows all the subdirectories, sub-subdirectories, sub-sub-sub-directories, and so on. For this reason, the left footer message is set to "This operation may take some time" while the subdirectories are being found. But you can get some fantastic displays of your system's directory tree. Try this: before going to lunch, and running File Manager on your own workstation, select the root directory, and click on *Show All Subfolders*. When you get back, there will be vertical and horizontal scrollbars for the view window, and you will be able to see the entire UNIX filesystem just by moving the scrollbars. Of course, you can also resize the window wider to see more of the display at one time:

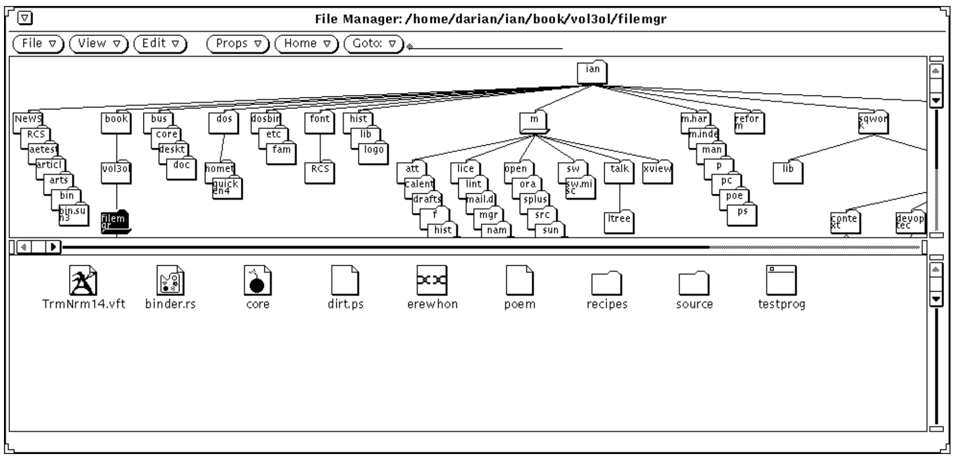

Figure 4-19. View Tree made wider

Begin Tree Here and *Add Tree's Parent* are used to truncate the tree (so that everything above the selected directory is removed from the top of the display) and to extend it upward by one directory level, respectively.

The menu has several choices to specify how files and directories are to be sorted and displayed. The types are either *List* or *Icon*. *Icon*, which is the default, uses the large file and directory icons shown throughout this chapter. *List* uses smaller icons like those shown in Figure 4-20. The *Icon* display can be sorted by name or by "type", which sorts all entries of a given type together. The *List* display can be sorted by name, type, size (largest first), or by modification date.

These smaller icons show less information than the larger ones; they only identify the object as "file," "folder" or "other." (Compare *erewhon*'s look here with its look previously.) However, the small icons allow you to show more files in a given space (useful in a large directory), and you can still select them, drag-and-drop them, etc., just as you can the full-size icons.

Which format you choose — large or small, sorted by name, or by type, or by date — is completely up to you. It's a matter of taste. Try them out from time to time, and see which makes most sense to you.

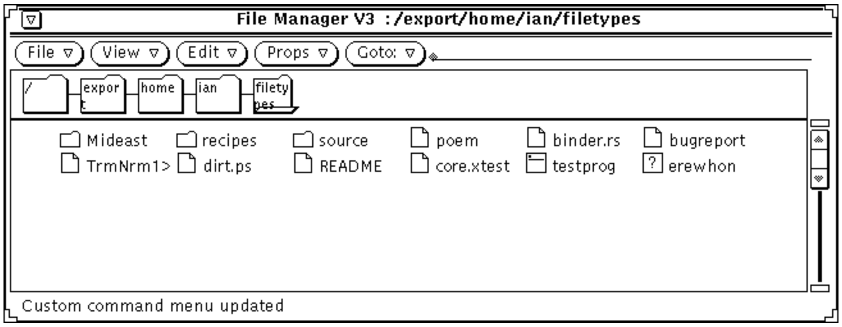

Figure 4-20. Folder Display set to List

4.4.2 The View/Customize Properties Sheet

There is a "Properties"-like window under the *View* menu, named *Customize View* (See Figure 4-21). Many of its items are similar to those of the *View* menu discussed previously,

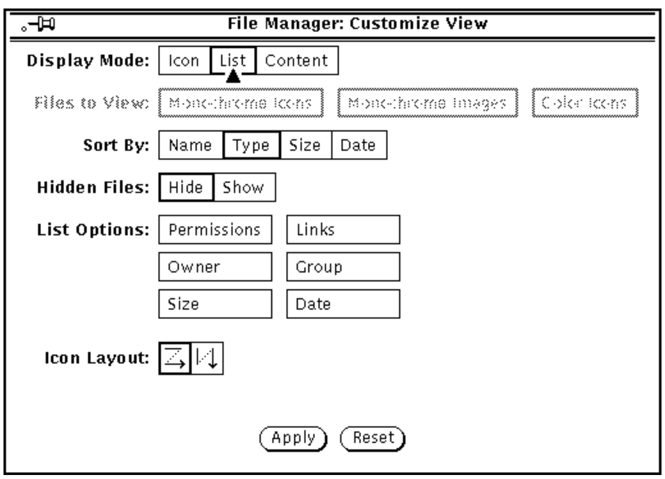

Figure 4-21. The Customize View Window

but the window also lets you control whether hidden files ("dot files", whose name begins with a period or dot) are displayed, and whether icons are layout out in rows across the screen or in columns down the screen.

You can also select *Content* as the Display Mode setting. This causes raster and icon files in known (Sun) formats to be displayed as themselves, at a slight cost in speed. However, at least in Version 3.0, this display is garbled if the raster files are of odd sizes.

If you select *List* as your display mode (either on the Customize View window or in the View menu), the Customize View window lets you control the display of the information equivalent to that of the UNIX command *ls -l*. You can enable or disable individually the viewing of permissions, size, owner, group, link count, and modification date. Figure 4-22 shows the display of just permissions and file size, along with the names which are always displayed.

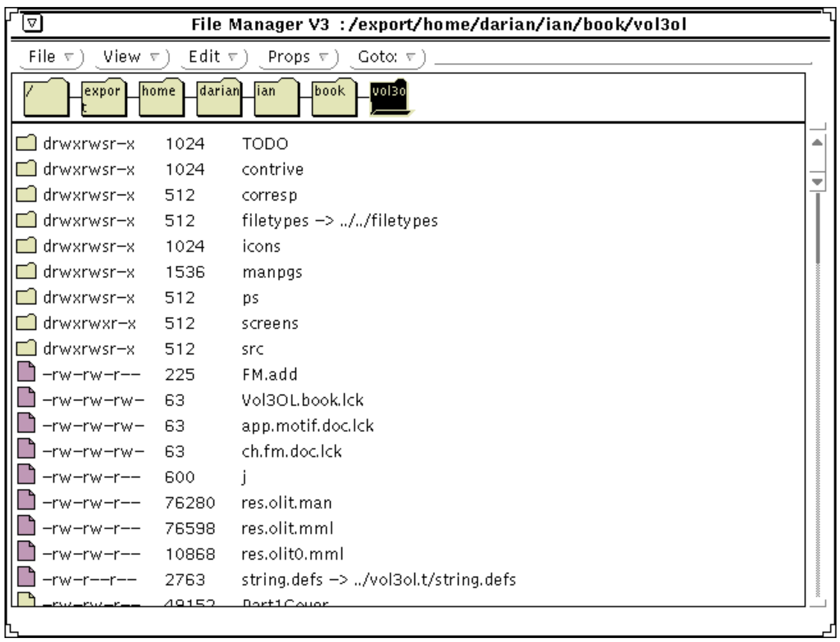

Figure 4-22. Folder display with permissions and size only.

4.4.3 The Tool Properties Sheet

You display the Properties sheet for the File Manager by pulling down the *Props* top-level menu and releasing on *File Manager*.The resulting screen looks like Figure 4-23 on Open-Windows.This gives you control over the optional features of the File Manager's behavior:

- the default print command
- something called the *View Filter*, which allows you to only look at certain filenames, for example, "**.txt.*"
- the longest filename to be displayed
- whether deleted files are kept in the *Wastebasket* folder, or actually unlinked
- selecting the default editor for documents.

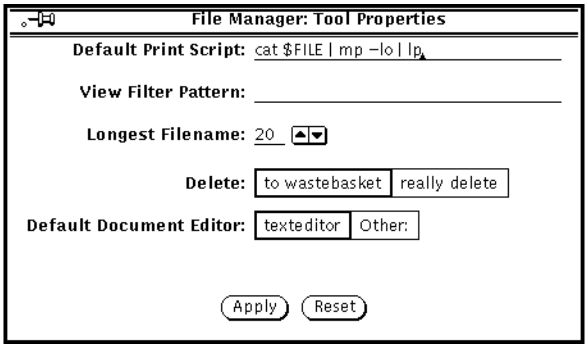

Figure 4-23. The File Manager Property Sheet

4.4.4 Customizing File Bindings

The third level of customization lets you control the way File Manager reacts to what it fins in your directories. Many of the responses are controlled by a series of configuration files. Unfortunately, the format of these configuration files does depend on which implementation you are using, AT&T's or Sun's.

4.4.4.1 Binder: Customizing the File Manager in OpenWindows

The OpenWindows version of File Manager uses a package called the *classing engine* to control the display of different file types. The Classing Engine is used by File Manager (and other DeskSet programs), as well as the OPEN LOOK Window Manager, to classify files. There are three sets of classing engine files; one "network wide" one for all files served by a given NFS server, one "system-wide" file (in the */etc* directory on each machine), and one personal one (in your home directory).

Now updating these files is not for everybody. You only need to do this if you have a new type of file that the Classing Engine doesn't know about. Some vendors' installation software, for example, updates the files automatically, without running the binder. There is a command-line interface to the Classing Engine files; see the man pages for *ce_db_merge* in OpenWindows for details.

You normally update the Classing Engine files using a program called *binder*. Once you've started *binder*, you should see something like Figure 4-24 in the *binder* dialog box.

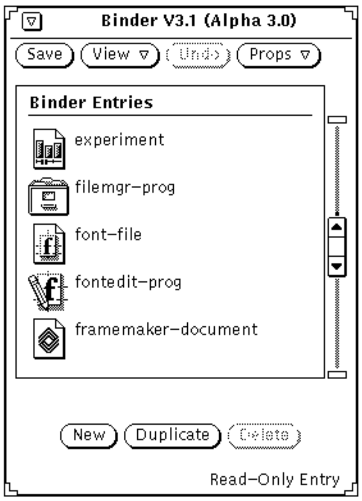

Figure 4-24. Binder dialog

The initial screen is a scrolling list of icons (the same ones used in the File Manager display) and the associated file type. Once you have selected one, you can customize either its icon properties or the filetype properties. Figure 4-25 shows using the *Icon* properties to change the color that a given icon type is displayed in.

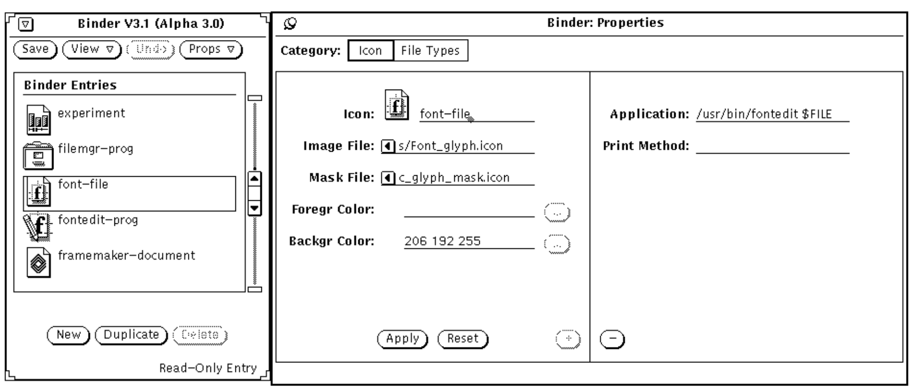

Figure 4-25. Binder Icon Properties Dialog

Users of OpenWindows Version 2 should note that the file layout is totally different, but use of the *binder* program is similar to that described here†

† .There is a bug in the OpenWindows 2.0 version that you should know about if you've read this far. If your file refers to an XBM file rather than an icon-format file, then *olwm* will still process it, but *binder* will reject the entry with a message like "icon file /home/darian/ian/xicons/xterm parse failure*" and will delete the entry if you save your file*.

Whether you update the system-wide file or your own personal file, you must restart the File Manager and the OPEN LOOK Window Manager before your changes will affect these programs.

We've now shown how to use the File Manager to navigate your directory hierarchy, to select files and operate on them in various ways, and showed several different methods of customizing this versatile and valuable tool. The next chapter discusses the terminal emulator and text editing under OpenWindows, after which we turn our attention to the OPEN LOOK Window Manager.