CHAPTER 6

More about the OPEN LOOK Window Manager

The OPEN LOOK Window Manager, *olwm*, is one of many window managers available for X11. However, if you are to make full use of OPEN LOOK, you are really committed to using *olwm*, for it is the only Window Manager that implements all that OPEN LOOK requires of a window manager. Both Sun and AT&T provide their own versions of *olwm*; in this chapter we discuss the common functions that they provide for a variety of window management functions. In Chapter 2, *Working in the OPEN LOOK Environment*, we described the OPEN LOOK-specified procedures for selecting the input focus window, raising windows, moving and resizing them, and opening/closing them using the pointer. Here we will discuss a number of additional features of the OPEN LOOK Window Manager. You can use *olwm* to:

- Create additional terminal windows.
- Move windows to a new location on the screen.
- Change the size of windows.
- Lower windows (move them to the back).
- Refresh a window, or the entire screen.
- Close (iconify) or remove (delete) windows.

The OPEN LOOK Window Manager lets you manage your windows in several ways:

- You can use the “window frame” (including the titlebar) and various features on it: the window mark (iconify or “minimize” button);
- You can use the *Window Menu* available from the window’s frame;
- You can use the *Workspace Menu* (which most X11 documentation calls the *Root Menu);*
- Or you can use keyboard keys as shortcuts for many of these operations.

In this chapter, we'll review some basics about focusing input to a window or icon, and consider some other window management functions that can be performed using the window's frame, the *Window Menu*, and the *Workspace Menu*.

We'll see that some window management functions can be done from the keyboard, using specially-assigned keys. These keyboard shortcuts are often called "accelerators" because they can speed up your work. Keyboard accelerators usually involve special keys that X11 calls *modifier keys*. So we'll have to spend a bit of time looking at what X considers "special."

The *olwm* program is moderately flexible – less so than some other window managers, but with the intent of keeping OPEN LOOK intact as a standard – so you can configure it in certain ways. You can change the focus policy from the default click-to-type to a more convenient point-to-type (keyboard focus was described in more detail in Chapter 1). You can also change the *Workspace Menu*; in Sun's version of *olwm* you can replace the entire menu, organizing it as you wish, while in AT&T's version you can only add entries to the *Programs* list in the menu. Chapter 13, *Customizing olwm*, is devoted to this topic.

An even more flexible OPEN LOOK Window Manager is *olvwm*, which is upwards compatible with the OpenWindows version of *olwm* (from which it is derived). It is not a supported product, but the source for it can be obtained by anonymous *ftp* from Uunet and other sites. It has a "virtual desktop" facility that gives the appearance of multiple screens on a single display—as many screens as you want. See Section 6.5, "The Virtual Desktop (Virtual Edges)," for details on *olvwm*.

This chapter is aimed at the user of the OPEN LOOK Window Manager program from either Sun's OpenWindows 3.0 or AT&T's OPEN LOOK GUI 4.0. Use of other versions of the OPEN LOOK Window Manager, or of locally-customized versions, will be similar, but you should expect a few small differences as you go through the chapter.

6.1 Using Special Keys

Undoubtedly you know how to use a keyboard. However, X interprets certain keys somewhat differently than the labels on the keys would indicate.

Most workstation keyboards have a number of "modifier" keys, so-called because they modify the action of other keys. Generally these keys are used to invoke commands of some sort, such as window manager functions.

Three of these modifier keys should be familiar to any user of a standard ASCII terminal or a personal computer—Shift, Caps Lock, and Control. However, many workstations have additional modifier keys. A PC has an "Alt" key, a Macintosh™ has a "clover" ("command") key, a Sony workstation has keys named "Nfer" and "Xfer," and most Sun Workstations have three additional modifier keys, labeled "Alt" or "Alternate", and either a diamond character (◊) or "Left" and "Right.

Because X clients are designed to run on many different workstations, with different keyboards, it is difficult to assign functions to special keys on the keyboard. A software developer can't be sure that any given key will exist on every user's keyboard!

For this reason, most X clients make use of "logical" modifier key names, which can be mapped by the user to any actual key on the keyboard.

Up to eight separate modifier keys can be defined. The most commonly used (after Shift, Caps Lock, and Control) has the logical key name "Meta."

We'll talk at length about this subject in Chapter 14, *Customization Clients*, but we wanted to warn you here. When we talk later in this chapter about pressing the "Meta" key, you should be aware that there is not likely to be a physical key on the keyboard with that name. For example, on Sun workstations with the normal Type 4 keyboard, the key is labelled with a diamond character, on another workstation, the Meta key might be labeled "Alt" and, on another, "Left." And as we'll show in Chapter 14, you can choose any key you want to act as the Meta key.

Unfortunately, if you or your system administrator has been busy modifying key assignments, there is no easy way to find out which key on your keyboard has been assigned to be the Meta key. When you need to know, please turn to the discussion of key mapping in Chapter 14, *Customization Clients*, to learn more.

6.2 Input Focus and the Window Manager

Input Focus is, as described in Chapter 1, *An Introduction to OPEN LOOK and the X Window System*, the direction of keyboard input into a given window by use of the pointing device (mouse). The management of the input focus is controlled by the Window Manager program in any version of The X Window System. The OPEN LOOK Window Manager provides both common focus policies, pointer-focus and click-to-focus, so you can use the system as it suits you. In click-to-focus, you move the pointer into the titlebar of the window you want to focus input on, and click the SELECT (left) button. In pointer-driven or click-to-focus mode, the focus is directed to a window just by moving the pointer into that window. Both focus styles have their uses.

6.2.1 Focusing Input on an Icon

When a window is iconified, or closed to a small icon, it normally does not receive input. But you can still direct the focus to it. For example, if you iconify a terminal emulator that is a shell window, and then move the focus into the icon and type commands, nothing will happen; the keystrokes will be lost. However, there is one exception: the OPEN LOOK Window Manager allows certain keyboard shortcuts - such as opening the window - to be typed to an iconified window. And, in *olwm*, you can move the focus into an icon and press the MENU (right) pointer button to get the *Window Menu,* which operates as described below.

6.2.2 Transferring the Focus with Keystrokes

In AT&T's implementation of the OPEN LOOK Window Manager, you can move from one window to the next by the key sequence ALT+F6. You can move to the previous window with Shift+ALT+F6. In the OpenWindows version of the OPEN LOOK Window Manager, use Next Application (Alt-n) or Previous Application (Alt-Shift-n). This only works if you have the olwm.KeyboardCommands resource set to Full; see Appendix K, *OPEN LOOK Mouseless Operations*

6.2.3 What to do if olwm Dies and the Focus is Lost

It may occasionally happen that the window manager will "drop out from under you." While it is an important and reasonably well-debugged program, it is possible for *olwm*, like any computer program, to fail under certain rare circumstances. In this case, you will see a brief flurry of action as all the titlebars disappear and any iconified windows pop open.

Fortunately, in most cases the input focus reverts to X11's native "point to type" mode, so you can just move the cursor into any terminal emulator window and type the command *olwm&* to get the OPEN LOOK Window Manager back. For those cases when the input focus is left in click-to-type mode and *not* in a terminal window, you can use the *Secure Keyboard* feature of *xterm* to get the focus back, as described in Appendix A, *The xterm/olterm Terminal Emulator*. Of course you have to have an *xterm* running to use this mode. Alternately, you can log in at another terminal, or remotely, and restart *olwm* manually. Or, if you have a File Manager running, pull down *File -> Custom Commands -> UNIX Shell...* which will try to start a shell window for you; from there, start *olwm* as before†.

6.3 The Workspace Menu (or Root Menu)

The overall screen area of your display is called the *Workspace* in OPEN LOOK terminology, or the *root window* in X11 terminology. Most window systems give you a special "master menu" when on this area, and OPEN LOOK is no exception. Moving the pointer into the background window and pressing the MENU pointer button causes the *Workspace Menu* (also called the *Root Menu*) to appear. The normal content of this menu is shown in Figure 6-1.

Figure 6-1. OpenWindows Workspace Menu

Both *Programs* and *Utilities* have a menu mark beside them, so selecting either one will bring up a menu. *Properties* has a window mark ("...") after it, to indicate that selecting it will bring up a pop-up window, the properties manager. The Properties Manager (or Workspace Manager) is described in Section 14.1, "Properties Resource Editor" on Page 331. Both *Help...* and *Desktop Intro...* give access to tutorial documents using the *helpviewer* application described in Chapter 7, *The OpenWindows DeskSet Clients*. The former is an

† The final act of desperation before rebooting a machine is, if you have a shell window but can't get the input focus into it using any of these techniques, use copy-and-paste to copy "o", "l', "w", "m", and end-of-line into a shell window!

overall table-of-contents handbook, as shown in Figure 2-6, and the latter is the Introductory tutorial that appears in an uncustomized OpenWindows session. And selecting *Exit...* will bring up a confirming dialog; if you click on *Exit* in the dialog, you will terminate your X11/OPEN LOOK session.

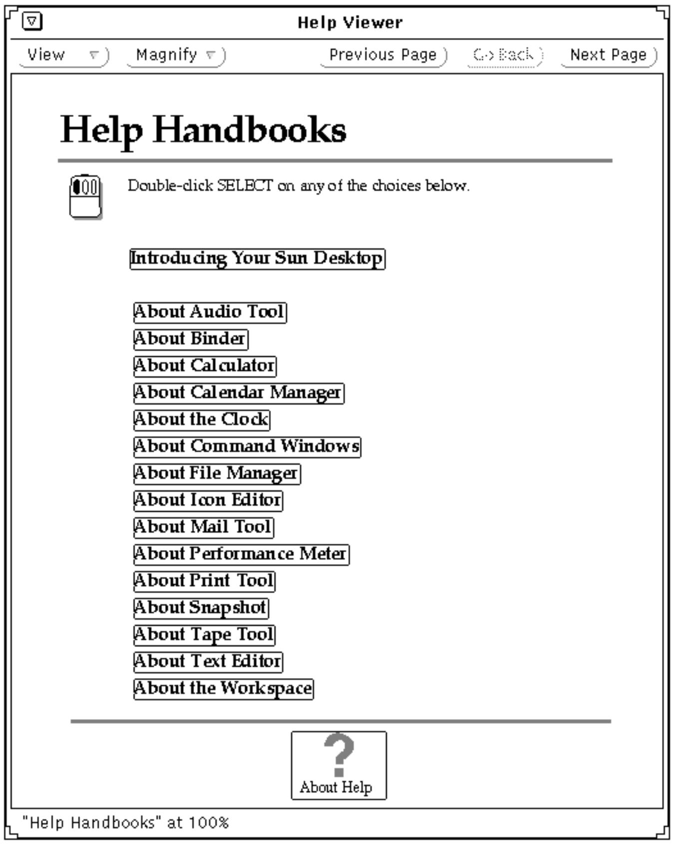

Figure 6-2. Top Level Help Handbook

Figure 6-3 is the *Programs* menu:

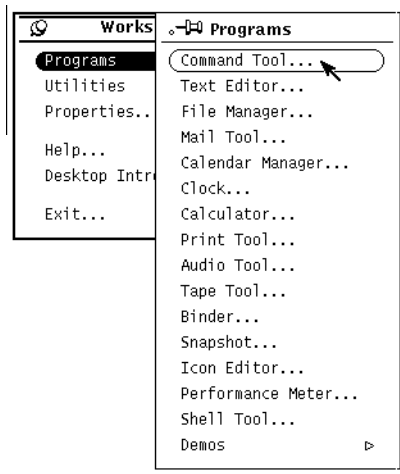

Figure 6-3. Workspace Menu, Programs submenu, OpenWindows Version

Selecting any one of these will start up the appropriate application. For example, selecting *Command Tool* will start up a *cmdtool* terminal emulator, described in Chapter 5, *The cmdtool/shelltool Terminal Emulator*. The individual programs are discussed throughout the remaining chapters of this book. Most of them are described in Chapter 7, *The OpenWindows DeskSet Clients*. The *Utilities* menu is presented in Figure 6-4.

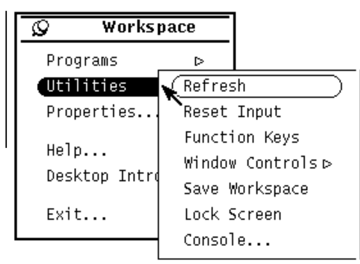

Figure 6-4. Workspace Menu, Utilities submenu, OpenWindows Version

The *Refresh* item will cause each window, including the root or Workspace window, to be re-painted or re-displayed. This is normally used when system output has inadvertently appeared over top of windows, for example, when console output is directed to the screen instead of to a terminal emulator such as *xterm -C* or *contool*...

The *Reset Input* item (which only appears on old versions of OpenWindows under SPARC, not on Intel x86 systems) has to do with the NeWS system.

The *Function Keys* brings up the Function Key map, as shown in Figure 6-5. See the *vkbd* man page for details..

Window Controls is described below.

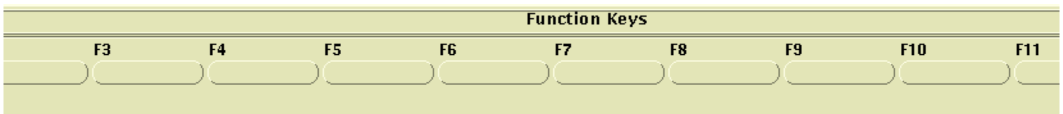

Figure 6-5. OpenWindows Function Key Popup

The Sun version of *Save Workspace* will save the names and locations of your applications in a file *($HOME/.openwin-init*) so that anytime you later start OpenWindows you will have your own customized set of clients. However, it may miss certain MIT X applications or those developed with non-OPEN LOOK toolkits. It may be preferable to customize your*.xinitrc* file instead, which is portable to more versions of X. See Section 3.4, "Customizing the X Environment: Specifying Resources" on Page 69.

Lock Screen starts a screenlock program; it is used when you will be away from your workstation or X terminal for a short period; you don't want to log out of X, but you don't want other people to use your account while you're away. While the screen is locked, nobody can type into your windows. To resume normal activities, you must hit RETURN and then correctly type your login password.

The last item, *Console*, starts up a terminal emulator *(Cmdtool*) with console output directed to it. This is useful to prevent further need for the *Refresh* item described above.

✗ A word of warning: the *Console* entry (as well as *contool* or *xterm -C*) should only be used on (run from) a workstation, not on a file server. Older versions of UNIX would allow any user to take over the system console output function, and your console window would then get all the operator-specific messages such as full filesystems, login problems, etc.

The Window Controls menu is shown in Figure 6-6.

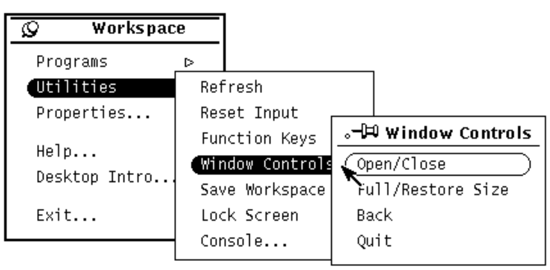

Figure 6-6. Workspace Menu, Window Controls Submenu

The *Window Controls* allow you to operate on one window or on a group of windows that you have selected. You can select a group of windows in two ways:

- Sweep out a rectangle on the Workspace that totally encloses the windows or icons you want

- Click SELECT on the titlebar, frame, or icon of the first, and ADJUST on the titlebars, frames or icons of the rest

Once you've done this, you can *Open/Close* them as a group, which will open any that are closed, and close any that are open. You can use Full/Restore Size, which will make any that are normal size occupy the full height of the display, and any that are full height return to their normal height and position. You can move the whole group to the *Back*. And if you want to clean up your screen quickly, you can *Quit* a whole group of windows. Note that the group association is temporary; it is broken as soon as you use SELECT for any other purpose on the same display.

This is the default set of menu items. On Sun's OpenWindows, you can change the Workspace menu or even completely replace it; on AT&T's OPEN LOOK you can add items to the *Programs* submenu. This is described in detail in Chapter 13, *Customizing olwm*.

6.4 The Window Menu: Moving, reshaping, and iconifying Windows

As with most window systems, you can use the titlebar to perform common window operations, such as moving, resizing, and closing the window. Unlike some other X11 window managers, however, OPEN LOOK lets you use the entire frame around the window to do these operations, so you don't have to have the titlebar visible. In the two-dimensional monochrome OPEN LOOK, the frame is very thin on the sides and bottom, but you can find it with a bit of practice. If you watch the screen cursor carefully, it "blinks" briefly as you move the pointer onto the frame around a window. You can set it to a larger thickness using X Resources, shown in the section *"Customizing the X Environment: Specifying Resources."*

In color OPEN LOOK, the frame is normally painted in a color that contrasts with the background and with other windows, so it's easier to see.

What can you do in the titlebar or frame? Use the MENU button to get the *Window menu*. The normal OPEN LOOK Window Manager *Window menu* has the items shown below:

Table 6-1

Choice	Function	Shortcut
Close	Selecting this choice will cause the window to be 'closed' or 'iconified', that is, replaced by a tiny icon window that represents the window.	Click SELECT on triangle at left of titlebar.
	(Clicking MENU on the icon will get a Window menu with the word 'Close' replaced by 'Open', which lets you open the window).	Double-click SELECT on icon

Table 6-1

Choice	Function	Shortcut
Full Size Normal Size	This will make the window assume its maximum size, which usually means the full height of the screen. If the window has already been set to “Full Size”, this selection will read “Normal size,” and selecting it will return the window to its original size.	Double-click SELECT on titlebar
Move	Sets up for mouseless (or mouseful) *move* operation.	Click and drag SELECT on titlebar
Resize	Sets up for mouseless (or mouseful) *resize* operation.	Click and drag SELECT on any resize corner.
Properties	This is not used at present. It will someday allow you to access the application’s Property sheet, if it has one, from the Window menu.	L3 key
Back	Move this window to the back (bottom) of the stack of windows.	L5 key
Refresh	Re-display the window, in case some bits of it got clobbered by accident.	
Quit	Causes the window and the program or application controlling it to be terminated.	Shell window: ^D or `exit`.

✗ If you’ve previously used Motif or Microsoft Windows, note that “Close” here means *iconify*, and does not terminate the program, while “Quit” means terminate. Motif and MS-Windows use the term “minimize” for *iconify*, and use “Close” to mean terminate.

There are several common operations that do **not** appear in the *Window menu*. For example, there is no *Front* item, because you can move any window to the top (front) of the window stack by clicking SELECT in its titlebar or frame. Since an icon is a (small) window, you can move an iconified window to the top just by clicking SELECT in it once (remember that clicking SELECT twice will cause the window to be opened). As well, the Sun keyboards have a *Front* (L5) key which has the same effect.

While there is a mouseless *Move* item on the *Window* menu, most users prefer to move windows around by pressing *and holding* the SELECT button on the titlebar or frame. When you do this, an outline of the window will follow the pointer to show the new position of the window. When you release the SELECT button, the window is actually moved. Similarly, you move an icon by clicking and holding SELECT on the icon. To use the mouseless *move*, click SELECT on the *Move* item, then use the arrow keys on the keyboard to move the window around. To move more quickly, hold down the SHIFT key while pressing the arrow key.

Similarly, there is a mouseless *Resize* item on the *Window menu*, but most users seem to prefer using the pointer. To change the size or shape of a window, move the pointer so that it is over one of the four "resize" corners. Notice that the cursor changes from an arrow to a circle, meaning that you can move in any direction at all. Again, a "rubber band" or outline will show the new size as you move, and when you release, the window will actually change size. To use the mouseless mode, click SELECT on the *Resize* item, then use the arrow keys to adjust the size of the window. Holding the SHIFT key will cause faster resizing. Most programs nowadays will react correctly to being resized. What about those that don't? For some versions of the *vi* editor, you should not resize the window while *vi* is running; if you do this, you must quit *vi* and restart it. For older systems, there is a program called *resize* that sets up the information on the resized window. It is described in the reference page in Part Three of this guide.

As mentioned under the description of "Window Controls" above, you can perform many of these operations on groups of windows. Say you wanted to open several iconified windows at the same moment. Click SELECT on the first one, then click on the other(s). Note that each one has its border highlighted to show that it is part of the selection. When you've SELECTed the last one, press MENU, and release on OPEN. With a mighty flash, all the windows you've selected will open at once. Another method of grouping is to click and hold SELECT on the background (root window), and drag SELECT until the rubber band box completely encloses all the windows you want to group. Now moving any one of them will cause them all to move. In either case, the association between these windows lasts only until you use the SELECT button to select any other items on the screen.

OpenWindows offers the following shortcuts, using extra keys that appear on the Sun Type-3 and Type-4 keyboards:

Alt-T Pressing the T key while holding the Alt key toggles the input focus between the last two windows that were selected.

L5 Pressing the L5 key when the pointer is in a window moves that window to the top (front). If the window is already at the top, L5 moves it to the bottom (back).

L7 Pressing the L7 key when the pointer is in a window iconifies the window; pressing it when the pointer is in an icon restores (opens) an icon to its window.

6.5 The Virtual Desktop (Virtual Edges)

The OPEN LOOK specification allows a window manager to provide "virtual edges" to the workspace (or root window). This allows the workspace to appear to be larger than the physical screen. However, neither supported version of *olwm* provides this at present. It is left to a contributed program called *olvwm*, derived from the Sun *olwm* source code, to provide a facility that, while not conforming to the OPEN LOOK specification for "Virtual Edges", provides a more useful solution to the problem of limited screen "real estate". *olvwm* is especially useful on small, low-resolution screens (800x600, 1024x768, ...) in which you can't usefully show much more than one full-sized terminal emulator window at a time. But it is also worthwhile on normal- and high-resolution screens (1152x900 and up). In fact, if your office desk looks anything like mine on a typical day, *olvwm* can make your workstation screen *really* look like your desk, bringing the paperless office one step closer to reality.

The following description of using *olvwm*'s virtual desktop facility is adapted from the manual page by Scott Oaks, who added the virtual desktop facility to *olwm* to produce *olvwm*. The full reference page is reprinted in Section Three of this guide. You can obtain *olvwm* by anonymous *ftp*, as described in the *Preface*.

Olvwm starts by displaying a virtual desktop manager (VDM) window along with any other tools normally started in your *.xinitrc* file. The virtual desktop manager is a reduced display of all windows active on the desktop at any time. The area of the desktop which is visible (that is, which appears on your screen) is outlined in the virtual desktop manager. For example, Figure 6-7 shows a normal VDM that has room for six screen-sized segments. The one in the upper left has a large terminal emulator labelled "darian". The one below it has a single window called "Snapshot" (see Chapter 9, *Graphics Clients*). The top-middle screen has two *Mailtool* windows (see Chapter 7, *The OpenWindows DeskSet Clients*). The window under that has one large *Xfig* window (a contributed graphics program).

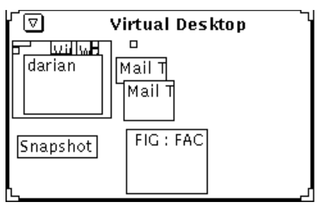

Figure 6-7. olvwm's virtual desktop manager

But what are those tiny boxes at the top? They are icons, clocks, and other gadgets. Figure 6-8, continuing the example, shows the screen in the upper left corner:

6.5.1 Moving windows with OLVWM

You move windows on the screen just as you normally would in *olwm*. You can also drag windows from the screen into the virtual desktop manager or from the virtual desktop manager to the screen, or just within the virtual desktop manager. Note that if you select on a window's frame which is overlapping the virtual desktop manager, you will end up dragging the window into the virtual desktop manager, and it will suddenly shrink to the scale of windows in the VDM. You can disable this with an X resource (`AllowMoveIntoDesktop`, see Chapter 12, *Setting Resources*) if you are bothered by this behavior.

If you select a window inside the virtual desktop manager (but not necessarily in the current segment), you can move the window. This is just like moving a window around on the workspace, except that it all happens in much-reduced scale.

If, instead, you double click on any area inside the virtual desktop, then that segment becomes the current segment.

Figure 6-8. The top left screen of OLVWM's virtual desktop

6.5.2 OLVWM Sticky Windows

The Virtual Desktop Manager window never moves on your screen when you change views. That's because the desktop manager window is permanently "sticky." "Sticky" windows never move position on the screen when you change your view into the desktop. To set a particular window as sticky, simply select *Stick* in its frame menu. You may similarly *Unstick* a sticky window via its menu.

6.5.3 Advantages of OLVWM

The most obvious advantage is that you have more space to work on; windows that you aren't using can be moved off the screen without iconifying them.

Another advantage is that it allows the Workspace Menu to be automatically pinned up; this feature is not in the standard *olwm* yet.

6.5.4 Limitations of OLVWM

There are two problems with *olvwm* that affect NeWS programs (such as FrameMaker 3.1 OPEN LOOK) and SunView programs running under OpenWindows.:

- ✗ NeWS programs built with The NeWS Toolkit use the NeWS Window Management functions, which look almost exactly like *olwm*; under *olvwm*. SunView programs use the SunView mouse conventions, described in Appendix L, *Running SunView Applications on OpenWindows*. These programs always appear on the current segment of the screen, and cannot be moved to other segments; their *Window Menu* does not have the *Stick* or *Unstick* options. That is, they are in effect sticky; they always appear even if you try to move away from them.
- ✗ Some NeWS programs *sometimes* behave as though click-to-type focus policy were in effect, even though pointer-focus policy is in effect.

Users of such programs should consider using *olwm* instead.

6.5.5 The Future

It remains to be seen whether *olvwm* will be "adopted" by Sun, or will remain in the contributed software category. At any rate, the facility if offers is sufficiently useful that Sun ought to provide it somehow.

6.6 The Workspace Manager (Properties Manager)

The OPEN LOOK specification calls for a Workspace Properties window to allow the user to change "global preferences". This is implemented by a separate program from the Window Manager, called the Properties Manager. It is mentioned here because it often seems to be part of the OPEN LOOK Window Manager. Indeed, it cooperates with *olwm* in that some of the property changes it makes are acted upon immediately by the Window Manager. However, it is a separate client program started by the Window Manager. Accordingly, the Workspace Properties Window is described more fully in Chapter 14, *Customization Clients*.

Note that this is not the same as the Properties item on the *Window* menu of individual windows; that item is disabled if it is not yet implemented, as in OpenWindows.

We have now discussed the most important features of the OPEN LOOK specification and the OPEN LOOK Window Manager. We now turn our attention to a detailed discussion of the important clients, starting with the terminal emulators. The *cmdtool/shelltool* terminal emulator clients are discussed in Chapter 5, *The cmdtool/shelltool Terminal Emulator*, while the pre-OPEN LOOK terminal program *xterm* is discussed in Appendix A, *The xterm/olterm Terminal Emulator*. Subsequent chapters discuss other important clients.