CHAPTER 13

Customizing the OPEN LOOK Window Manager

Some window managers have the flexibility to let you paint your root window sky blue by pressing the middle pointer button in a titlebar while holding down the shift, control, and ALT keys. As we've seen in Chapter 2, *Working in the OPEN LOOK Environment*, the OPEN LOOK User Interface specifies the window system's behavior in such detail that the user can't be given this degree of flexibility.Hence, The OPEN LOOK Window Manager does not have the blazing configurability of some other X11 Window Managers. It is designed to allow users flexibility, but still keep you within the confines of the OPEN LOOK Graphical User Interface specification.

If you had that kind of flexibility, users could (and some would) configure their systems in ways that would violate the OPEN LOOK specification. So OPEN LOOK Window Manager is considerably easier to configure—or less flexible, depending on your point of view—than some other X Window System window managers, such as the widely-used *twm* and *mwm.* Customization of *olwm* is, in fact, limited to changing or replacing the *root menu* or *Workspace Menu*, specifying options on the command line, and specifying a number of X Resources that control *olwm*'s behavior. These topics will occupy our attention in the remainder of this chapter.

13.1 The Workspace Menu

The OPEN LOOK specification describes two levels of conformance, called *Level One* and *Level Two*. Level One describes a basic, minimal implementation, while Level Two has more features than Level One. The AT&T-OL version of *olwm* conforms to Level One, which only requires that you be able to change the list of programs in the *Programs* part of the *Workspace Menu*. The OpenWindows version of *olwm* conforms to Level Two, which specifies that the user can replace *all* the entries in the *Workspace Menu*. The following section describes both versions.

13

13.1.1 Level 1 Customization (AT&T-OL)

The list of Programs in the *Programs* sub-menu of the AT&T-OL *Workspace* menu is maintained from the Properties program. Selecting the Category *Programs Submenu* produces a window that lets you add additional programs. There is a Scrolling List listing the optional programs, a *Name* and *Invocation String* and *Mnemonic* text field that lets you put in the values for each program. Below this are several buttons.

To add a new program, for example, select Insert. Type the name (such as “Local Xterm”) in the *Name* text field. Type the invocation string (such as “exec xterm”) in the *Invocation String* text field. Pick a single-character name that isn't already a mnemonic for one of the programs, and type it in the *Mnemonic* field. Then click SELECT on *Apply Edits* and your changes will be saved. From now on, the *Programs* submenu of the *Workspace* menu will include this item.

You can use the scrolling list in the normal way to edit or delete items in the scrolling list. To change and item, for example, click SELECT on it in the scrolling list. Update one or more of the text fields (remember to click on each field to activate it), and click SELECT on the *Apply* button to save your changes. To delete an item, select it, click on the DELETE button, and click SELECT on the *Apply* button to make your changes take effect.

13.1.2 Level 2 Customization (OpenWindows)

To customize the *Workspace* menu in OpenWindows, you need only edit a text file. The default file is */usr/openwin/lib/openwin-menu*, which looks like this:

```
# # @(#)openwin-menu23.15 91/09/14 openwin-menu # #OpenWindows
default root menu file - top level menu #
"Workspace" TITLE
"Programs" MENU$OPENWINHOME/lib/openwin-menu-programs
"Utilities" MENU$OPENWINHOME/lib/openwin-menu-utilities
"Properties..."PROPERTIES
SEPARATOR
"Help..."exec $OPENWINHOME/bin/helpopen handbooks/top.toc.handbook
"Desktop Intro..."exec $OPENWINHOME/bin/helpopen
handbooks/desktop.intro.handbook
SEPARATOR
"Exit..."EXIT
```

This is what *olwm* actually uses to create the normal workspace menu, consisting of:

Table 13-1. Olwm normal root window

Programs
Utilities
Properties... -
-
Help Desktop Intro...

Table 13-1. Olwm normal root window

-
Exit...

Each line in the menu text file consists of three fields:† the name to appear on the menu, optional keywords, and the action to take when the line is selected. Comment lines beginning with '#' in column one, and null lines, are ignored (you can usefully put a '#' before a line that is temporarily not working, to make it invisible). Look at the example a minute and the overall pattern should become clear.

The easiest and safest way to change this file is to copy it into your home directory under the name *.openwin-menu*, for example, */home/darian/ian/.openwin-menu*. Then make any changes you want. But make changes one at a time, and keep track of your changes (hint: use RCS or SCCS). Why? Because if you make a mistake, and your menu is incorrect, *olwm* may just ignore the menu file completely, and use the system one (which is why you should not tamper with the system copy). If your menu file is huge, it can be hard to spot errors. The current *olwm* is pretty good at reporting many errors:

```
olvwm: menu label mismatch in file /home/xyz/ian/.openwin-menu, line 104
```

But it's still easier to make changes in small doses so you can easily back up and see what you did wrong.

As an example of a simple change, if you'd prefer to start *xterm* instead of *cmdtool* from the *Programs* menu, change the line

```
"Command Tool..." DEFAULTexec $OPENWINHOME/bin/xview/cmdtool
```

to these two lines:

```
"Xterm..." DEFAULTexec $OPENWINHOME/demo/xterm
"Command Tool..." exec $OPENWINHOME/bin/xview/cmdtool
```

The DEFAULT keyword is optional, but useful: it specifies what menu item will be selected when you click or release SELECT on the parent menu, in this case, on the word *Programs* in the top-level *Workspace* menu. The word "exec" is present as an optimization. A copy of the UNIX shell is used to interpret the string that is given as the command, and the *exec* prevents this extra shell from sticking around waiting for the application *xterm*, for example) to terminate.

As another example, If you get tired of the "Please confirm exit from window system" prompt, you can suppress it by changing

```
"Exit" EXIT
```

to

```
"Exit!" EXIT_NO_CONFIRM
```

† The syntax is the same as the older SunView root menu, except that you cannot (yet?) specify an icon in place of the name field.

You probably should have both; the latter will exit without confirming. Use with care!

There are several keywords that can be specified, listed in Table 13-2. They must be in upper case.

Table 13-2. Sun OPEN LOOK Window Manager menu file keywords

Keyword	Function
TITLE	Specify title for menu (or submenu).
MENU, END	Make a menu, or invoke one as in examples above
DEFAULT	Specify default case for menu
PIN	(after END) makes menu pinnable.
REFRESH	Refresh all windows
POSTSCRIPT	Send rest of line to NeWS (OpenWindows only)
SAVE_WORKSPACE	Save windows in $HOME/.openwin-init
PROPERTIES	Start up Workspace or Properties manager
EXIT	Exit window system
EXIT_NO_CONFIRM	Exits without confirming notice.
WM_EXIT	Exit Window Manager, but don't kill clients.
RESTART	Re-run olwm (e.g., after changing the menu file)
FLIPDRAG	Toggle dragging of window contents when moving windows.
FLIPFOCUS	Flip focus policy (click vs. follow-mouse)
NOP	Does nothing (placeholder).
SEPARATOR	Produce a blank line.
PRINT_SCREEN	Not implemented yet.
CLIPBOARD	Not implemented yet.
WINDOW_CONTROLS	Not implemented yet.
REREAD_MENU_FILE	Just what it says

Table 13-3

You may not like the default root window menu, so you can completely change it. My own root menu looks like this: box; l. Local Windows Window Programs OpenWin Demos

Refresh Screen Lock Screen Terminations The first three pull down menus. The Local Windows lets me at several different terminal emulators. Another menu, normally commented out but made available when I'm on a LAN or the Internet, has remote logins for the hosts I normally use,

```
"ozzify"xterm -T ozzify -geom +200+140 -e rlogin ozzify
```

for each one. The *Window Programs* menu has a **long** list of X11 (and NeWS, and SunView) applications. Some are straight invocations, like

```
"Author/Editor"exec $SQBIN/ae
```

Others are more involved, like running programs remotely with a display back to my workstation. These three entries let me use *xman* (see Chapter 8, *Other Standard Clients*) on any of four different versions of SunOS:

```
"Man pages (SunOS3.5)" rsh sq /usr/bin/X11/xman -display $DISPLAY
"Man pages (SunOS4.0)" rsh sqarc /usr/bin/X11/xman -display $DISPLAY
"Man pages (SunOS4.1)" xman -display $DISPLAY
"Man pages (SunOS5.0)" rsh sqlaris /usr/bin/X11/xman -display $DISPLAY
```

There is no limit to what you can put in this file; review the discussion of DISPLAY in Chapter 1, *An Introduction to OPEN LOOK and the X Window System* and the examples here. As a final example, to try out some of the built-in functions described earlier, and to confirm that certain functions were still unimplemented, we just added this to our own menu file:

```
TestMENU "Weird stuff"TITLE
    FlipFocusFLIPFOCUS
    FlipDragFLIPDRAG
    winctl  WINDOW_CONTROLS
    prtscr  PRINT_SCREEN
TestEND PIN
```

Remember that you have to re-start the window manager for changes to take effect; that's why our have an entry that calls RESTART on the top-level *Workspace* menu. A hint: if you make any mistakes, your entire structure of menu files will be ignored. To avoid having to restart olwm when modifying your own files, you may wish to modify the system Utilities menu *$OPENWINHOME/lib/openwin-menu-utilities* to include the following lines:

```
"Re-read menu file"REREAD_MENU_FILE
"Restart Window Manager"RESTART
```

If you are having trouble with your *.openwin-menu* file and want to see the error messages from programs run from it, you can exit your window manager and restart olwm from a visible cmdtool, as follows:

Either choose "Exit Window Manager" if you've installed WMEXIT in your Terminations or Utilities menu, or, give the command *ps -ax* (*ps -ae* on Solaris 2 or System V), and use the UNIX kill command to kill the olwm process, for example,

```
% ps -ax
 PID TT STAT TIME COMMAND
 0 ? D 0:00 swapper
 1 ? IW 0:00 /sbin/init -
 2 ? D 0:00 pagedaemon
```

```
 ...
 172 co IW 0:00 olwm
 174 co IW 0:00 olwmslave
 176 co I 0:00 clock -Wn -Wp 930 90 -digital
 178 p1 S 0:09 cmdtool -Ww 80 -Wh 24 -Wl darian -scale extra_l
 298 p1 R 0:00 ps -ax
% kill 172
%
```

Then give the command

```
% olwm -display :0 &
```

from within the terminal emulator. This will start a new copy of olwm, with its standard output and standard error displayed in the terminal window. Then when olwm has trouble reading or parsing your menu file, or a program you start doesn't work, you can see why.

13.2 OPEN LOOK Window Manager Command Line Options

13.2.1 OPEN LOOK Window Manager Options - AT&T-OL Version

The AT&T-OL version of *olwm* has very simple command-line customization. In fact, it accepts only one optional command-line argument: the name of the display to use. It defaults to the display named in the DISPLAY environment variable setting; if this is not set, it defaults to `unix:0`, that is, the console of the 386 PC (or other workstation) on which it is invoked.

13.2.2 OPEN LOOK Window Manager Options - OpenWindows Version

The following is a list of command-line options for the OpenWindows version of *olwm.* Most have counterparts in the X11 resource database (see Chapter 12, *Setting Resources*, and the section *Configuring OLWM* with resources below). A command-line option will override the corresponding setting from the resource database.

`-2d` Use two-dimensional look. This is the default, and the only possible look, for monochrome displays.

`-3d` Use three-dimensional look. This is the default for a color screen, and is ignored on a monochrome screen

You can usually ignore the two options above, since the window manager usually does the right thing.

`-c, -click`. Use click-to-focus mode. This is the default focus mode. The opposite of `-follow.`

`-f, -follow` Use pointer-driven (focus-follows-mouse) focus policy. Default mode is click-to-focus.

You normally set this in an X11 resource file; you can specify the other policy on the command line to try it out, for example.

`-parent` Ignore windows that are already on the screen. In earlier versions of *olwm* this was used to make the window manager to leave titlebars off certain "desk accessory" windows. However, the preferable way to achieve this is to name them in the "MinimalDecor" resource, since windows ignored by `-parent` can never be moved or resized, and will never receive the input focus.

13.2.2.1 Debugging Options

There are several *Debugging Options* described in the *olwm* Reference Manual Page in Section Three of this guide that are only meant for people debugging X applications or *olwm* itself. Don't use them unless you know what you are doing.

13.2.2.2 Generic Options

The OpenWindows version recognizes the following generic options (see Chapter 11, *Command Line Options*):

Table 13-4. Olwm generic options

`-display` display	Specify which display to manage.
`-fn` fontname `-fontname` fontname	Set the font for window titles.
`-name` resource-name	Use name to find resources in X11 resource database.
`-xrm` resource-string	Put X11 resources on the command-line.

You would normally start *olwm* from your*.xinitrc* file, for example:

```
olwm -display unix:0 -fn zapfchancery-mediumitali c&
```

in your *.xinitrc* file.

13.3 Configuring OLWM with resources

As with most X11 programs, there is a significant amount of configurable behavior built into the window manager, and much of it can be configured with X Resources. Unfortunately, the two versions of *olwm* — AT&T-OL and Sun OpenWindows — have totally incompatible resource lists, so it is necessary to know which version you are using and to choose the appropriate resources.

13.3.1 Resources for Configuration — AT&T-OL

AT&T does not document the resources needed to configure their version of *olwm*, on the grounds that you are expected to use the Workspace Properties Manager to update them.

However, it has been determined that the following resources are at least known to their version of *olwm*.

Table 13-5. AT&T-OL Resources

Configuration Resources
PointerFocus
SaveSet
IconGridSize
IconGrid
WindowAttributes
WindowFrameColor
Warnings
HelpDirectory
WMIconGravity
HeaderFont

Table 13-6. AT&T-OL Resources that set Key Values

Keys for shortcuts
wmOpenCloseKey
wmSizeKey
wmPropertiesKey
wmRefreshKey
wmBackKey
wmQuitKey
wmDismissThisKey
wmDismissAllKey
wmMoveKey

Table 13-6. AT&T-OL Resources that set Key Values

Keys for shortcuts
wmResizeKey
wmOwnerKey

13.3.2 Resources for Configuration — OpenWindows

Here are the resources allowed by the Sun OpenWindows version of *olwm*. Most of these resources can be set by the Properties program (See Chapter 14, *Customization Clients*). However, they are listed here for reference, and for those who wish to edit the Resource files manually.

The "class" name for Sun's version of *olwm* is "OpenWindows", so the string "OpenWindows." must be used as part of the name in the resource file Thus, to enable automatic "raising" (move to front) of the focus window, you would put "OpenWindows.AutoRaise: true" in your X resource file.

Some resources specify key bindings. A key specification is a list of words separated by space, each of which is a KeySym name. Every word except the last must specify a modifier key. All KeySym names are case-sensitive. For example, to bind the Color-Lock key to control-shift-F1, you would use the following resource specification:

```
OpenWindows.ColorLockKey: Control Shift F1
```

You can specify that a key will have effect under all modifier combinations by using the special keyword "Any" instead of a real X11 modifier.

Boolean values can be specified in any of several ways: the pairs "true", "false", "on", "off", "yes", "no", "1", "0", "t", and "nil" will all work.

Table 13-7. OpenWindows OPEN LOOK Window Manager Resources

Resource	Type	Function
AutoRaise	boolean	Move focussed window to front
Beep	enumeration	Never, Notice or Always (==on warnings, etc.)
ButtonFont	font name	Font for buttons
ClickMoveThreshold	integer	
ColorFocusLocked	boolean	Color focus lock policy
ColorLockKey	key specification	Key to lock colormap
ColorUnlockKey	key specification	Key to unlock colormap

Table 13-7. OpenWindows OPEN LOOK Window Manager Resources

Resource	Type	Function
ConfirmKey	key specification	Key selects default from Notices
CursorFont	font name	Font for cursor (don't change)
DefaultTitle	string	Title if application has none
DragRightDistance	integer	For menus
DragWindow	boolean	Move whole window or just outline
EdgeMoveThreshold	integer	For mouse moves
FlashFrequency	integer	How often to flash after Owner?
FocusLenience	boolean	Don't enforce ICCCM on "input hint"
FrontKey	key specification	Key to front (or back) window
GlyphFont	font name	Font for drawing; don't change
IconFont	font name	Font for icon names
IconLocation	enumeration	Where to stack icons
MinimalDecor	list of strings	Desk Accessory windows; no titlebar
MoveThreshold	integer	For mouse moves
MultiClickTimeout	integer	For pointer clicks
OpenKey	key specification	Key to open (or close) a window
PPositionCompat	boolean	Compatibility for X11R3 clients
PopupJumpCursor	boolean	
RefreshRecursively	boolean	
RubberBandThickness	integer	
SaveWorkspaceTimeout	integer	
SetInput	enumeration	SnapToGrid
TextFont	font name	Font for normal text
TitleFont	font name	Font for titlebar
Use3D	boolean	Use 3D (or 2D) OPEN LOOK
Use3DFrames	boolean	Use 3D frame thickness

Table 13-7. OpenWindows OPEN LOOK Window Manager Resources

Resource	Type	Function
WindowColor	color specification	BG1 color
WorkspaceColor	color specification	Root window color, bitmap, or null

Each of these is discussed in more detail in the *olwm* reference manual page in Section Three of this book.