Interacting with Motif Applications

Motif applications contain various visual components that enable you to interact with the program. Although no two Motif applications are identical, you can expect certain behaviors in all Motif applications. This chapter describes some of the components that you are likely to see in applications that comply with the guidelines established in the *Motif Style Guide*. This chapter provides information on the following topics:

- Exploring the MainWindow
- Using buttons
- Using Menus
- Using DialogBoxes
- Using adjustable components
- Using Lists
- Using ComboBoxes
- Using Containers
- Using Notebooks
- Using SpinBoxes
- Entering and editing text
- Using keyboard traversal
- Using drag and drop

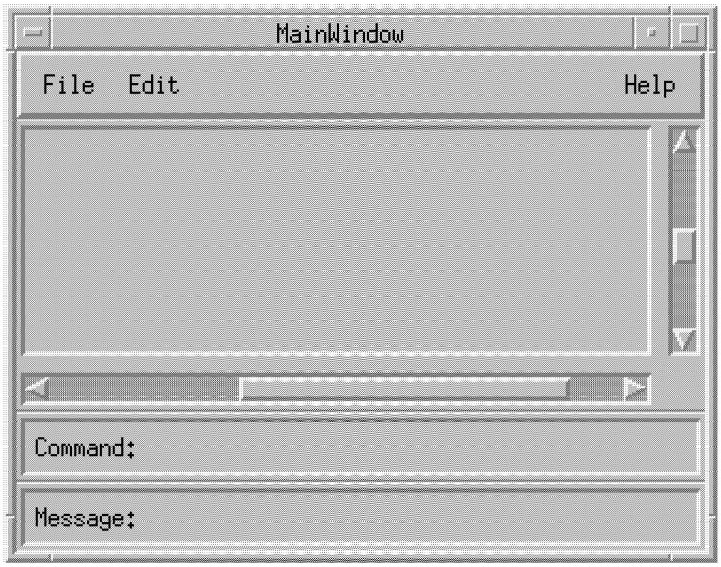
Exploring the MainWindow

Every Motif application contains a *MainWindow*, which organizes the contents of the application. Generally, you access primary functionality of an application through components in the MainWindow. The components included in a MainWindow will vary from application to application. The MainWindow in a typical application may contain a MenuBar, ScrollBars, and areas for entering commands and viewing messages, as well as the components that are specific to the application.

The *MenuBar* is a horizontal bar located just under the window’s title bar. The MenuBar organizes the most common features in an application. The area below the MenuBar contains the components for the primary functionality of the application. For example, in a spreadsheet program, the spreadsheet grid would be displayed in this area. In an electronic mail application, this area would contain components for reading and sending mail messages. Vertical and horizontal ScrollBars allow you to view data that requires more space than is available in the MainWindow at any one time. Applications that use a command−line interface in addition to a graphical interface provide a command area for entering commands. The command area normally runs from border to border across the bottom of the MainWindow. An application may also include a region for displaying short messages.

Figure 3−1 shows the basic components of a MainWindow for Motif applications. Most applications will have additional components in their MainWindows.

Figure 1 The Basic MainWindow for Motif Applications



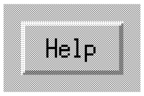
Using Buttons

Buttons are basic components that activate functionality in Motif applications. Buttons operate as standalone interface components and are also used in building Menus and DialogBoxes.

Using PushButtons

A *PushButton* contains a label that indicates the function of the button. The label can be either text or an image. The Motif PushButton has a 3−dimensional appearance. A PushButton appears raised out of the screen. When the button is pushed, it appears recessed into the screen. Figure 3−2 shows a typical PushButton.

Figure 1 A PushButton



To activate the functionality of a PushButton with the mouse:
1.Move the pointer to the PushButton.
2.Click Button 1.

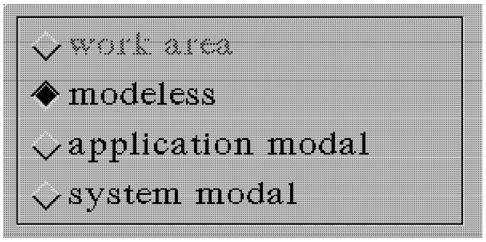
If the PushButton has the input focus, you can activate it from the keyboard by pressing `Select`or `Space`.

Using ToggleButtons

A *ToggleButton* is a button that has three states: on, off, and indeterminate. ToggleButtons typically allow you to select one or more options from a list of choices. Every ToggleButton has a label and a graphic that indicates its current state. Some of the possible indicators are checkboxes, crossboxes, plain boxes, diamonds, and circles.

RadioButtons are a group of ToggleButtons where only one of the ToggleButtons can be on at any one time, like an old−style car radio. RadioButtons do not have indeterminate states. The associated graphic indicator for the state of RadioButtons can be either a diamond or a circle. When the diamond or circle is filled in, it indicates that the button is on; when the diamond or circle is empty, it indicates that the button is off. Figure 3−3 illustrates a RadioBox with diamond graphic indicators.

Figure 2 RadioButtons



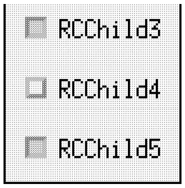
To select a RadioButton with the mouse:
1.Move the pointer to the RadioButton’s label.
2.Click Button 1.

Clicking on a RadioButton that is in the off state changes its state to on. Any other RadioButton in the group that is in the on state is set to the off state. Clicking on a RadioButton that is currently on causes the button to go on and off again quickly.

If a RadioButton has the input focus, you can turn it on by pressing `Select` or `Space`. Any RadioButton in the group that currently is set is turned off. You can navigate through a group of RadioButtons with the arrow keys.

CheckButtons are a group of ToggleButtons where any number of the ToggleButtons can be on at the same time. The CheckButton’s associated graphic indicator is usually a square, which is filled in or differently colored when the button is on (see Figure 3−4).

Figure 3 CheckButtons



To select a CheckButton with the mouse:
1.Move the pointer to the CheckButton’s label.
2.Click Button 1.

Clicking on a CheckButton in the off state changes its state to on. The button retains this setting as you select other CheckButtons. Clicking on a CheckButton in the on state changes its state to off or indeterminate, depending on how the CheckButton is set up. If the CheckButton is set up to be able to have an indeterminate state, then clicking on a CheckButton in the on state changes its state to indeterminate. Otherwise, clicking on a CheckButton in the on state changes its state to off.

If a CheckButton has the input focus, you can toggle between the on and off states by pressing `Select` or `Space`. You can navigate through a group of CheckButtons with the arrow keys.

Using Menus

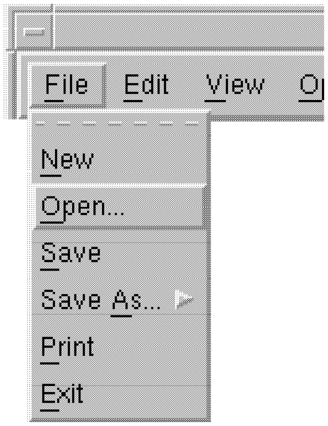
Menus provide controls for frequently used features of an application. Once you have selected a Menu, choosing an item on that Menu initiates an associated action. Remember that while a Menu is posted, you cannot interact with other components of the application. Motif supports these types of Menus:

- Pulldown Menus
- Popup Menus
- Option Menus
- TearOff Menus

Using Pulldown Menus

Pulldown Menus are accessed from the labels that represent them. When you select the label of a Pulldown Menu, the Menu pane appears like you had pulled down a window shade. The most commonly used Pulldown Menus in an application are placed on the MenuBar of an application. Figure 3−5 shows a File Menu pulled down from the MenuBar.

Figure 1 File Menu Pulled Down from the MenuBar



Pulldown Menus contain items that perform common application actions. Pulldown Menus can also contain nested submenus called *cascading Menus*.

Motif provides two different styles of mouse−based interaction with Pulldown Menus: *spring−loaded* interaction and *posted* interaction.

To select a Menu item using spring−loaded interaction:
1.Move the pointer to the name of the Menu.
2.Press and hold down Button 1 to display the Menu.
3.Drag the pointer down into the Menu pane and highlight the Menu item that you want to select.
4.Release Button 1 to select the Menu item.

During a spring−loaded interaction, you can examine the contents of each Menu on the MenuBar by pressing Button 1 and dragging the pointer along the Menu titles in the MenuBar. As you drag through each title, the corresponding Menu will be displayed. You can cancel a spring−loaded interaction by moving the pointer anywhere outside of the Menu system and releasing Button 1.

To select a Menu item using posted interaction:
1.Move the pointer to the name of a Menu.
2.Click Button 1 to post the Menu.

3.Click Button 1 on the Menu item that you want to select.

During a posted interaction, you can cancel the Menu by moving the pointer to the Menu name or anywhere outside of the Menu system and pressing Button 1.

To select a Menu item using the keyboard:
1.Press `F10` to select the MenuBar.
2.Use the left and right arrow keys to select a Menu.
3.Press `Return`, `Select`, or `Space` to post the Menu.
4.Use the arrow keys to move to the Menu item that you want to select.
5.Press `Return`, `Select`, or `Space` to activate the Menu item.

You can cancel a Menu using the keyboard by pressing `F10`. The input focus will return to the component that had it prior to the Menu interaction. Press `Cancel` to back out of the Menu hierarchy. If the input focus is in the MenuBar or a Menu pulled down from it, `Cancel` works just like `F10`. Otherwise, `Cancel` unposts just the last cascaded submenu.

In *Motif Style Guide* compliant applications, you can always find the *Exit* command in the first Pulldown Menu on the MenuBar, which is typically the File Menu. If an application includes a Help Menu, it is always the farthest right Menu on the MenuBar. Compliant applications may also use other standard Menus that are defined by the *Motif Style Guide*, such as the Edit, View, and Options Menus.

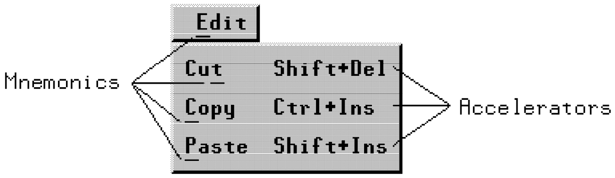
Using Menu Shortcuts

In addition to the item name, a Menu selection can also provide additional visual clues to its functionality. For example, the presence of a cascading submenu is indicated by a triangle at the right side of a Menu item. A grayed out item name indicates that the selection is not valid at the current time. The ... (ellipsis points) after a Menu item means that the application requires more information to perform the action of the Menu item.

An application can provide rapid keyboard access to Menus and Menu items by specifying mnemonics. If a Menu or a Menu item has a mnemonic associated with it, the mnemonic character is underlined in the label for the Menu or Menu item. Typing `Alt` plus the mnemonic for a Menu posts the associated Menu. Once a Menu is posted, typing the mnemonic for a Menu item is equivalent to selecting the associated Menu item. Mnemonics for Menu items only work when the Menu containing the items is posted.

Some applications use a combination of keystrokes, called accelerators, to create shortcuts for accessing Menu items. If a Menu item has an accelerator associated with it, the required keystrokes are shown to the right of the item name. Entering an accelerator sequence activates the Menu item whether or not the Menu is posted. Figure 3−6 shows a Pulldown Menu that provides both mnemonics and accelerators.

Figure 2 Menu Mnemonics and Accelerators



Using Popup Menus

Different areas of an application can have *Popup Menus* associated with them. You can only access a Popup Menu when the pointer is over an area that is associated with a Popup Menu. A Popup Menu does not provide any visual clues to let you know of its existence. Therefore, Popup Menus only provide short cuts to functionality that is available elsewhere in an application. The contents of a Popup Menu can be context sensitive, which means that the items on the Menu can vary based on a number of factors, such as the location of the pointer when the Menu is posted or the selection state of a component.

To activate a Popup Menu with the mouse:
1.Position the pointer on a component that contains a Popup Menu. Refer to the documentation for the particular application to find out where the Popup Menus are located.
2.Press Button 3.

Popop Menus provide both spring−loaded and posted interaction styles. Select items on a Popup Menu just as you would on a Pulldown Menu.

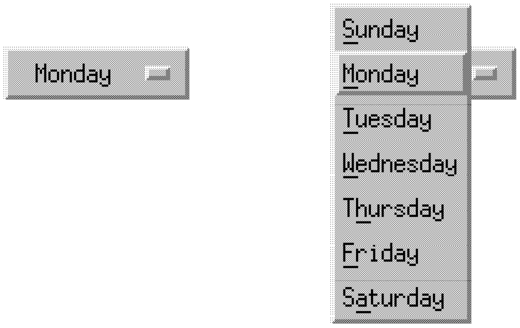
You can select items on a posted Popup Menu using either Button 1 or Button 3. To cancel a posted Popup Menu, click Button 1 or Button 3 outside of the Menu pane.

If the input focus is in an area where there is an inactive Popup Menu, you can post the Menu by pressing `Menu` or `Shift F10`. You can select an item on a Popup Menu with the keyboard by using the arrow keys and pressing `Return`. To cancel a Popup Menu from the keyboard, press `Cancel`.

Using Option Menus

An *Option Menu* presents a list of choices within a relatively small space. A bar graphic on the right side of an OptionButton signifies the presence of an Option Menu. When you select an OptionButton, the associated Option Menu is presented and you can make a selection. When the Option Menu is not posted, the OptionButton displays the current selection. Figure 3−7 shows an OptionButton with its bar graphic and the associated Option Menu.

Figure 3 An OptionButton and Associated Option Menu



To post an Option Menu with the mouse:
1.Move the pointer to the OptionButton.
2.Press Button 1.

Option Menus provide both spring−loaded and posted interaction styles. Select an item on an Option Menu just as you would on a Pulldown Menu. To cancel a posted Option Menu, click Button 1 on the bar graphic or anywhere outside of the Menu pane.

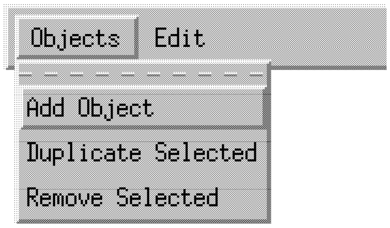
If an OptionButton has the input focus, you can post the Menu from the keyboard by pressing `Select` or `Space`. Select an item on the Option Menu by using the arrow keys and pressing `Select` or `Space`. To cancel an Option Menu from the keyboard, press `Cancel`.

Using TearOff Menus

TearOff Menus allow you to keep frequently used Menus on the screen. Normally, a Menu disappears after you make a selection, but a TearOff Menu remains posted until you cancel it. TearOff Menus are so called because they imitate in appearance a coupon that is torn off.

Pulldown Menus, Popup Menus, and Option Menus can all be TearOff Menus. If the tear−off feature is available on a particular Menu, a TearOffButton with a dashed line representing perforations is shown at the top of the Menu (see Figure 3−8).

Figure 4 A TearOff Menu



To tear off a Menu:
1.Post the Menu using either the keyboard or the mouse.
2.Select the TearOffButton as you would any other Menu entry. You can use the arrow keys or the mouse to move to the TearOffButton.

The Menu is torn off and placed at the location where it originally appeared. You can move it to a new location as you would any other window.

To tear off a Menu and move it to a new location:
1.Post the Menu using either the keyboard or the mouse.
2.Press and hold down Button 2 over the TearOffButton.
3.Move the Menu to the location where you want the Menu and release Button 2.

After a Menu is torn off, the original Menu is unposted. A torn−off Menu is like an ordinary secondary window. It is decorated with a title bar, window border, and window manager buttons. The torn−off Menu can be repositioned by using the title bar. You can have only one copy of each TearOff Menu at a time. If you tear off a Menu that you have already torn off, the first version will disappear and the new version will replace the older version. At any time, you can post the Menu in its usual location without affecting the torn−off version.

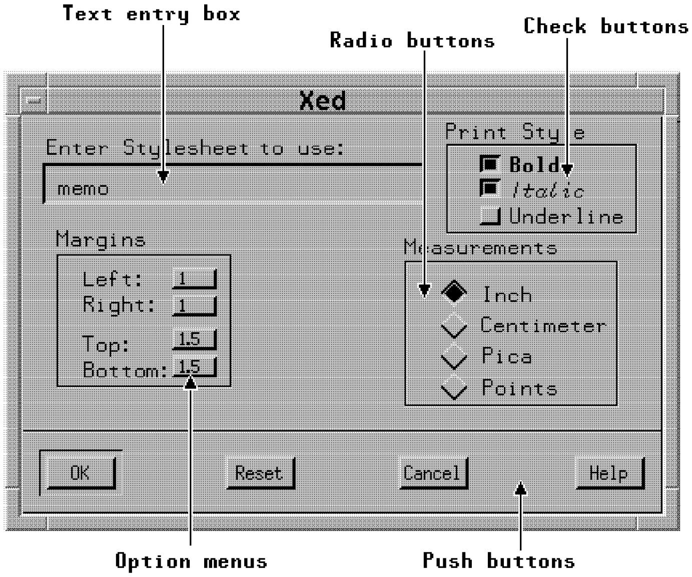
To remove a TearOff Menu, you can select *Close* from the Window Menu or press `Cancel` while the input focus is in the TearOff Menu.

Using DialogBoxes

Motif applications use *DialogBoxes* for tasks that are of secondary importance in the application. A DialogBox may provide you with information, such as an error message, or ask you to enter information, such as a file to open. DialogBoxes may limit how you can interact with other windows in an application. A *modal* DialogBox requires that you respond before you can interact with any other components in the application. A *modeless* DialogBox does not limit your interaction with the rest of an application, so it can remain available for the duration of an application and be used to perform multiple actions.

Applications can design DialogBoxes with any number and any combination of components. Figure 3−9 shows a DialogBox that contains many different components for adjusting the settings for a text editor.

Figure 1 A Typical DialogBox



Xed

Enter Stylesheet to use: memo

Margins

Left: 1

Right: 1

Top: 1.5

Bottom: 1.5

Print Style

■ Bold

■ *Italic*

□ Underline

Measurements

◆ Inch

◇ Centimeter

◇ Pica

◇ Points

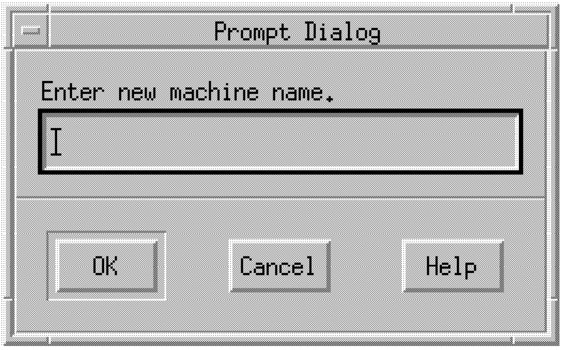
At a minimum, a DialogBox contains a Label and a PushButton for supplying a response. The Motif toolkit provides the following types of DialogBoxes for general functions that are common to many applications:

- *PromptDialogs* for entering input
- *CommandDialogs* for entering keyboard commands
- *SelectionDialogs* for making a selection from a list of choices
- *FileSelectionDialogs* for entering the name of a file for processing
- *MessageDialogs* for displaying information

Each of these DialogBoxes defines its own user interaction.

An application typically uses a PromptDialog to get information that it needs to continue processing. A PromptDialog usually interrupts your interaction with the rest of the program. The PromptDialog includes a message, a Text input area where you type the requested information, and PushButtons for responding to the DialogBox. Figure 3−10 shows a PromptDialog.

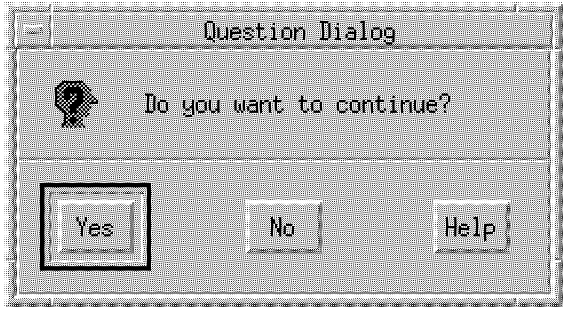
Figure 2 A PromptDialog



Enter new machine name.

Motif also provides a number of different types of MessageDialogs, including an *ErrorDialog*, an *InformationDialog*, a *QuestionDialog*, a *WarningDialog*, and a *WorkingDialog*. Each type of MessageDialog contains a graphical symbol that represents its purpose, a textual message, and PushButton commands that are appropriate for the DialogBox. Figure 3−11 shows a QuestionDialog. A QuestionDialog contains a question symbol, a message that asks you a question and PushButtons that allow you to answer the question.

Figure 3 A QuestionDialog



Do you want to continue?

Every DialogBox contains a row of PushButtons at the bottom of the box that allow you to respond to the DialogBox. The *Motif Style Guide* defines a number of common actions for DialogBoxes. The intent of the common actions is to provide you with consistent behavior in different Motif applications. In *Motif Style Guide* compliant applications, the common DialogBox actions have the following meaning and behavior:

- *Yes* and *No* are answers to clearly posed questions. Selecting either of these buttons closes the DialogBox.
- **OK** applies any changes that you have made in the DialogBox and closes the DialogBox.
- *Close* closes the DialogBox without performing any action.
- *Apply* applies any changes that you have made in the DialogBox but does not close the DialogBox.
- *Retry* attempts the task in progress again. You will typically see this action in ErrorDialogs.
- *Stop* ends the current task at the earliest possible breaking point. This action is commonly found in WorkingDialogs.
- *Pause* pauses the task that is currently in progress. You will usually find this action in WorkingDialogs.
- *Resume* is used in conjuction with *Pause*. It causes a paused task to be resumed.
- *Reset* cancels any changes that have not already been applied to the application.
- *Cancel* closes the DialogBox without performing any actions that have not already been applied to the application.
- *Help* provides information about the dialog.

Using Adjustable Components

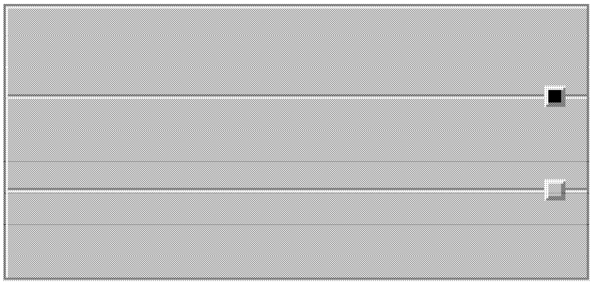
Motif includes the following adjustable components:

- *PanedWindows* for adjusting the relative size of components
- *ScrolledWindows* for viewing the entire contents of a component when the viewing area cannot accomodate all of the contents at any one time
- *Scales* for selecting a value within a range

Using PanedWindows

A PanedWindow groups components into separate panes. Panes are distinct sections within one window. Adjacent panes are separated from one another by a horizontal separating line and a *Sash* (see Figure 3−12). The Sash allows you to adjust the relative size of adjacent panes. You move the Sash to change the position of the separating line between two panes, which changes the size of the panes. As the Sash moves, the pane located in the direction of the movement gets smaller while the other pane gets proportionally larger.

Figure 1 A PanedWindow



To adjust a PanedWindow with the mouse:

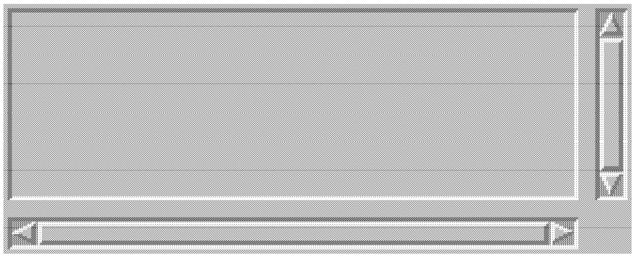
1.Move the pointer to the Sash.
2.When the pointer shape changes to a + (plus sign), press and hold down Button 1.
3.Drag the mouse until the panes are in the proportion you want.
4.Release Button 1.

If a Sash has the input focus, you can adjust a PanedWindow using the keyboard. Use the up and down arrows keys to move the separating line in small increments. You can move the separating line by larger increments by holding down `Ctrl` as you press the arrow keys.

Using ScrolledWindows

A ScrolledWindow frames other components and provides *ScrollBars* as necessary to view the contents of the window. You use ScrollBars to adjust the visible contents of a window when the entire contents do not fit in the window. A ScrollBar consists of a movable component, called a slider, which moves within a trough running the length of the scrolled region. The position and size of the slider within the trough indicates the relative position and size of the visible area of the scrolled component. A ScrollBar also provides arrow buttons at each end of the trough for moving the slider (see Figure 3−13).

Figure 2 A ScrolledWindow



In addition to the illustrated slider, a ScrollBar's slider visual can also be an etched line, a shadowed circle, or three etched lines in the middle of the slider.

You can use the following techniques to scroll a region with the mouse:

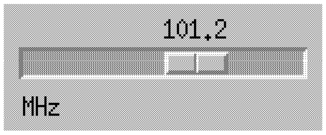
- To move the slider to a particular location, press Button 2 in the trough. The slider will jump to that location. You can then drag the slider back and forth to further scroll the region.
- To drag the slider to a particular location, move the pointer to the slider. Press and hold down Button 1. You can then drag the slider back and forth to any position you want.
- To scroll up or down one page (using vertical ScrollBars), click Button 1 on the trough above or below the slider. Horizontal ScrollBars provide the same mechanism for scrolling left or right by a page.
- To scroll the region one unit at a time, click Button 1 on the arrow button that corresponds to the direction that you want to scroll the region.

To navigate through a ScrolledWindow using the keyboard, see the following sections on Lists and Text components for specific information.

Using Scales

A Scale is used to select a value from a continuous range. Like a ScrollBar, a Scale consists of a slider within a trough and arrow buttons. A Scale can be oriented vertically or horizontally. The position of the slider indicates the current value relative to the whole range. The current value of a Scale may be displayed next to the slider. A Scale can also contain Labels that specify the value of the Scale when the slider is at particular locations (see Figure 3−14).

Figure 3 A Scale



In addition to the illustrated slider, a Scale's slider visual can also be a rectangle, a shadowed circle, or three etched lines in the middle of the slider.

You can use the following techniques to manipulate a Scale with the mouse:

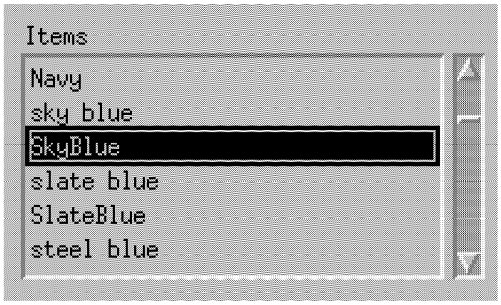
- To move the slider to an arbitrary location, press Button 2 in the trough. The slider will jump to that location. You can then drag the slider back and forth to adjust the value of the Scale.
- To drag the slider to a particular location, move the pointer to the slider. Press and hold down Button 1. You can then drag the slider back and forth to any position you want.
- To adjust the Scale one unit at a time, click Button 1 on the arrow button that corresponds to the direction that you want to change the value.

When a Scale has the input focus, you can adjust its value using the keyboard. To move the slider in single increments, press the appropriate arrow key. To move the slider in larger increments, press and hold down `Ctrl` as you press the appropriate arrow keys.

Using Lists

A *List* component presents a list of items for selection. Motif provides List components that allow you to select single items, multiple items, or discontiguous ranges of items. An application can use Lists to handle different types of selections; the purpose of the List within the application determines the number of items that you can select. A List can also provide vertical and horizontal ScrollBars when the contents of the List are larger than its window. The selected item(s) in a List are highlighted (see Figure 3−15).

Figure 1 A Single−Selection List



To select a single item from a List:
1.Move the pointer to the item that you want to select.
2.Click Button 1.

A single−selection style List may also support browse−style selection. Browse−style selection allows you to select an item by pressing Button 1 on an item, dragging the pointer through the List, and releasing Button 1 on the item that you want to select.

To select a contiguous group of items from a List:
1.Move the pointer to the first item in the group.
2.Press Button 1 and drag the pointer to the last item in the group.
3.Release Button 1.

You can also select a group of items by selecting a single item, moving the pointer to the last item in the group, and pressing `Shift` as you click Button 1. All of the items between the first and the last items will be selected.

To add a discontiguous group of items to a selection:
1.Move the pointer to the first item in the discontiguous group.
2.Press and hold down `Ctrl` as you press Button 1.
3.Drag the pointer to the last item in the group.
4.Release Button 1.

If a List has the input focus, you can navigate through the items in the List and make selections using the keyboard. List components use a *location cursor* to indicate which item in the List has the input focus. The location cursor is shown by a box around a List item. Use the up and down arrow keys to move the location cursor through the items in the List. Depending on the type of List you are using, the selection may move with the location cursor. If the selection does not move with the location cursor, you can select an item by pressing `Select` or `Space.`

The List component allows you to make all of the different types of selections from the keyboard. For more information on keyboard−based selection, see the *Motif Style Guide.*

In a List that allows you to select multiple items, you can select all of the items in the List by pressing `Ctrl /`. You can deselect all of the items in the List by pressing `Ctrl \`.

You can cancel an in−progress drag−based selection operation by pressing `Cancel` at any time during the operation.

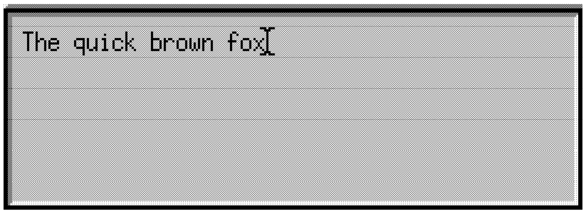
Entering and Editing Text

Many applications include areas for entering and editing text. The *Text* entry area may be a single line for entering the answer to a question in a DialogBox. However, the Text area can be any length, depending on the needs of the application.

> **Note:** This guide assumes that you are using applications in an English locale, where text is entered horizontally and wraps from one line to the next. In a locale such as Japanese, text is entered vertically and is wrapped from one column to the next. For information on using applications based on other locales, refer to Chapter 4.

Text components use an *insertion cursor* to indicate where the information that you are typing will be inserted. When a Text component has the input focus, the insertion cursor is indicated by a blinking I−beam cursor (see Figure 3−16).

Figure 1 A Text Component



The quick brown fox

To move the Text insertion cursor:
1. Move the pointer to the position where you want to begin typing.
2. Click Button 1. The I−beam insertion cursor will move to that location to let you know that you can begin typing.

If a Text component has the input focus, you can navigate through the text using the keyboard. The left and right arrows keys move the insertion cursor by a character. The up and down arrow keys move the insertion cursor by a line. Pressing `Ctrl` and using the left and right arrows navigates by words. Pressing `Ctrl` and using the up and down arrows navigates by paragraphs. `PageUp`, `PageDown`, `PageLeft` or `Ctrl PageUp`, and `PageRight` or `Ctrl PageDown` move the insertion cursor by pages.

You can use any of the following methods to select Text:
- Press Button 1 and drag over the region that you want to select.
- Click Button 1 twice in a word to select that word.
- Click Button 1 multiple times to select larger areas, as determined by the application.

To deselect a region with the mouse, move the pointer anywhere outside of the selected region and click Button 1. If you want to move the insertion cursor without changing the selection, move the pointer to the location where you want to begin typing, hold down `Ctrl`, and click Button 1.

Text components provide four ways to copy or move text within the same component or from one component to another: clipboard transfer, drag transfer, primary transfer, and quick transfer.

To use clipboard transfer:
1. Select the text that you want to copy or move.
2. Type `Ctrl Insert` to copy the text to the clipboard or `Shift Delete` to cut it to the clipboard.
3. Move the insertion cursor to the location where you want to insert the text.
4. Type `Shift Insert` to paste the text to the new location.

To move or copy text using drag transfer:
1. Select the text that you want to copy or move.
2. Move the pointer over the selected text and press Button 2.
3. Drag the pointer to the location where you want to insert the text.
4. Release Button 2. Typically, the text will be moved to the new location. Hold down `Shift` while releasing Button 2 to ensure a move operation. Hold down `Ctrl` while releasing Button 2 to ensure a copy operation. (See Section 3.9 for more detailed information about drag transfer.)

To use primary transfer:

1.Select the text that you want to copy or move.
2.Move the pointer to the location where you want to insert the selected text.
3.To copy the text, click Button 2. To move the text, hold down `Shift` and click Button 2.

To move or copy text using quick transfer:

1.Move the pointer to the location where you want the text to be inserted.
2.Click Button 1.
3.Move the pointer to the location of the text that you want to move or copy.
4.Hold down `Alt` and press and hold down Button 2 as you drag the pointer over the text that you want to transfer. The text in the region you select will be underlined.
5.Release Button 2 to copy the text. To move the text, press and hold down `Alt` and `Shift` as you make your selection and release Button 2.

You can cancel an in−progress quick transfer operation by pressing `Cancel` before releasing Button 2 or by releasing Button 2 outside of the Text component.

Note that which mouse buttons are to be used for text transfer depends on how these buttons have been defined for your mouse. In some cases, Button 1 is used to both select and transfer the text and Button 2 is used to adjust or extend the selection and for primary transfer, and in other cases, Button 1 is used to select the text, and Button 2 is used to transfer it.

You can also select, copy, and move text using the keyboard instead of the mouse. To make a selection, press and hold down `Shift` and use the various Text navigation keys to highlight the text that you want to select. Release `Shift` to complete the selection.

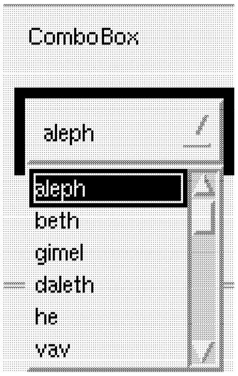
Once the selection has been made, any further use of the navigational keys will deselect the region. You can either use the mouse as previously described to copy or move the selection or press `Shift F8` to enter add mode. When you are in add mode, you can move the insertion cursor without causing the selection to be deselected. After moving the insertion cursor, press `Alt Ctrl Insert` to copy the selection. Press `Alt Shift Delete` to move the selection.

To delete text within a Text component, use `BackSpace` and `Delete`. If text is selected, either key will delete the selection. If there is no selected text, `BackSpace` deletes the character preceeding the cursor and `Delete` deletes the character following the cursor.

Using ComboBoxes

A ComboBox combines the capabilities of the TextField and List widgets. Applications use ComboBoxes to allow users to either type in information as their selection or to select information from a list of choices provided by the application. If the List contains more items than are displayed, a vertical ScrollBar appears next to the List. Figure 3−18 illustrates a ComboBox.

Figure 1 ComboBox

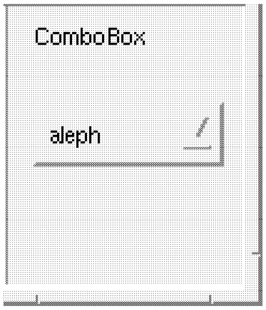


An item can be selected in a ComboBox in many ways, using either the mouse or the keyboard. To select an item using the mouse:
1.Move the mouse pointer to the desired item.
2.Press Button 1 on that item.

If the ComboBox has the input focus, List items can be traversed to and selected with the keyboard by using the up or down arrow keys. Traversing to the item will select the item, as will pressing `Return`.

The ComboBox illustrated in Figure 3−18 is a variation of the ComboBox called a drop−down ComboBox. In a drop−down ComboBox, the List is hidden until requested by the user. The only things showing in a drop−down ComboBox before such a request is a downward−pointing arrow next to the TextField. Figure 3−19 illustrates the drop−down ComboBox before the list has been dropped.

Figure 2 Standard ComboBox



To drop down, or show, the List, using the mouse:
1.Move the mouse pointer to the arrow.
2.Press Button 1.

The List can be unposted by pressing Button 1.

For the keyboard, if the ComboBox has the input focus, you can press `Ctrl` and the down arrow or `Ctrl` and the up arrow to display the List.

The List can be unposted by pressing `Ctrl` and the up arrow, `Ctrl` and the down arrow, `Return`, or `Cancel`. Note that pressing `Return`will not only unpost the List, but also show the current contents of the TextField.

Some ComboBoxes have text entry fields that are non−editable. You can tell whether or not the TextField is editable by its appearance; editable text entry fields appear inset, while non−editable text entry fields do not.

You can traverse to the ComboBox’s text field whether it is editable or not, using the mouse:
1.Move the mouse pointer to the text entry field.
2.Press Button 1 on the field.

To traverse to the ComboBox’s TextField from the keyboard, you use the `Tab`key to first get to the ComboBox.

When the ComboBox has focus, a highlight outline, if a thick enough one was specified, is placed around the ComboBox.

In both cases, if the TextField is editable, subsequent keystrokes apply to the editable TextField, and the selection is typed directly into it. When an item in the List is selected, the List item is highlighted. Note that if you are entering the name of an item that already exists in the List, then you must enter the item’s entire name to select that item. However, ComboBox does not automatically select a List item if you type that string into the TextField. It selects the item when you Press Button 1 on the down arrow, move the focus, or hit `Return` or `osfActivate.` The application may or may not allow you to enter a selection name that does not already exist in the List. This depends on how the application is configured. For example, if the ComboBox contains a List of existing directory names, typing in a non−listed directory name should cause an error.

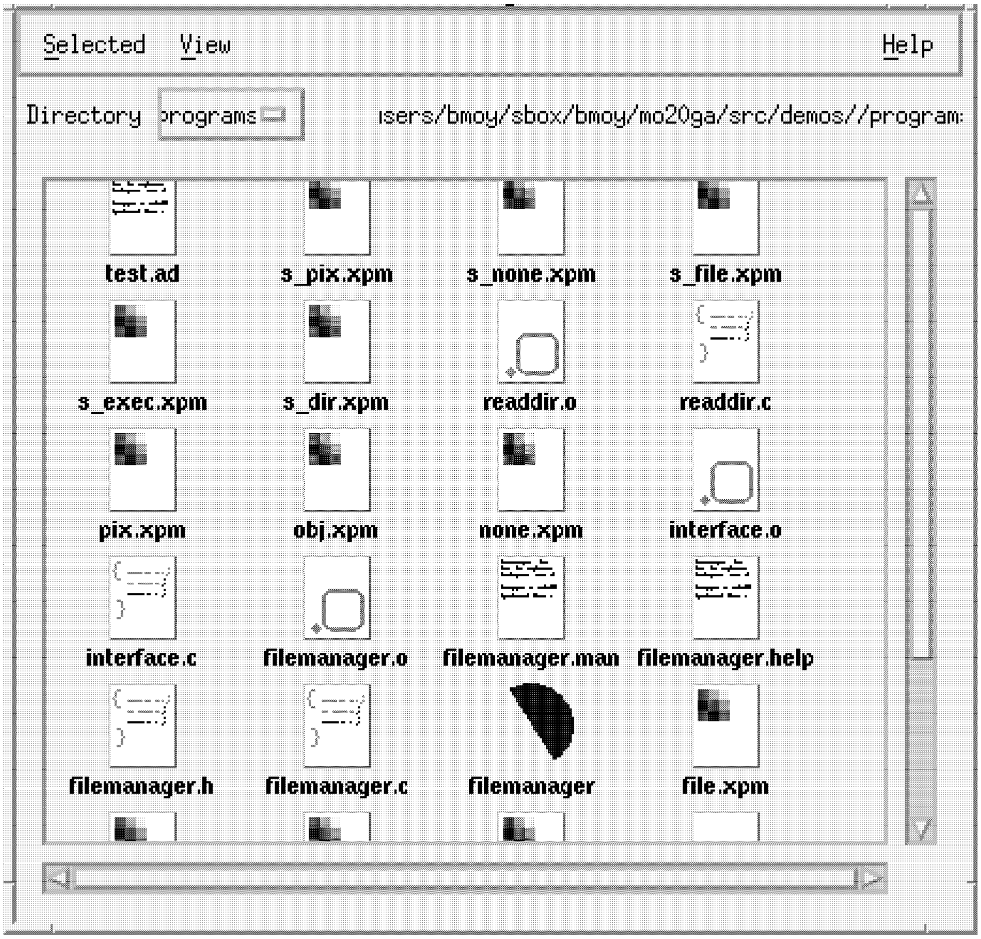
If the ComboBox does not have an editable text entry field, typed text will not appear in the TextField. Instead, depending on how the ComboBox is configured, either the ComboBox will ignore what is being typed, or the ComboBox will invoke a matching algorithm. If the ComboBox is configured to activate a matching algorithm, then the ComboBox attempts to match the text, as it is being typed, with an item in the List. When the ComboBox finds a match, even if it is only from the first letter of an item, it displays the matched item in the text entry field.

Using Containers

A *Container* component presents information about application−defined objects on a 2−dimensional background for selection and/or manipulation. Each object is represented by an icon, which can display a text label and either a large or small pixmap. Container can present objects in one of three views: Large−Icon, Small−Icon, or Any−Icon.

The Large−Icon view forces each object in the Container to display its label and large pixmap. The Small−Icon view forces each object in the Container to display its label and small pixmap. The Any−Icon view allows each object to use its own defaults for specifying what to display. The icons are displayed on a 2−dimensional background (see Figures 3−24 and 3−25). The location of each icon is determined by the application, but the user can change the icon location within the Container by dragging the icon and dropping it onto a new location within the Container.

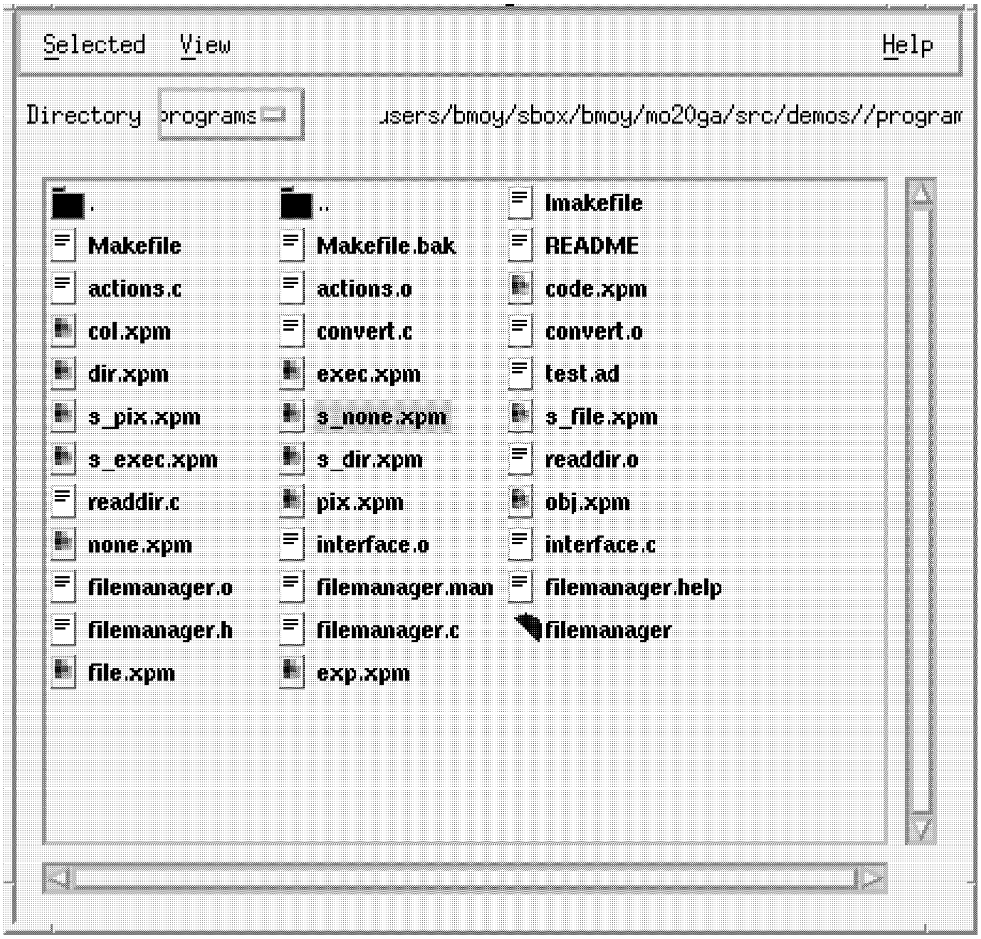
Figure 1 Container in the Large−Icon View



Directory programs

ısers/bmoy/sbox/bmoy/mo20ga/src/demos//program:

Figure 2 Container in the Small−Icon View



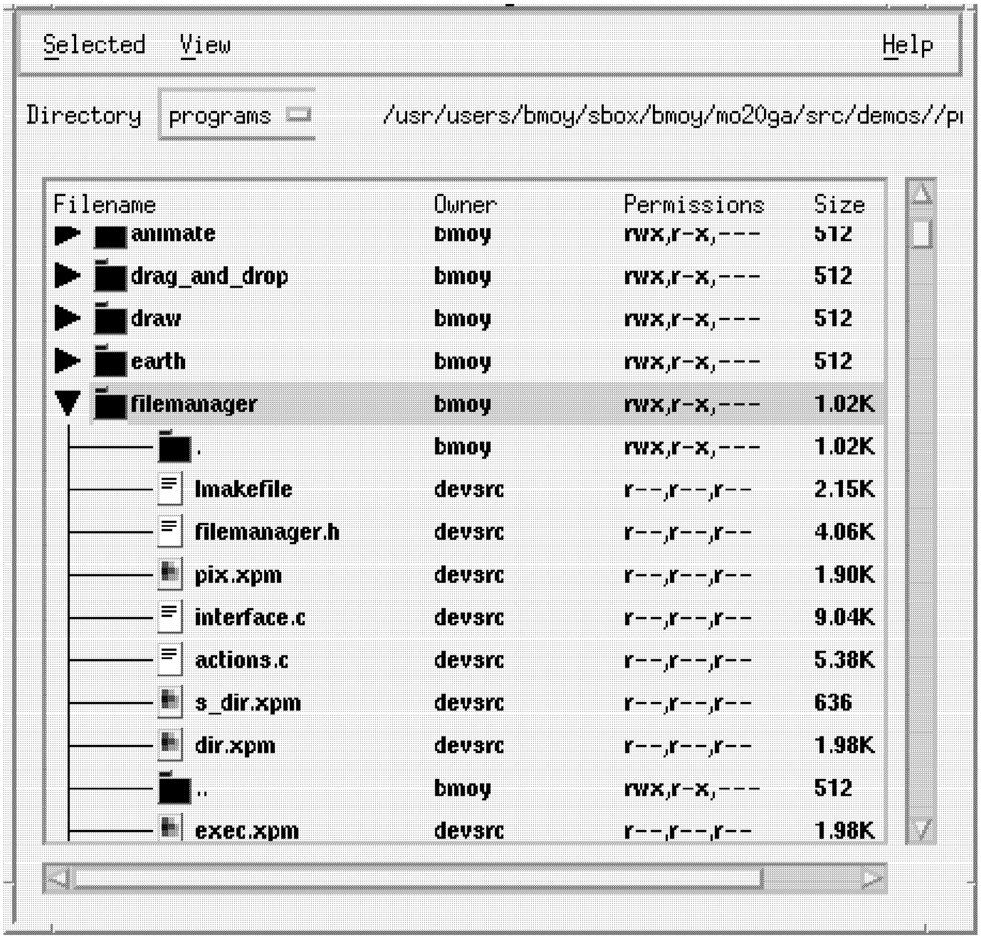
Directory programs

users/bmoy/sbox/bmoy/mo20ga/src/demos//program

A Container also provides an Outline Layout option that allows an application to show parent−child relationships between icons. A PushButton component is displayed next to each icon that has children. The user can press and release the PushButton to display or hide the children of a particular icon. The user cannot change the locations of icons when the Outline Layout option is displayed.

Detail layout is the same as Outline Layout with the addition that the objects' information is presented as a list of items, where each item has several columns. All columns except the first contain textual information about the object; the first column usually contains an icon in either the Small−Icon or Large−Icon view. The items are displayed from top to bottom and the user cannot change their locations (see Figure 3−27).

Figure 3 Container with Outline Layout



Directory programs /usr/users/bmoy/sbox/bmoy/mo20ga/src/demos//pı

Filename	Owner	Permissions	Size
animate	bmoy	rwx,r-x,---	512
drag_and_drop	bmoy	rwx,r-x,---	512
draw	bmoy	rwx,r-x,---	512
earth	bmoy	rwx,r-x,---	512
filemanager	bmoy	rwx,r-x,---	1.02K
.	bmoy	rwx,r-x,---	1.02K
Imakefile	devsrc	r--,r--,r--	2.15K
filemanager.h	devsrc	r--,r--,r--	4.06K
pix.xpm	devsrc	r--,r--,r--	1.90K
interface.c	devsrc	r--,r--,r--	9.04K
actions.c	devsrc	r--,r--,r--	5.38K
s_dir.xpm	devsrc	r--,r--,r--	636
dir.xpm	devsrc	r--,r--,r--	1.98K
..	bmoy	rwx,r-x,---	512
exec.xpm	devsrc	r--,r--,r--	1.98K

Selection

A Container has the capability to allow the user to select single items, multiple items, or noncontiguous ranges of items. An application that uses the Container component determines which types of selections are appropriate for the application.

Single Selection

To select a single item from a Container:
1.Move the pointer to the item that you want to select.
2.Click Button 1.

Browse Selection

A single−selection style Container may also support Browse−style selection. To use Browse−style selection:
1.Press and hold Button 1 over an item.
2.Drag the pointer through the Container.
3.Release Button 1 on the item that you want to select.

Group Selection: Detail or Outline Layout Only

To select a contiguous group of items from a Container that is in the Detail view (see Figure 3−26) or is using the Outline Layout option:

1.Move the pointer to the first item in the group.
2.Press and hold Button 1 and drag the pointer to the last item in the group.
3.Release Button 1.

You can also select a group of items by selecting a single item, moving the pointer to the last item in the group, and pressing `Shift` as you click Button 1. All items between the first and last items are selected.

To add a noncontiguous item or group of items to existing selected items in a Container that is using the Outline or Detail Layout option:

1.Select the first group of items with either of the previous methods.
2.Move the pointer to the first item in the next group.
3.Press and hold down `Ctrl` as you press Button 1.
4.Drag the pointer to the last item in the group.
5.Release Button 1.

Group Selection: Spatial Layouts Only

To select an item or a group of items from a Container that is using the Spatial Layout option: (see Figure 3−25):

1.Move the pointer to the first item.
2.Press Button 1 and drag the pointer over each item to select.
3.Release Button 1.

To add a noncontiguous item or group of items to existing selected items in a Container that is using the Spatial Layout option:

1.Select the first group of items with any of the previous methods.
2.Move the pointer to the first item in the next group.
3.Press and hold down `Ctrl` as you press Button 1 and drag the pointer over each item to select.
4.Release Button 1.

Group Selection: Spatial or Outline Layout

To select an item or a group of items from a Container that is using the Spatial or Outline Layout:

1.Move the pointer to a blank space in the Container.
2.Press Button 1 and drag the pointer to create a box around all the items to be selected.
3.Release Button 1.

To add a noncontiguous item or group of items to existing selected items in a Container that is using the Spatial or Outline Layout:

1.Select the first group of items with any of the previous methods.
2.Move the pointer to a blank space in the Container.
3.Press and hold down `Ctrl` as you press Button 1 and drag the pointer to create a box around all the items to be selected in the additional group.
4.Release Button 1.

Keyboard Selection

The Container component also supports keyboard selection. If a Container has the input focus, you can use the keyboard to navigate through the items and make selections. Container components use a location cursor to indicate which item has the input focus. The location cursor is shown by a box around a Container item. Use the up and down arrow keys to move the location cursor through the items. Depending on the type of Container you are using, the selection may move with the location cursor. If the selection does not move with the location cursor, you can select an item by pressing `Return` or `Space`.

The Container component allows you to make all of the different types of selections from the keyboard. For more information on keyboard −based selection, see the *Motif Style Guide*.

In a Container that allows you to select multiple items, you can select all of the items by pressing `Ctrl−/`. You can deselect all of the items in the Container by pressing `Ctrl−\`.

You can cancel a drag−based selection operation in progress by pressing `Cancel` or `Esc` at any time during the operation.

Using Notebooks

A *Notebook* simulates a real notebook by grouping components into separate pages. It also contains components for accessing pages and describing pages. Only one page can be seen at a time (see the example Notebook application in Figure 3−28). The page shown in the example is a Container component that displays dialog icons; however, the Notebook can be used to contain any component and so can be used for many different applications.

The Page Scroller component (shown on the lower right corner of the page Figure 3−28) can be used to display different pages of the Notebook much as a reader would flip pages back and forth in a book. By default, a SpinBox component is used as the Page Scroller but the application can specify a different component. For example, an application could specify a ScrollBar component to be used as the Page Scroller component of the Notebook.

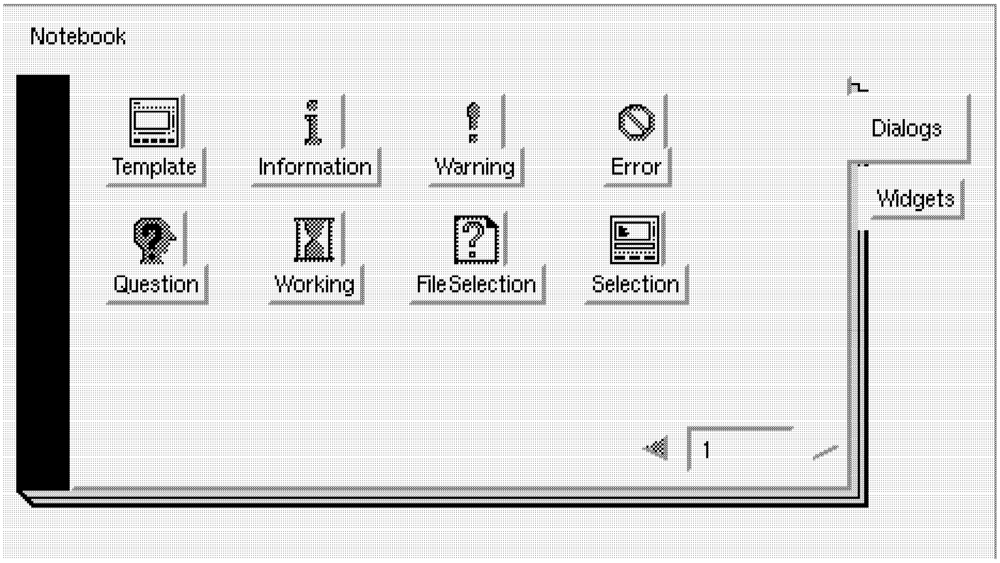
You can also change the page displayed by activating application−supplied button Tabs, which the Notebook displays as Major Tabs and Minor Tabs. Major Tabs are used by applications to divide the pages of the Notebook into major sections. In the example shown in Figure 3−28, Major Tabs (on the right−hand side of the page in the example) are used to divide the pages into dialogs and widgets. Minor Tabs are used by applications to subdivide the major sections. This example does not use Minor Tabs. If the Notebook is too small to show all the available Major or Minor Tabs, then Tab Scroller buttons are displayed on each side of the row or column of Tabs; you can activate these Tab Scroller buttons to scroll to the Tab you wish to see.

The Status area (which this example does not use) is used by the application to display additional information about a page. It is supplied by the application and can be different for each page shown.

Notebooks can contain five fields: the page itself, Page Scroller, Major Tabs, Minor Tabs, and Status area. However, the tabs and Status area are not required fields. Their use depends on the needs of the particular application.

For keyboard navigation, Notebook components are grouped into five fields: the page, Page Scroller, Major Tabs, Minor Tabs, and Status area. However, since all components are or can be application−supplied and may not be traversable, not all five fields may be present. See section 3.12 for instructions on keyboard navigation.

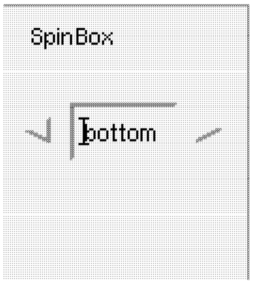
Figure 1 A Notebook



Using SpinBoxes

A *SpinBox* allows you to display, in sequence, a ring of choices. The SpinBox consists of a non−editable text area where a single choice is displayed; there are two arrow buttons next to the the text area, one on each side, for cycling through the choices. (See Figure 3−29.)

Figure 1 A SpinBox



To use a SpinBox to change the displayed choice:

1.If a SpinBox has more than one text area, click Button 1 on, or tab to, the text area you want to change. Whichever text area has focus is the area that will be "spun" by the arrow buttons.
2.Move the pointer to the SpinBox’s left or right arrow.
3.Click Button 1 on an arrow.

- To advance or retreat by one choice, click Button 1 once on an arrow button.
- To "spin" forward or backward through all choices, press and hold Button 1 on an arrow, similar to setting a digital clock. When the desired choice is displayed, release the mouse button.

One arrow acts as the increment arrow, and the other as the decrement arrow. The increment arrow lets you progress through the list of choices or increase the numeric value. For example, if the SpinBox displays a date, the increment arrow changes "March" to "April." If the SpinBox displays a numeric value (for example, a quantity), the increment arrow increases the value. Similarly, the decrement arrow lets you back up through the list (from "March" to "February"), or decrease the numeric value.

If you have clicked on SpinBox arrows or text, or if you have tabbed to the SpinBox via the keyboard, the SpinBox has input focus. You can then press the arrow keys on the keyboard to spin the choices, instead of pressing Button 1 on a displayed arrow.

Using Keyboard Traversal

Motif applications that are *Motif Style Guide* compliant must support methods for interaction that do not depend on a mouse. For the purposes of keyboard navigation, components of the application are organized into groups called *fields*. To move the input focus forward from one field to the next, press `Tab`. To move the input focus in the reverse direction, press `Shift Tab`. Use the arrow keys to move the input focus among components within a field.

Menu navigation in Motif applications uses special key bindings, which are summarized in the following list:

- `F10` traverses to and from the MenuBar.
- `Menu` or `Shift F10` posts and cancels Popup menus in areas that have Popup menus.
- `Space` and `Select` activate Option Menus.
- The arrow keys traverse Menu panes.
- `Cancel` backs out of a Menu hierarchy.

In addition, the `Tab` key may be modified to move forward like an arrow key until a tab group (one or more components that can be navigated) boundary is reached. At the boundary, the `Tab` key then moves forward or backward to the next tab group, depending on the direction it was navigating. This `Tab` key modification is possible only if the display is set up to be able to

Applications may also use page navigation keys to scroll a region one page at a time. Appendix A contains the default Motif key bindings.

Using Drag and Drop

The *drag and drop* facility allows you to pick up an object from one part of your display and move it to a new location. The object you pick up is called the *drag source*. The location where you drop it is called the *drop site*.

With drag and drop, you can do the following tasks:

- Move text or other information between windows (cut and paste)
- Invoke actions
- Obtain information about drop sites

For example, you can print the contents of a file by dragging an icon representing the file and dropping it on an icon representing a printer.

The source and destination of the drop can both be in the same application, or they can be in different applications. Text components, Labels, Menus, and Lists are drag sources in all Motif applications. Text components are also automatically drop sites. Other drag sources and drop sites are defined by the specific applications.

Additionally, the contents of Labels and Scales may be draggable.

Starting a Drag

To start a drag:

1. Decide what you want to drag, and select it as follows:
   1. For Text components, select the text to be dragged.
   2. For Lists, if you want to drag several items, select them. A single item can be dragged without first being selected.
   3. Other items usually can be dragged without being selected first; however, applications may require you to select single items before dragging them.
2. Move the pointer to the item that you want to drag. This item is the drag source.
3. Press and hold down Button 2.
4. Specify the operation that you want to happen when the drop is made by pressing modifier keys on your keyboard (see Table 3−1).

Table 1 Drag Operations

Key	*Action*
`Shift`	Move; information is moved from the initiator to the receiver
`Ctrl`	Copy; information is copied from the initiator to the receiver
`Shift Ctrl`	Link; information is linked from the initiator to the receiver

If a modifier key is not pressed, a move operation is generally assumed, but the actual operation chosen depends on the source and destination. The applications that are controlling the source and destination may have specified that only some of the possible operations are allowed. A move operation deletes the information from the original location after it has been copied to the new location.

Note that which mouse buttons are to be used for transfer depends on how these buttons have been defined for your mouse. In some cases, Button 1 is used to both select and transfer the data and Button 2 is used to adjust or extend the selection and for primary transfer, and in other cases, Button 1 is used to select the text, and Button 2 is used to transfer it. This extension of the capabilities of Button 1 and Button 2 is possible for Text, Lists, and Containers.

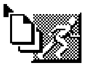
If the system selects an operation that you do not want, press the appropriate modifier keys to change to a new operation at any time during the drag.

A *drag icon* representing the drag source appears instead of the normal pointer. The drag icon used depends on the drag source. The drag icon consists of three parts:

- Source Icon: The source icon is a graphical representation of the object being dragged.
- State Icon: The state icon represents the relation of the drag icon to what is under it. It may change depending on whether the drag icon is over a valid drop site, over an invalid drop site, or not over a drop site.
- Operation Icon: The operation icon represents the operation to be performed when the drop is made (copy, move, or link). If you have requested an operation that either the initiator or receiver does not allow, this icon will be blank.

Figure 3−30 shows a drag icon. The source icon is the running figure, the state icon is the arrow in the upper left corner, and the operation icon is the box just below the arrow.

Figure 1 A Drag Icon

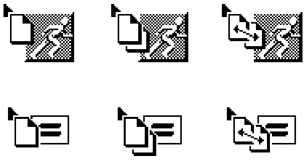


Motif defines a default source icon for textual information, which is used if the drag source is a List, a Label, or a Text component. If the source is a graphics component, Motif will attempt to use the graphic image as the source icon. If the source is of a different type and the application has not provided a source icon for that type of data, the running figure is used as the source icon.

The state and operation icons are optional in a drag icon. Motif provides default operation icons for move, copy, and link operations. The default state icon for all three states is an arrow in the upper left corner of the drag icon.

Figure 3−31 shows the default drag icons for graphics and textual information for move, copy, and link operations.

Figure 2 Default Drag Icons



An application can specify different icons for each of the operations and states, as well as arrange the icons in a different way than the one shown in the illustration. You can also specify different icons by setting resources (see Chapter 7).

During the Drag

Without releasing the mouse button, move the mouse around. The drag icon representing the source moves around the screen instead of the normal pointer shape. A drag can also be operated from the keyboard, usually with one of the arrow keys.

Every drag icon has a hot spot. The whole icon is considered in or out of a drop site when the hot spot is in or out of the drop site. If a drag icon displays a state icon, the hot spot is in it. Otherwise, the hot spot is in the source icon. The hot spot in the default state icon is located at the point of the arrow.

As the drag icon moves around the screen, it may change color or shape to indicate whether the drag icon is over a valid drop site, an invalid drop site, or no drop site. These changes are called *drag−over effects*. The default behavior is that there is no apparent change during the move. However, an application may provide other drag−over effects or you can specify your own with resources.

The operation icon indicates which operation will occur when the drop is made. You can change modifier keys to switch the operation at any time during a drag. The operation icon will change to show the new operation. If you have not specified an operation using the modifier keys, the default operation depends on what drop site the drag icon is over.

The appearance of a drop site may also change as the drag icon moves in and out of it. These changes are called *drag−under effects*. The default drag−under effect for a valid drop site is to display a border around the drop site. Another common drag−under effect is called *pause drag*. This is where the drag icon is held over an area (without releasing) with some action resulting in the area underneath. A scrolled window, for example, might scroll if the drag icon is paused on the scroll arrows, or at the border of the window. Other drag−under effects depend on the application that is controlling the drop site.

Dropping on a Destination

To drop a dragged object on a destination:

1. Move the drag icon over a drop site.
2. Release Button 2 while still pressing any modifier keys that you are using to determine the operation. The drag operation can also be completed from the keyboard, usually by pressing `osfActivate` or `<Enter>.`

If the drop site is valid for that drag source, then the drag icon shows the completed drop by melting into the destination. If the drop site is not valid for that drag source, then the drag icon snaps back to the point from where the drag was initiated.

If the drag source is a Label, a List, or a Text component, the data being dragged is textual information and can be dropped on any Text destination. Other combinations of source type and destination type are defined by the application.

If the destination holds data or information, the dragged information generally is inserted at the pointer location. If the destination represents an application or device, generally a drop starts some action. What the action is depends on the application. For instance, if you drag an icon representing a file to an icon representing a printer, then the action will be that the contents of that file are printed on the printer indicated by the icon.

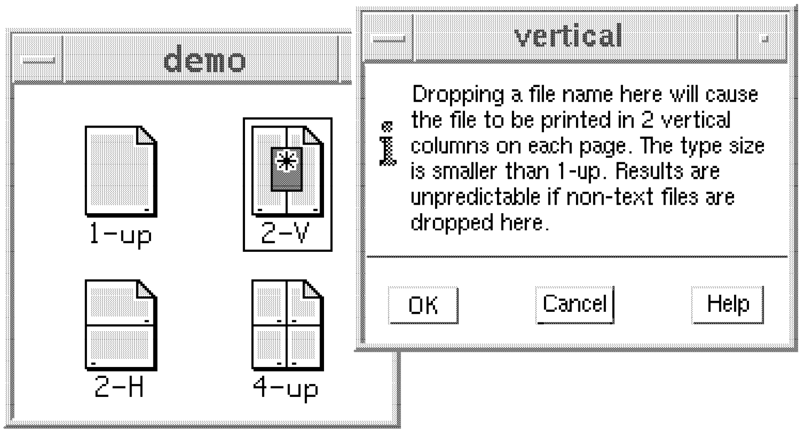
Getting Help

You can request help about a particular drop site if the application has provided help information. The help information usually explains the sources that are expected by a drop site and the actions that will occur as a result of a drop.

To request help:
1.Move the drag icon over a drop site.
2.Press `Help` or `F1`.

Figure 3−32 shows an example of Help information.

Figure 3 Help Information



Dropping a file name here will cause the file to be printed in 2 vertical columns on each page. The type size is smaller than 1-up. Results are unpredictable if non-text files are dropped here.

Canceling a Drag

At any time during a drag, you can cancel the drag operation. To cancel a drag in progress, press `Cancel`. The drag icon is replaced by the normal pointer shape and any special drag−over and drag−under effects go away.