Control Panel Guidelines

This chapter describes guidelines you must follow if you are designing control panels. Specifically, the following are discussed:

- window design
- fonts, menus, and icons
- settings behavior
- control panel access
- assigning keyboard equivalents
- on-screen help and assistance

Control panels are appropriate for allowing users to configure:

- settings with system-wide effects (for example, the Date & Time control panel)
- settings that have no visible effect (for example, the File Sharing control panel)
- hardware (for example, the Monitors & Sound control panel)

Control panels are not appropriate for allowing users to configure specific applications or utilities, or tasks that would require multiple layered dialog boxes or custom menu items.

Windows

Control panels should follow the guidelines established in "Utility Windows" (page 98) and *Macintosh Human Interface Guidelines*. In addition you must observe the following:

- Each control panel has one window—a modeless dialog box —that opens when the control panel is launched. The preferred maximum size for this window is 400 x 300 pixels (width x height), but it may be larger if necessary. The absolute maximum size for the control panel window is 492 x 340.
- The window must have a close box. The control panel quits whenever the window closes.
- You should put as many of the control panel's settings as possible in the window. Infrequently used settings or minor features may be relegated to

modal or movable modal dialogs boxes. These dialog boxes should be accessible by controls (usually pushbuttons) in the control panel window. Only if the window cannot accommodate such controls should menu commands be used instead.

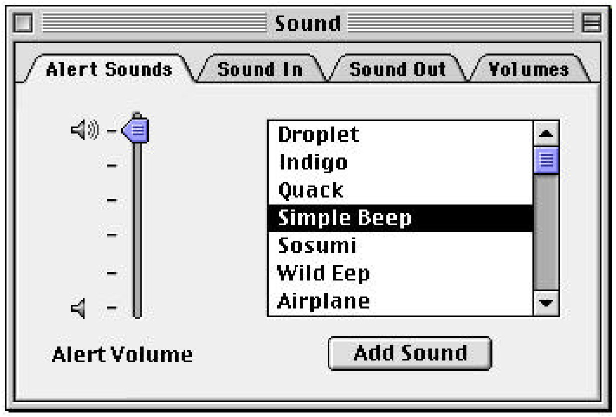
- If the control panel contains a large number of frequently used settings, you can make the window a multi-pane modeless dialog box. In this case, any secondary dialog boxes opened from the control panel window should be modal.

Multi-Pane Windows

If you use a multi-pane modeless dialog box for the control panel window, there are several design options for navigating between individual panels.

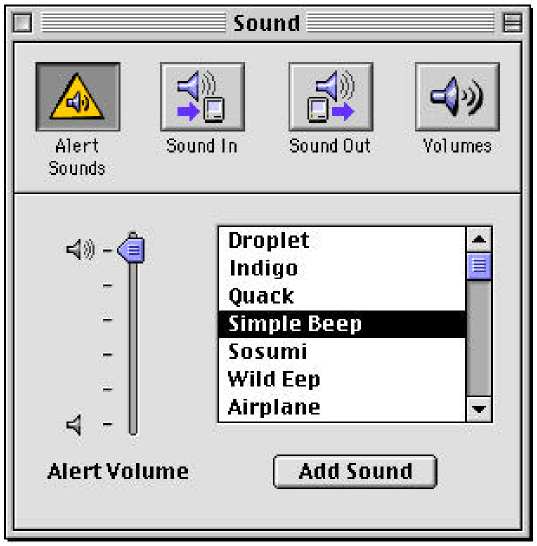
You should use tab controls if the number of panes is fixed and the control panel has adequate horizontal space to accommodate all the tabs in a single horizontal row. Tab controls can simultaneously and automatically display all choices to the user; they can also display icons in addition to (but not instead of) the text label. However, tab controls use considerable horizontal space and are not as extensible as a scrolling list or pop-up menu. The size of any icon in a tab control is also limited. Figure 6-1 shows the use of tab controls in a multi-pane control panel.

Figure 6-1 Using tab controls to navigate a multi-pane control panel



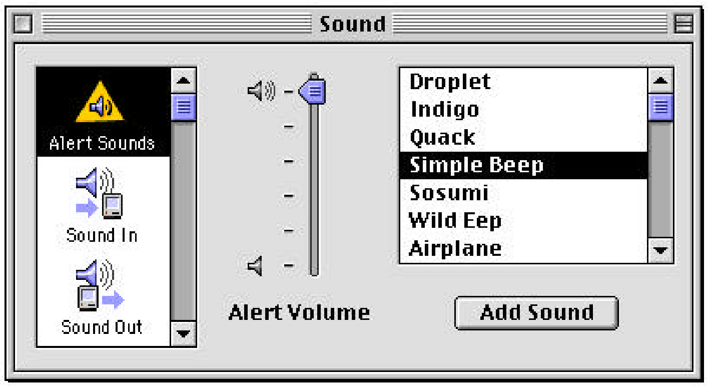
You can use a group of push buttons if the number of panes is fixed and the control panel has adequate space to accommodate all the buttons and labels in a single horizontal or vertical row. Like tab controls, push buttons simultaneously and automatically display all choices to the user. They can display both text and large or small icons; a button's text label may wrap to a second line, so a group of push buttons may require less horizontal space than tabs. However,push buttons still use considerable horizontal and vertical space. Figure 6-2 shows the use of push buttons in a multi-pane control panel.

Figure 6-2 Using push buttons to navigate a multi-pane control panel



You can use a scrolling list of icons (horizontal or vertical) if the number of panes is indefinite (extensible) and the control panel has adequate space to accommodate the list. A scrolling icon list can display both text and large or small icons. However, they use considerable horizontal and vertical space, and they require a user action (clicking the scroll bar) to display all choices. Also, the user is unlikely to be able to see all choices simultaneously. Figure 6-3 shows a vertical scrolling list of icons used in a multi-pane control panel.

Figure 6-3 Using a scrolling list of icons to navigate a multi-pane control panel



Droplet
Indigo
Quack
Simple Beep
Sosumi
Wild Eep
Airplane

If no other method is suitable, you can use a pop-up menu. Pop-up menus can display a fixed or indefinite number of control panel panes. Pop-up menus use minimal space and display all choices to the user simultaneously. However, they can display only small icons and they require a user action (clicking on the pop-up menu) to display the choices. In the unlikely—and undesirable—event that a pop-up menu has too many items to display simultaneously (that is, the user must scroll to see them all), you should consider rearranging your control panel or, at worst, dividing the settings into two control panels. Figure 6-4 shows a pop-up menu used in a multi-pane control panel.

Figure 6-4 Using a pop-up menu to navigate a multi-pane control panel

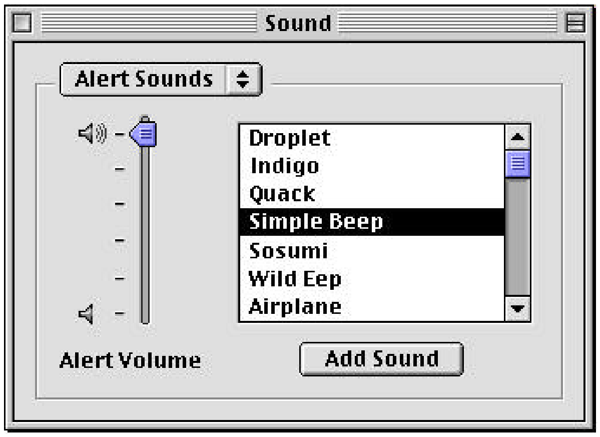


Table 6-1 summarizes the four different ways of navigating multi-pane windows.

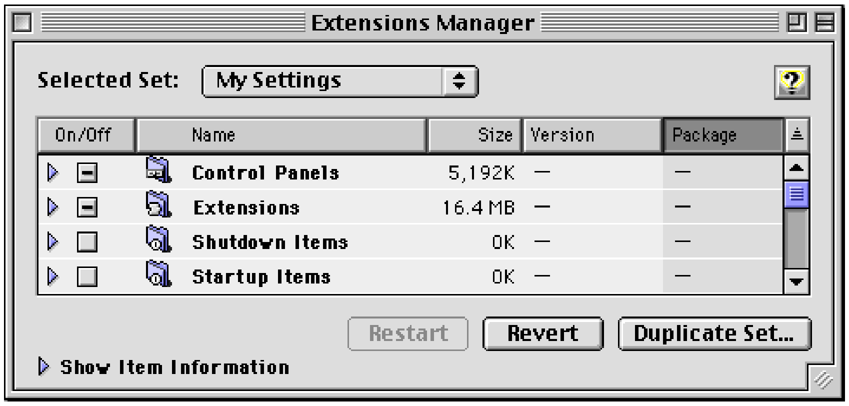
Table 6-1 Multi-pane window navigation methods

Control type	Advantages	Disadvantages
Tab controls	Displays all choices simultaneously; can display icons in addition to text	Uses considerable horizontal space; limited extensibility
Buttons	Displays all choices simultaneously; can display both icons and text; may require less horizontal space than tab controls	Uses considerable horizontal and vertical space; limited extensibility
Scrolling list of icons	Unlimited extensibility; can display both icons and text	Uses considerable horizontal and vertical space; requires a user action to see all choices; may not display all choices simultaneously
Pop-up menu	Displays all choices simultaneously; unlimited extensibility; uses minimal space	Requires a user action to see all choices; can display only small icons

Expanding and Contracting Windows

If a control panel contains a significant number of settings that users are not likely to adjust frequently, then you can let the user to show or hide those settings by expanding or contracting the window. The preferred mechanism for this is a labeled disclosure triangle that expands the window downward or contracts it upward. The triangle's label should indicate the nature of the settings (for example, "Server settings"). Figure 6-5 shows a control panel with the disclosure triangle in the closed state.

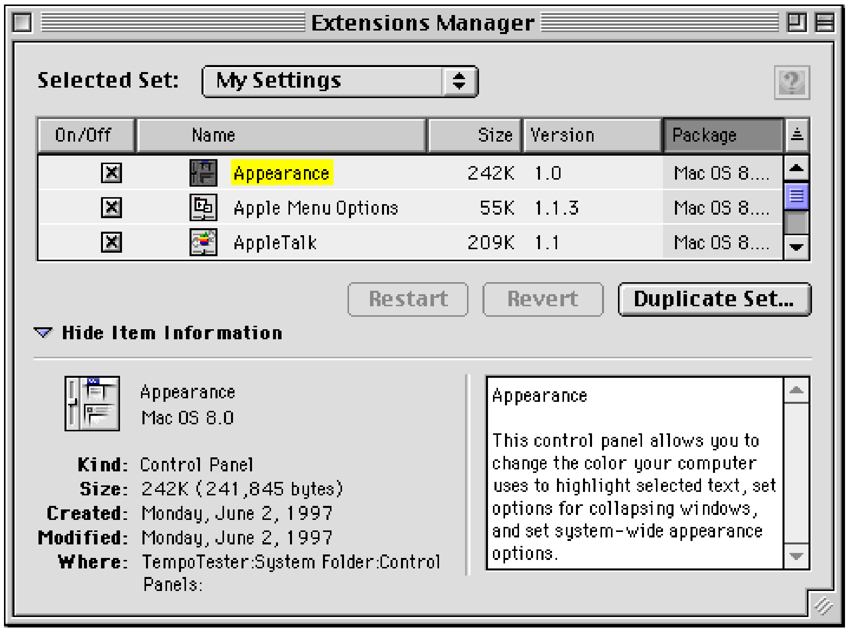
Figure 6-5 A control panel with the disclosure triangle in the closed state



On/Off	Name	Size	Version	Package
▷ ⊟	Control Panels	5,192K	—	—
▷ ⊟	Extensions	16.4 MB	—	—
▷ ☐	Shutdown Items	0K	—	—
▷ ☐	Startup Items	0K	—	—

Figure 6-6 shows a control panel with the disclosure triangle in the open state.

Figure 6-6 A control panel with the disclosure triangle in the open state



Selected Set: My Settings

On/Off	Name	Size	Version	Package
☒	Appearance	242K	1.0	Mac OS 8....
☒	Apple Menu Options	55K	1.1.3	Mac OS 8....
☒	AppleTalk	209K	1.1	Mac OS 8....

Appearance
Mac OS 8.0

Kind:	Control Panel
Size:	242K (241,845 bytes)
Created:	Monday, June 2, 1997
Modified:	Monday, June 2, 1997
Where:	TempoTester:System Folder:Control Panels:

Appearance

This control panel allows you to change the color your computer uses to highlight selected text, set options for collapsing windows, and set system-wide appearance options.

See “Disclosure Triangles” (page 37) for more information about disclosure triangles.

If, due to space or layout problems, you cannot have your control panel open downwards, you can use a pushbutton to expand or contract the window’s right edge.

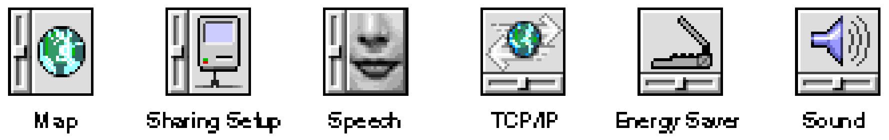
Note
When a control panel is localized, it may be the left edge that expands and contracts. u

As with the disclosure triangle, the pushbutton label should indicate the nature of the settings. In addition, it should indicate “Show” or “Hide” as appropriate (for example, “Show Server Settings” or “Hide Server Settings”).

Icons

Desktop icons for control panels should be based on the standard control panel icon, with the control slider on either the left side or the bottom. Figure 6-7 shows some examples of control panel icons.

Figure 6-7 Desktop icons based on the standard control panel icon



Fonts

Control panels should follow the guidelines established in “Font Guidelines” (page 66).

Menus

Because Mac OS 8 control panels are applications (file type `'appc'`), they generate their own menus. Control panels must follow the guidelines established in “Menu Guidelines” (page 89) as well as those in this section.

IMPORTANT

Because older control panels did not generate their own menus, some users are likely to assume that Mac OS 8 control panels also do not. Custom menus and menu items therefore should be used sparingly, and only minor or infrequently used features should be accessed exclusively through menu items. s

Standard Menus

Each of the standard menus should contain specific items.

Control panels should have an About *Control Panel Name* ... command as the first item in the Apple menu. This command opens the control panel's About box. See the "The About Box" (page 122) for more information.

The File menu should always include a Close Window (Command-W) command. Closing the control panel window quits the application. The final command of the File menu should always be Quit (Command-Q). This command is always preceded by a separator line, even if the only other command is Close Window.

If the control panel requires an explicit save action and it saves settings globally, then the File menu should include Save Settings (Command-S) and Revert to Saved Settings commands. For more information, see "Settings" (page 117).

The Edit menu should include an Undo (Command-Z) command that operates on as many controls as possible, including any Save and Revert buttons. The Undo command is followed by a separator line.

If the control panel includes any edit text fields, then the Edit menu should also include Cut (Command-X), Copy (Command-C), Paste (Command-V), and Clear commands. The Edit menu should also include Select All (Command-A) if appropriate.

In the unlikely event that a control panel has enough preference settings to justify placing them in their own dialog, the preferred mechanism for accessing them is through an Options button in the control panel window. However, if the window cannot accommodate such a button, then a Preferences command may be used instead. This command is always the last item in the Edit menu, and it is always preceded by a separator line.

Contextual Menus

Control panels should support the contextual menu feature introduced in Mac OS 8. The contextual menu appears whenever the user Control-clicks an object in the control panel. Only those commands that apply to the object under the pointer appear in the contextual menu. The first item in the contextual menu is always Help. For additional information about contextual menus, see "Contextual Menus" (page 91).

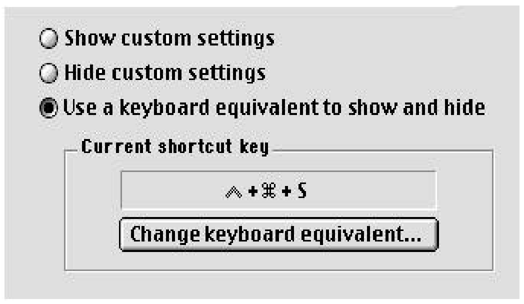
Keyboard Equivalents Support

You should provide keyboard equivalents (shortcut keys) for frequently used controls. If a dialog control does not normally support a keyboard equivalent (for example, pushbuttons or radio buttons) you can duplicate the control as a menu command. For example, a control panel that has an Edit Connection button could also have a Connection command (say Command-E) in the Edit menu. However, control panel menu commands should be used in addition to—not instead of—dialog controls.

To help users avoid or eliminate conflicts, keyboard equivalents may be user definable. You should maintain the following interaction sequence for defining keyboard equivalents whenever possible:

Display the current keyboard equivalent and a button for redefining it. Figure 6-8 shows an example.

Figure 6-8 The current keyboard equivalent and a button for redefining it



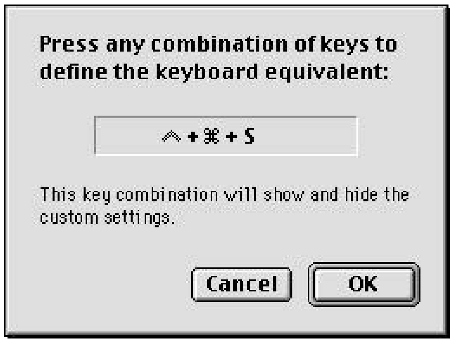
Show custom settings [ ]
Hide custom settings [ ]
Use a keyboard equivalent to show and hide [✓]

Current shortcut key

⌃ + ⌘ + S

Clicking the button opens a dialog with instructions and feedback for changing the keyboard equivalent. Figure 6-9 gives an example.

Figure 6-9 Changing the keyboard equivalent



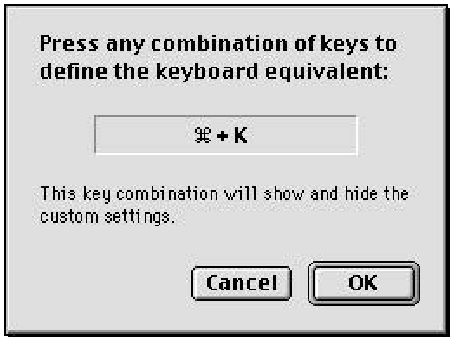
Press any combination of keys to define the keyboard equivalent:

⌃ + ⌘ + S

This key combination will show and hide the custom settings.

As soon as the user finishes typing a new key combination, the feedback text field displays the new combination as shown in Figure 6-10.

Figure 6-10 Displaying the new keyboard equivalent combination



Press any combination of keys to define the keyboard equivalent:

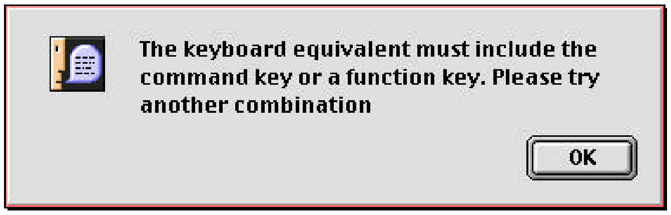
⌘ + K

This key combination will show and hide the custom settings.

If the new key combination is invalid, the feedback text field remains unchanged and an appropriate alert appears. Alerts should indicate the following cases:

- The keyboard equivalent selected did not contain a command key or a function key.
- The keyboard equivalent selected is already in use by another item. The alert should indicate which item is using the shortcut.

Figure 6-11 An Alert for an invalid keyboard equivalent combination

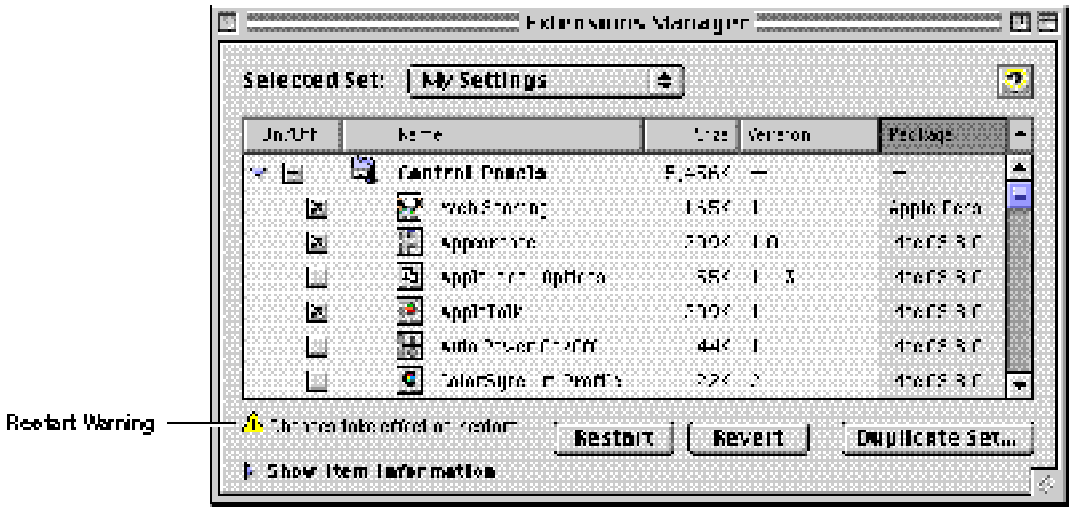

The keyboard equivalent must include the command key or a function key. Please try another combination

See the "Menus" chapter of *Macintosh Human Interface Guidelines* for additional guidelines and a list of reserved keyboard equivalents.

Settings

In most cases, changes made to control panel settings should take effect immediately, with no additional user action required (for example, without having to close a window or quit the control panel). If a change is potentially disruptive, however, you should postpone the effect to avoid possible problems. For example, the Network control panel does not actually save the settings until you indicate (by selecting Save Settings) that you want to do so. In such cases, the effect may be postponed until the control panel is closed or, if necessary, until the system is restarted. If the system must be restarted to effect the new settings, the control panel's window should display a brief explanation (accompanied by a small alert icon) of this behavior when changes are made. Figure 6-12 shows this message in the Extensions Manager control panel.

Figure 6-12 Alert message in the Extensions Manager control panel



As a rule, control panel settings should be saved automatically when the control panel quits. Only if the settings are complicated or technical should saving them require an explicit user action. In such instances, requiring an explicit save allows the user to retreat from unintended changes.

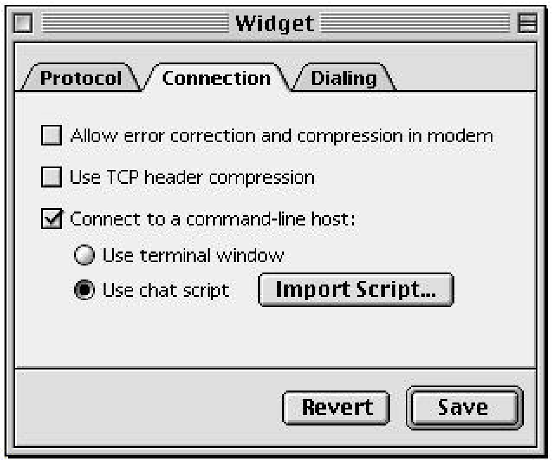
The user should save and restore settings in logical sections rather than globally if possible. For example, in a control panel that configures several different devices, the user might be able to save and restore settings for each device rather than for all devices simultaneously.

Global save and restore commands may be used instead if it is impossible to separate a control panel's settings into logical groups. In this case, the control panel window should include Save and Revert buttons, and the File menu should include corresponding Save Settings and Revert to Saved Settings commands. A confirmation dialog box should be displayed if the user quits the control panel with unsaved changes.

When designing the control panel, you must make it clear to the user which settings are affected by the save and restore functions. The user must know whether settings are saved globally or in sections; if they are saved in sections, then the user must be able to discern the boundaries between sections. The simplest solution is to design the control panel so that each section of savable settings is edited in a separate dialog with its own OK and Cancel buttons.

Saving and restoring settings in modal multi-pane dialogs is especially problematic and should be avoided if possible. If you must use multi-pane windows in such cases, then you must make sure the user can tell whether the save and revert commands apply to a single pane or to the entire dialog. Figure 6-13 shows an example of saving globally (the Save and Revert buttons are separate from the individual panes).

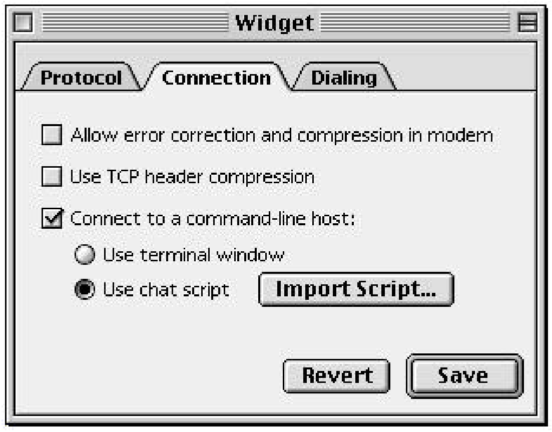
Figure 6-13 Saving and restoring globally in a multi-pane control panel



- [ ] Allow error correction and compression in modem
- [ ] Use TCP header compression
- [x] Connect to a command-line host:
  - ○ Use terminal window
  - ● Use chat script

Figure 6-14 shows an example of saving in sections (each pane has its own Save and Revert buttons).

Figure 6-14 Saving and restoring sections in a multi-pane control panel



☐ Allow error correction and compression in modem
☐ Use TCP header compression
☑ Connect to a command-line host:
○ Use terminal window
● Use chat script

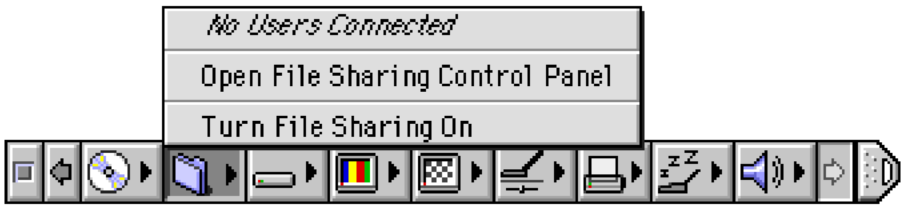
Miscellaneous Guidelines

This section discusses other guidelines you should keep in mind when creating control panels.

Persistent Access

A control panel may generate some sort of screen representation that persists even when the control panel itself is closed. Such screen representations should always include a mechanism for opening the control panel. If a control panel has an associated Control Strip module, that module should include a command that opens the control panel. Figure 6-15 shows control panel access from a Control Strip module.

Figure 6-15 Access to a control panel from its associated Control Strip module



Assistance

Control panels should be “assistance savvy,” which means that they should

- display the Help button (that is, the button with the question mark icon) for context-sensitive help
- provide suitable entries in the AppleGuide databases for “Guide Me” and “Do It For Me” assistance

Most control panels will be included in the general AppleGuide database, but a complex control panel may require its own database. For more information regarding making control panels assistance-savvy, see **<Lisa’s doc references the internal document Human Interface Guidelines for Control Panels v0.3 here. What’s an appropriate public doc to reference instead?>**.

Control panels should also provide help balloons. See the “Language” chapter of *Macintosh Human Interface Guidelines* for general guidelines on Balloon Help behavior and content.

Scripting

Control panels should be scriptable using AppleScript.

Extensions Manager

A brief description of the control panel’s purpose should be included as a CCI™ (Conflict Catcher Information™) resource string for use by the Extensions Manager.

The About Box

The About box for a control panel should include the following items:

- the control panel’s logo, name, and version number
- a brief description of the control panel’s purpose
- credits
- copyright information
- an OK button that dismisses the About box