Using the Motif Window Manager

The Motif Window Manager (**mwm**) allows you to move windows, change their size, change them into icons, and create new windows. By editing configuration files, you can alter the appearance and behavior of **mwm** to suit your preferences. Refer to Chapter 5 for information about customizing **mwm**. This chapter describes the default behavior of **mwm** and provides information on the following topics:

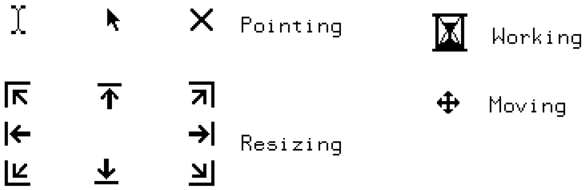
- Interacting with **mwm**
- Accessing window managing functions
- Selecting a window for input and raising a window
- Using the **mwm** window menu
- Moving and resizing windows
- Maximizing windows
- Using icons
- Closing windows
- Using the default root menu
- Using the **mwm** virtual desktop support

Interacting with mwm

You interact with **mwm** with a pointing device, usually a mouse, or by pressing various combinations of keys on the keyboard, or both. The Motif Window Manager (**mwm**) uses an object−action model of interaction. You select a window, menu, icon, or other graphic object and indicate an action for that object.

A graphic image on the display, called a *pointer*, tracks the movement of the mouse on your desk. When X starts up, the pointer has an × shape. (This shape is the default; the pointer shape can be customized.) When you move the mouse, the image of the pointer may change. The pointer shape is meant to provide a visual cue as to the activity that is being performed. For example, inside a terminal window or text editor, the pointer may look like an I−beam (see Figure 2−1).

Figure 1 Pointer Shapes Provide Visual Cues to Activity



Using the Mouse

You can perform the following actions with a mouse:

Pointing

Sliding the mouse on the desktop to position the pointer on an object on the screen

Pressing

Holding down a mouse button

Clicking

Pressing and releasing a mouse button without moving the pointer

Double−Clicking

Clicking a mouse button twice in rapid succession

Dragging

Pressing and holding down a button while moving the mouse

(See the Preface for a description of the conventions used in this guide when referring to mouse buttons.)

Using the Keyboard

You can perform many window management functions without using a mouse. Often, window management functions from the keyboard require using modifier keys. When you press these keys, they modify the action normally associated with other keys. Because Motif runs on many different kinds of terminals and workstations, the names of the keys vary from one system to the another. (See the Preface for the model keyboard conventions used in this guide.)

All **mwm** keyboard actions conform to the behavior specified in the *Motif Style Guide.* For example, pressing `Shift Escape` pops up the window menu for the active window. You browse through the menu items using the arrow keys. Pressing `Return` selects a menu item. Pressing `Cancel` cancels the current window management operation.

Accessing Window Managing Functions

You can access most window managing functions through decorations on the **mwm** window frame (see Figure 2−2).

Figure 1 The Client Window Frame

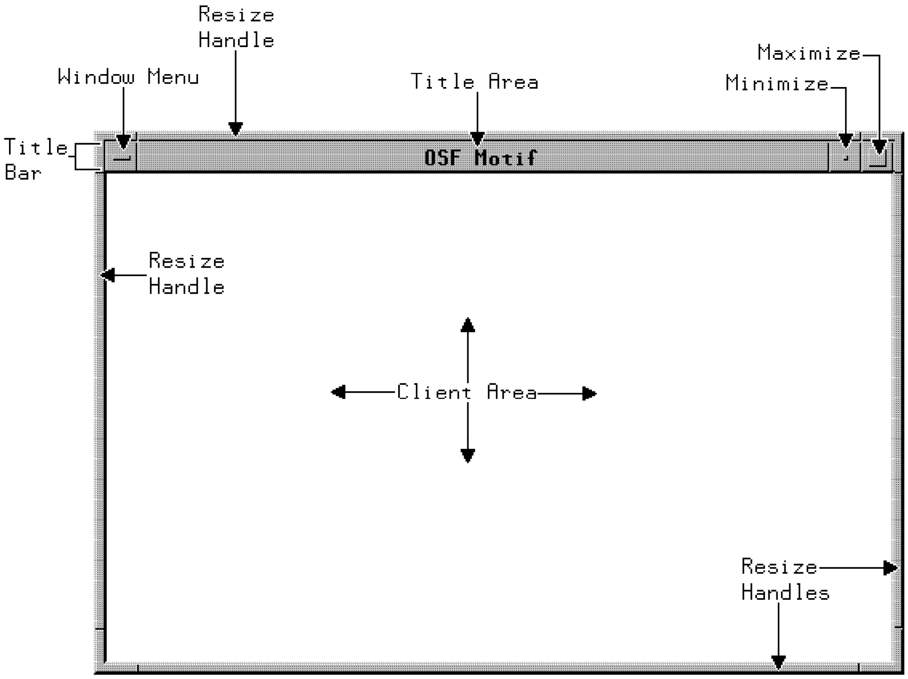


Table 2−1 lists the window frame decorations and their associated functions.

Table 1 Window Frame Decorations

This frame part...	***Is used to...***
Title area	Move a window
Window Menu button	Display the Window Menu
Minimize button	Iconify a window
Maximize button	Expand a window to maximum size
Resize frame handles	Stretch or shrink a window horizontally, vertically, or diagonally (in two directions)

Selecting a Window for Input

Before a window can receive input or other window management functions, you must select a window for input *focus*. To avoid ambiguity, only one window can be selected at a time. This window is said to be the *active window*. When a window is active, the color or shade of the window frame changes. You can choose the color or shading for active and inactive windows (see Chapter 5).

The input *focus policy* determines how a window is selected. The window manager supports both an *explicit focus* policy and an *implicit focus* policy. The implicit focus policy is also known as the *pointer focus* policy. The default input focus policy is explicit, which means that you must click Button 1 on the window frame or press `Alt Tab` to change the input focus from one window to another.

Pointer input focus policy causes **mwm** to change the input focus to match the pointer's position; no other actions are required. Thus, the window that contains the pointer always has the input focus. You can specify this input focus policy by modifying the defaults in the X configuration file. (See Chapter 5 for further information about setting input focus policy.)

The easiest way to select a window for input is to move the mouse so that the pointer rests within the frame of the window you want to use. If an explicit focus policy is in effect, you also need to click Button 1. With explicit focus policy, only the window selected for *keyboard focus* receives keyboard input, regardless of where the pointer is located. If you are using the explicit focus policy, you can also move the input focus using the keyboard. Pressing `Alt Tab` moves the input focus to the next window in the window stack; pressing `Shift Alt Tab` moves the input focus to the previous window.

Raising a Window

To raise a partially concealed window to the top of the window stack, or in other words to the front of the display, position the pointer on the window's frame and click Button 1. Be sure that the pointer is not resting on one of the window frame command buttons: minimize, maximize, or Window Menu (see Figure 2−2).

The window manager also provides default methods for raising a window from the keyboard, as shown in Table 2−2.

Table 1 Raising a Window from the Keyboard

Pressing these keys...	***Does this...***
<Alt> <Tab>	Raises the next window in the stack and moves input focus to that window (explicit focus only)
<Alt> <Shift> <Tab>	Raises the previous window in the stack and moves input focus to that window (explicit focus only)
<Alt> <Escape>	Raises the next window in the stack without changing the input focus
<Alt> <Shift> <Escape>	Raises the previous window in the stack without changing the input focus

When you are using an explicit input focus policy, raising a window to the top of the stack also sets the input focus to that window. If you are using a pointer focus policy, however, the window manager does not automatically set input focus to the raised window. In other words, a window can continue to receive input even though it is not at the top of the stack. You can, however, set a resource in the the X configuration file to move a window automatically to the top of the stack when you set input focus to that window (see Chapter 5).

Using the mwm Window Menu

The *Window Menu* provides an alternate way to access window manager functionality. There are two ways to use the Window Menu:

- Position the pointer on the *Window Menu button* in the top left corner of the window frame and press Button 1. If you want the menu to remain posted, click Button 1. You can also post the Window Menu by positioning the pointer anywhere on the window frame and pressing Button 3. Once the menu appears, you can browse through the menu items by dragging the mouse. As the pointer touches an available selection, the selection is highlighted. To choose a selection, release the mouse button when the item you want to select is highlighted.
- Set the keyboard input focus to a window and press `Shift Escape` or `Alt Space`. This displays, or *posts*, the Window Menu. Use the up and down arrow keys to browse through the menu items. Press `Return`, `Select`, or `Space` to select the item that is highlighted. Alternatively, you can remove the posted menu by pressing `Cancel`.

Figure 2−3 illustrates a posted Window Menu.

Figure 1 The Motif Default Window Menu

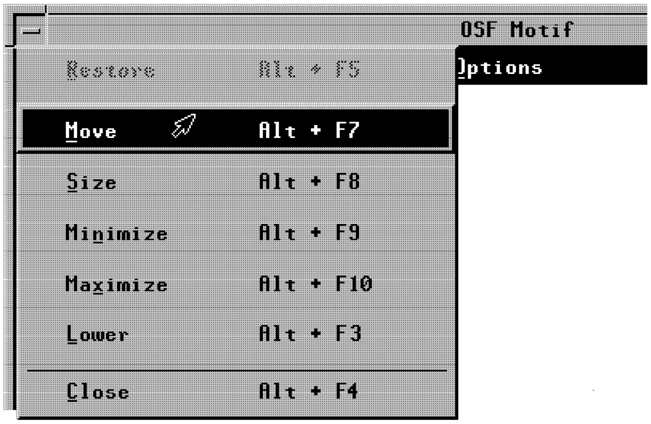


Table 2−3 lists the default Window Menu selections and their corresponding functions.

Table 1 The Default Window Menu Selections

This selection...	***Does this...***
Restore	Restores a window to its normal size after being iconified or maximized
Move	Initiates a move operation for a window
Size	Initiates a resize operation for a window
Minimize	Shrinks a window to an icon (a graphic representation)
Maximize	Enlarges a window to its maximum size
Lower	Pushes a window to the bottom of the window stack
Close	Terminates the client

Menu selections for functions that are not immediately applicable are grayed out; selecting these items has no effect.

You can also select Window Menu items from the keyboard. *Accelerators* are key bindings that provide quick access to menu functions without posting the menu. For example, `Alt F3` lowers the window. Accelerators appear in the menu to the right of their associated menu items.

Mnemonics select a menu item after the Window Menu is posted. Mnemonic keys are indicated in a menu item by an underlined character in the menu item label. For example, pressing `M` while the Window Menu is posted has the same effect as selecting *Move* with the mouse.

Moving Windows

To move a window, position the pointer in the window’s title area, press and hold down Button 1 and drag the mouse. The pointer usually changes shape for the duration of the operation. As you move the mouse, by default a wire frame outline shows the current location of the window. You can set a resource so that the move operation displays the window itself rather than an outline (see Chapter 5). A feedback window is also displayed, which shows the current location of the window in pixels relative to the origin of the root window.

When the wire frame is where you want the window to be located, release the button. The window moves to that location. A move in progress can be canceled by pressing `Cancel`.

You can also move a window from the keyboard. First, select *Move* from the Window Menu or press the accelerator `Alt F7`. The arrow keys move the window in the direction of the arrow. Holding down `Ctrl` as you press the arrow keys moves the window in larger increments (see Table 2−4). You can also move the window by moving the mouse after the move operation has been initiated from the keyboard. Press `Return` to complete the move operation.

Table 1 Moving a Window from the Keyboard

Pressing this key...	***Does this...***
<[uarr]>	Moves up one pixel
<[darr]>	Moves down one pixel
<[larr]>	Moves left one pixel
<[rarr]>	Moves right one pixel
<*Ctrl*> <*arrow key*>	Moves several pixels
<*Return*>	Completes the interactive placement

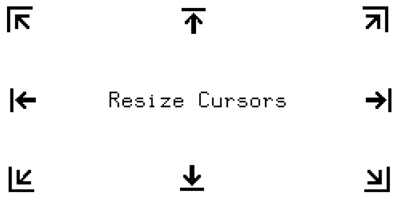
Resizing Windows

You can change the size of a window by positioning the pointer on one of the following window frame resize handles:

- Top edge
- Bottom edge
- Left side
- Right side
- Lower right corner
- Lower left corner
- Upper right corner
- Upper left corner

Each handle has a distinct cursor shape (see Figure 2−4).

Figure 1 Resize Handles Have Distinctive Cursor Shapes



The direction and outcome of the resize operation depends on which handle you grab. Press Button 1 while the pointer is on one of these handles to grab the window. Move the mouse to expand or contract the window. A wire frame outline shows the changing dimensions. Also, a small feedback window displays the current dimensions of the frame. Release Button 1 to complete the resize operation. You can cancel the operation at any time by pressing `Cancel`.

You can also resize a window from the keyboard. First, select *Size* on the Window Menu or press the accelerator `Alt F8`. To change the size of the window, press the arrow key that points in the direction of the dimension that you want to change. Then, you can increase or decrease the size of the window by pressing on the same or opposite arrow key. Press `Return` when the window is the size that you want.

Maximizing Windows

Maximizing a window makes the window its largest allowable size. A window's largest possible size can be limited by the application program or by configured resources.

To *maximize* a window, position the pointer on the *maximize button* on the far right of the window frame and click Button 1. The client window expands to the maximum size set by either the application program or the configured resources. By default, the maximized window fills the screen.

Once the window is maximized, the large square image on the maximize button appears pressed into the screen. Clicking Button 1 on the recessed maximize button restores the window to its previous size and position.

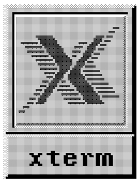
You can also maximize a window by selecting *Maximize* on the Window Menu or pressing the accelerator `Alt F10`. You can restore a maximized window by selecting *Restore* on the Window Menu or by pressing the accelerator `Alt F5`.

Using Icons

Icons are small graphic representations of windows that are used to reduce clutter on the screen. You can turn a client window into an icon or change an icon back into a window at any time.

A typical icon has two parts, an image and a label, as shown in Figure 2−5.

Figure 1 A Typical Icon



By default, the window manager arranges icons along the bottom of the display, from left to right. Icons can also be placed into an *icon box*. The icon box is an **mwm** window that holds icons. The icon box can also be minimized (turned into an icon). You can specify the creation of an icon box by setting a value in the X configuration file (see Chapter 5).

An icon also has a menu, which contains the same selections as the Window Menu of the associated client window. To display the icon's menu with the mouse, position the pointer anywhere on the icon and press Button 3. To display the icon's menu from the keyboard, press `Shift F10` or `Alt Space`. You browse through the menu either by dragging the mouse or pressing the arrow keys. To select an item, either release the mouse button when the item is highlighted or press `Return`. Menu selections that are not applicable are grayed out. Once the menu is posted, you can cancel it by pressing `Cancel` or by clicking Button 3 anywhere outside the menu.

Changing a Window into an Icon

To change a window into an icon, position the pointer on the *minimize button* on the window frame and click Button 1. The minimize button is located to the left of the maximize button. You can also iconify a window by selecting *Minimize* on the Window Menu or by pressing the accelerator `Alt F9` when the window is selected.

Moving Icons

To move an icon, position the pointer on the icon, press and hold down Button 1, and move the mouse. The icon appears in outline form. When the outline is in the position that you want, release the mouse button.

You can also initiate the move by selecting *Move* on the icon's menu or by pressing the accelerator `Alt F7`. Drag the mouse until the outline is where you want the icon to be placed. Click Button 1 or press `Return` to complete the move operation. In explicit focus policy mode, the window manager selects the icon for input focus after it has been moved.

Restoring a Window from an Icon

To restore a window from an icon, position the pointer on the icon and double−click Button 1. The icon changes back into the window, returning to its previous size and location. You can also change an icon back into a window by selecting *Restore* on the icon's Window Menu or by pressing the accelerator `Alt F5`.

Closing Windows

Closing a window removes it from the screen. To close a window, select *Close* from the Window Menu or press the accelerator `Alt F4` when the window is selected. If the window is the MainWindow of a running application, the application usually will terminate. Not all applications respond gracefully to this type of termination. Some applications may force you to perform additional steps to terminate if you close its MainWindow. If an application uses multiple independent primary windows, selecting *Close* may either close just that window or cause the application to exit.

Using the Default Root Menu

The *root menu* provides access to **mwm** functionality that relates to the screen as a whole. For example, the root menu may contain selections to create a new window or restart **mwm**. You can specify items for the root menu in the **mwm** configuration file (see Chapter 6).

To display the root menu, position the pointer anywhere on the root window and press mouse Button 3. Figure 2−6 shows the default Root Menu.

Figure 1 The Motif Default Root Menu

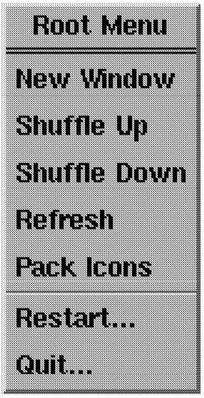


Table 2−5 lists the default root menu selections and their associated functions.

Table 1 The Default Root Menu Selections

This selection...	***Does this...***
New Window	Opens a new terminal emulation window on the screen
Shuffle Up	Raises the window on the bottom of the window stack
Shuffle Down	Lowers the window on the top of the window stack
Refresh	Redraws the contents of the entire screen
Pack Icons	Aligns icons into close formation
Quit	Exits **mwm**
Restart	Exits and then restarts **mwm**

Usually, you use *Restart* when you have changed **mwm** resources and you want the changes to take effect (see Chapter 4). When you select *Restart*, the window manager verifies that you really want to restart. Move the pointer to the control labeled **OK** and click Button 1 to restart; move the pointer to the control labeled *Cancel* and click Button 1 to cancel the procedure.

Using the mwm Virtual Desktop

The window manager provides virtual desktop capabilities to the Motif user. Some window managers limit the canvas upon which the user positions windows to the area visible on the display screen. The Motif window manager does not have this restriction. Instead, the display screen may be thought of as a viewport onto an infinite−sized canvas. This allows you to pan the view over a root window that appears to be much larger than the actual size of the display screen. The **mwm** can be panned to any location that a window can be placed. This allows you to position windows on a root window of virtually unlimited size.

Access to **mwm**’s virtual desktop capabilities is provided via the **mwm** and X resource files. There are basically three ways to move about and view objects on the virtual desktop:

- €You can make the viewport jump to an absolute location somewhere on the virtual canvas. For example, you can create a key binding such that, whenever you pressed that key, the top left of the viewport would automatically jump to coordinates 500,2000.
- €You can pan the viewport a relative distance from its current location. For example, you can create a key binding such that, whenever you pressed that key, the viewport would slide 10 pixels to the left.
- €You can interactively and continuously pan the viewport by sliding the mouse. For example, if you establish the appropriate mouse bindings, you can interactively slide the viewport left as you slide the mouse left.

The **mwm** provides these capabilities through resource files that you can set up and access via an **mwm** menu, or through key or button bindings. Refer to Chapter 6 for a description of menu functions and how they work.