NOTE: This document excerpts Silicon Graphics' User Interface Style Guidelines, in order to provide an overview of a concerted set of style enhancements to baseline OSF/Motif Style, as implemented in release 1.1 from the Open Software Foundation.

SGI Style Guidelines

Chapter 2

Component Specifications

A standard set of user interface components are specified, in order to promote a familiar set of interaction techniques across all applications within the SGI User Environment. While many developers will use available toolkits which provide these components readymade, off the shelf, the following specification provides sufficient guidance for programmers who wish to implement their own style–conforming components.

Table of Components [excerpted]

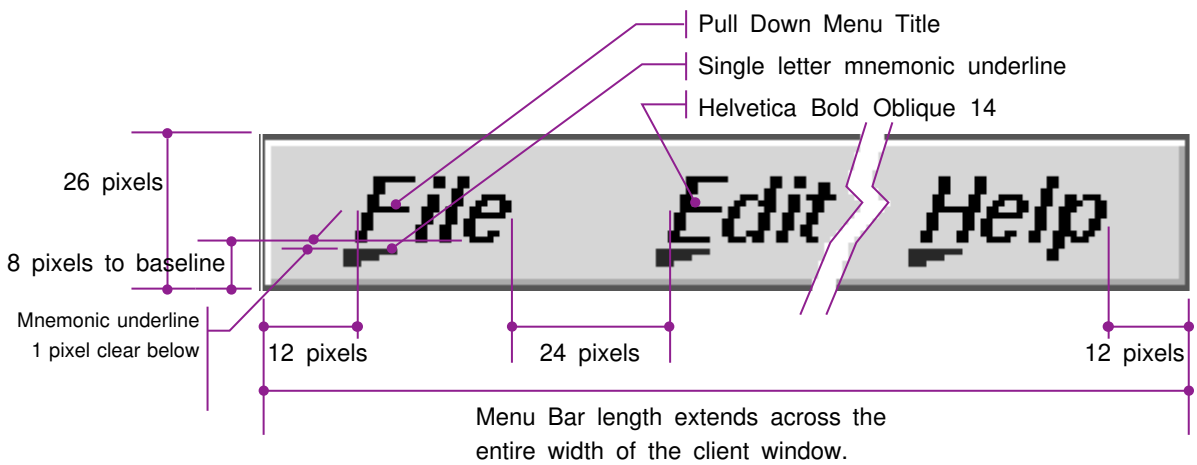
Menu Bar	1
Pull Down Menu	3
Popup Menu	6
Option Menu	9
Palette Menu	11
Push Button	14
Indicator Lamp	20
Toggle Button	22
Check Button	24
Radio Button	29
Field Title and Label	34
Text Field	36
Dial	39
Slider	42
Scroll Bar	44
Arrow Button	48
Scrolling List	49
Dialog Box	52

Menu Bar

Use a Menu Bar to provide convenient access to an application's primary operations within a given window. The Menu Bar consists of a horizontal bar fixed across the very top edge of the window, just below the Window Border's Title Bar. The Menu Bar presents the title of each of the application's available Pull Down Menus. As the user presses the Select Mouse Button (LeftMouse) or the Popup Mouse Button (RightMouse) and drags the pointer across these titles, each corresponding Pull Down Menu pops up in place. This action supports quick browsing sideways, from menu to menu, as well as the descent downwards, to select any one of the menu items desired.

Do not embed fixtures that directly execute commands, such as Push Buttons, within the Menu Bar. The use of such fixtures inhibits menu browsing.

Rendering Specification



Menu Bar

Presentation States

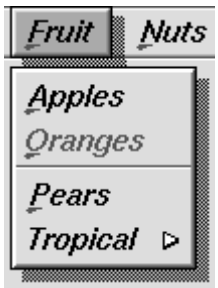
Neutral

Located

Loc+Sel

Selected

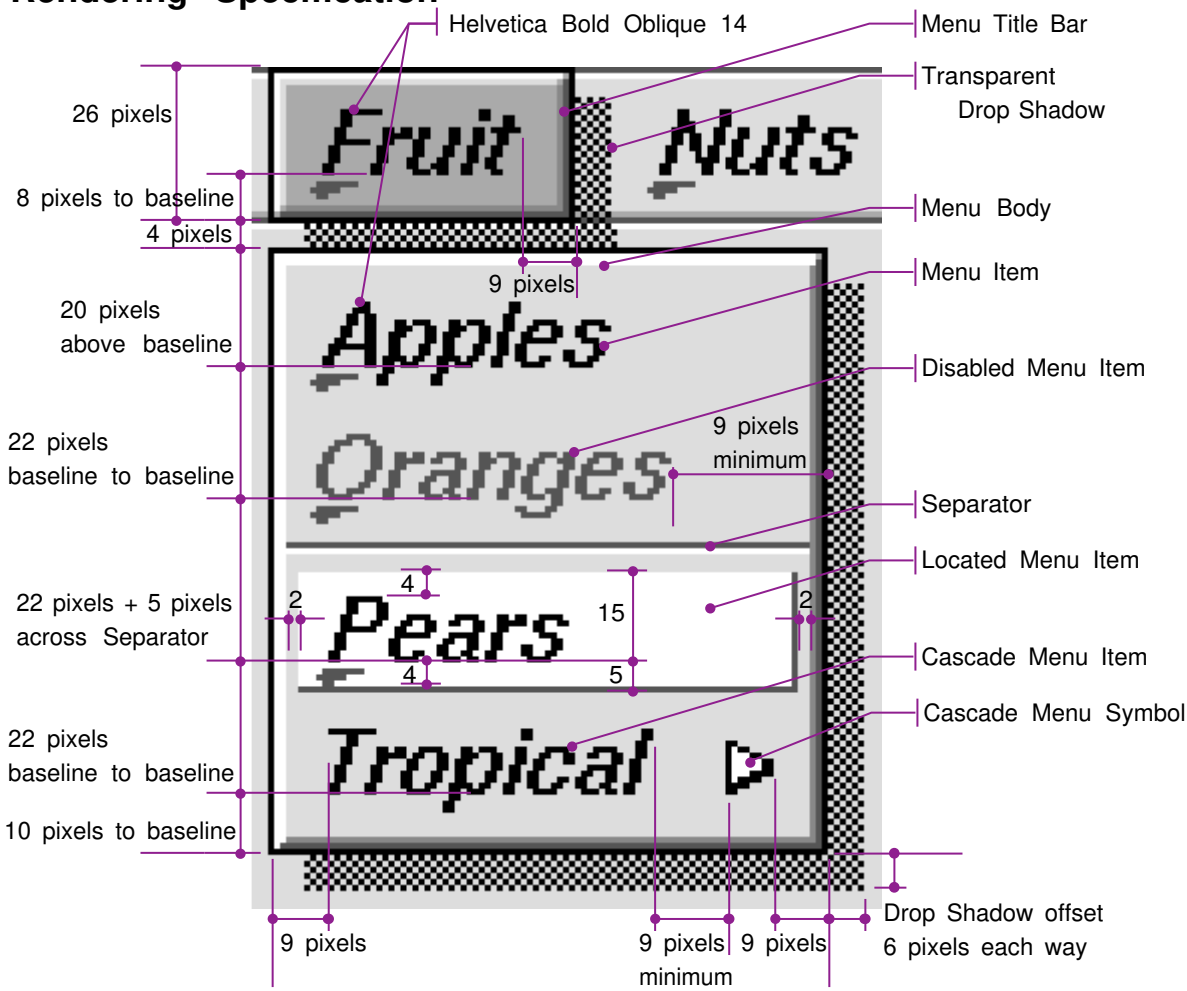
Disabled

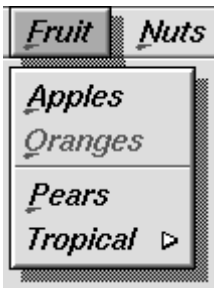


Pull Down Menu

Use Pull Down Menus with Pull Down Menu titles in the Menu Bar, to provide quick access to application commands. See "Menu Bar" for a fuller description.

Rendering Specification



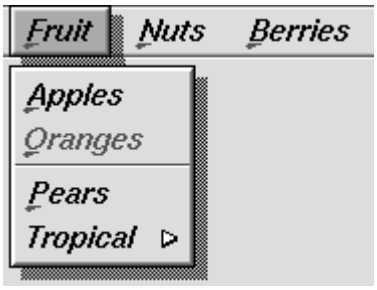


Pull Down Menu

Presentation States

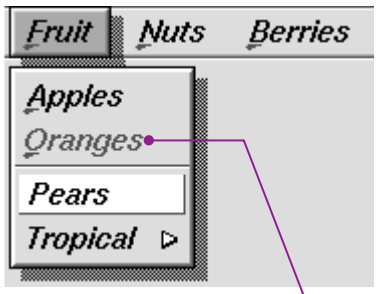
Top–level Pull Down Menu

Neutral



Located Not Applicable

Loc+Sel

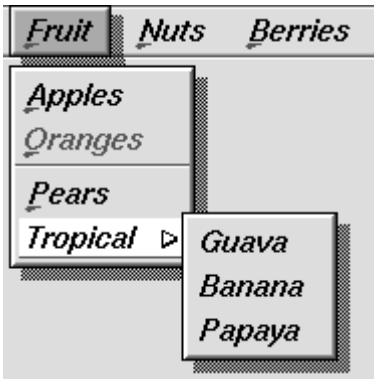


Selected Not Applicable

Disabled Grayed out per Menu Item

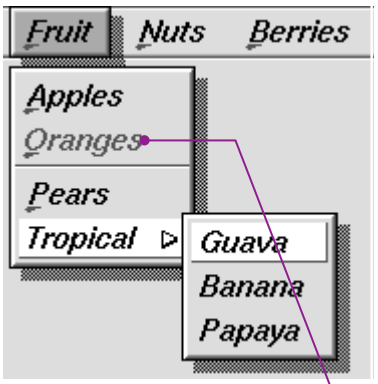
Cascaded Submenu

Neutral



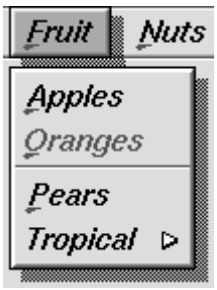
Located Not Applicable

Loc+Sel



Selected Not Applicable

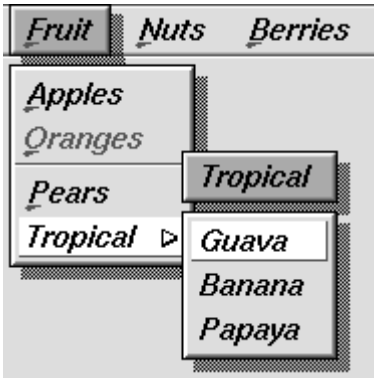
Disabled Grayed out per Menu Item



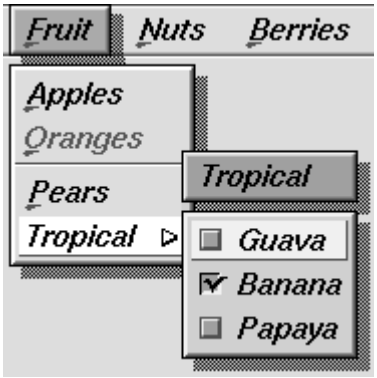
Pull Down Menu

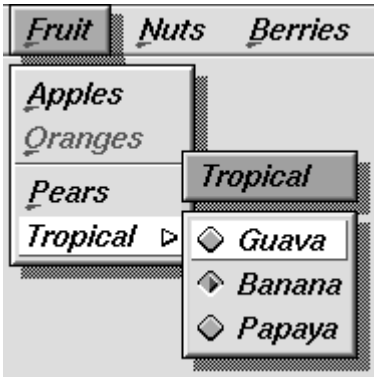
Variations

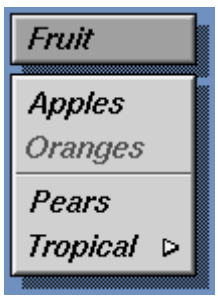
A Menu Title Bar may be added to the Cascading Submenu for further emphasis.



Check Button Items and Radio Button Items may be used within Pull Down Menus.



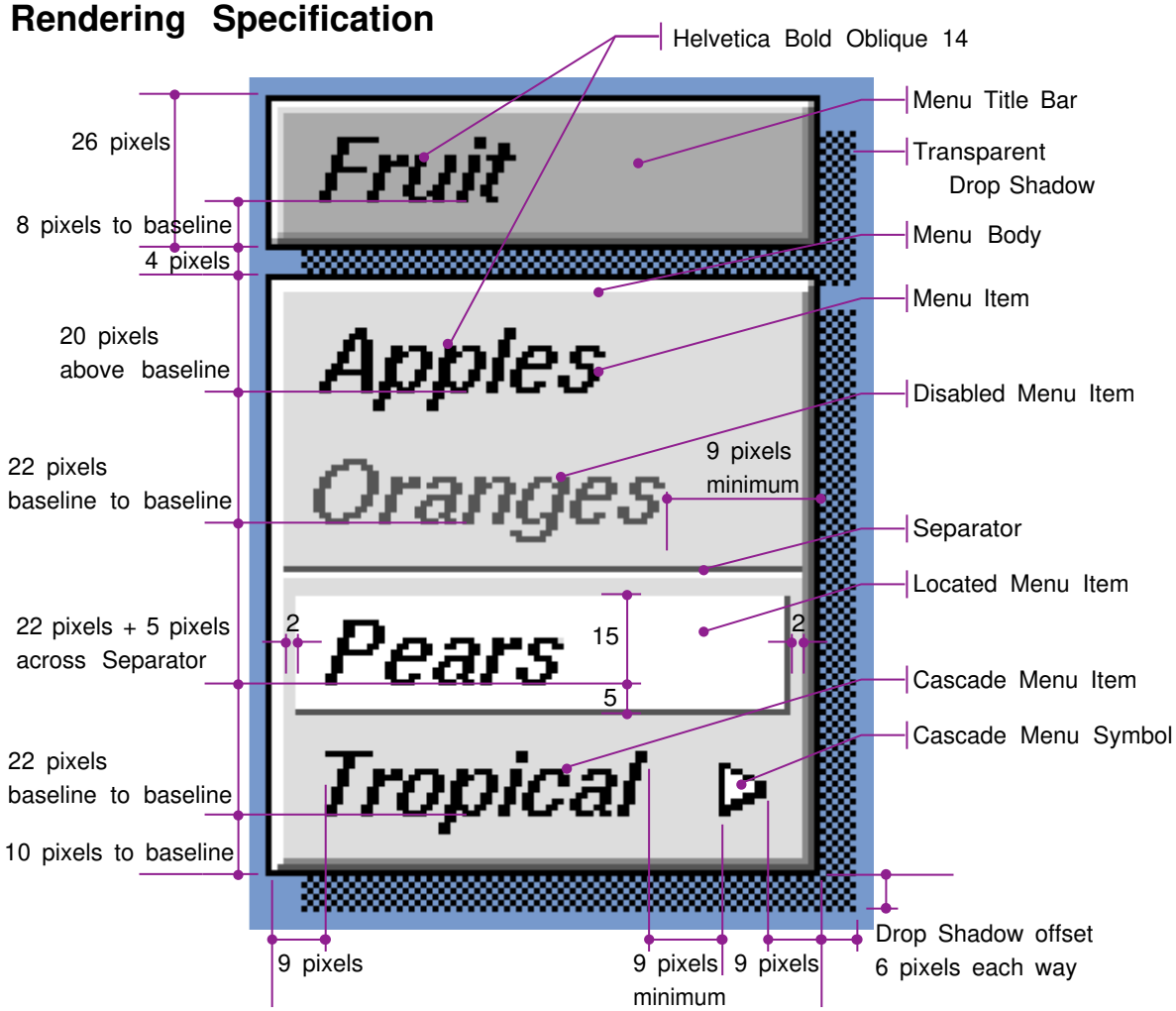




Popup Menu

Use Pull Down Menus with Pull Down Menu titles in the Menu Bar, to provide quick access to application commands. See "Menu Bar" for a fuller description.

Rendering Specification

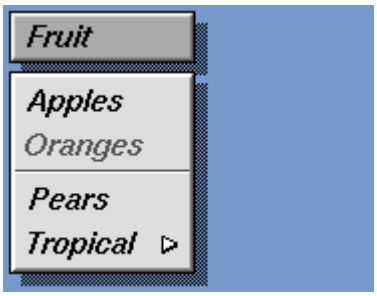


Popup Menu

Presentation States

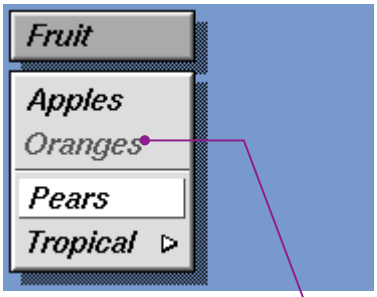
Top–level Popup Menu

Neutral



Located Not Applicable

Loc+Sel

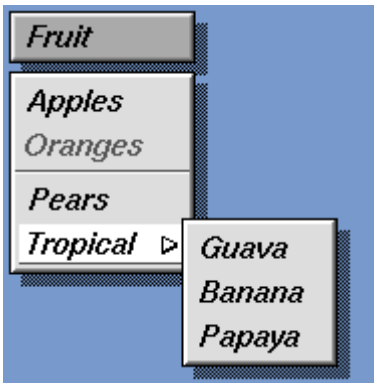


Selected Not Applicable

Disabled Grayed out per Menu Item

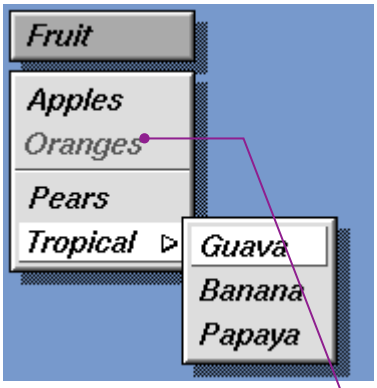
Cascaded Submenu

Neutral



Located Not Applicable

Loc+Sel



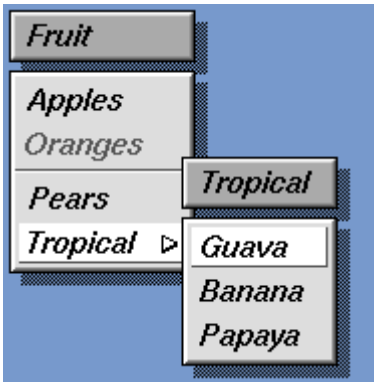
Selected Not Applicable

Disabled Grayed out per Menu Item

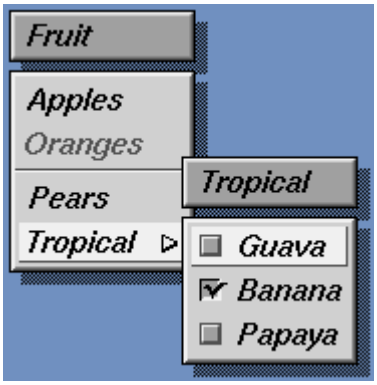
Popup Menu

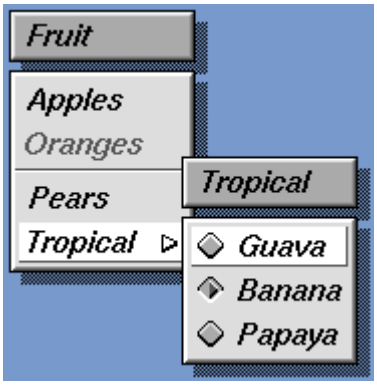
Variations

A Menu Title Bar may be added to the Cascading Submenu for further emphasis.



Check Button Items and Radio Button Items may be used within Popup Menus.

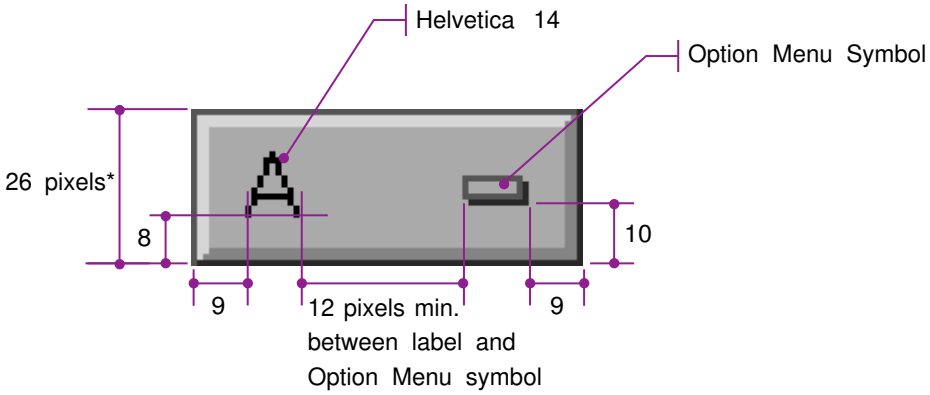




Option Menu

Use an Option Menu to establish a field where users can choose one setting from a limited list of options. The field is represented by a fixed Option Menu Button, with the text label presenting the value of its current effective setting. When the user presses down either the Select Mouse Button (LeftMouse) or the Popup Mouse Button (RightMouse) over the Option Menu Button, a Popup Option Menu appears. The user can make a new selection in the same way as with a standard Popup Menu. When the selection is completed, the Popup Option Menu disappears, and the selected menu item appears on the face of the Option Menu Button.

Rendering Specification



* Height to match Menu Bar

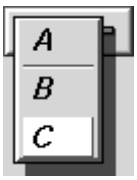
Option Menu

Presentation States

Neutral

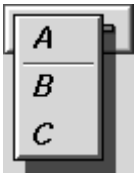
Located

Loc+Sel



After a new item has been selected from the popup Option Menu, its label is displayed on the Option Menu Button.

Selected

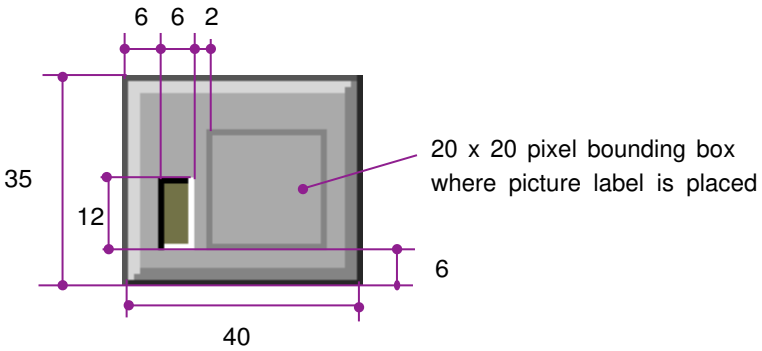


Disabled

Palette Menu

The Palette Menu presents a set of Palette Buttons as a fixed menu. Typically, you use it to present a set of "tools" or option settings to the user, in conjunction with a larger working area, or "canvas". Palette Buttons usually indicate their function with graphical labels instead of text labels, and frequently use an imbedded indicator lamp to indicate the current tool selection. Palette Buttons operate as Toggle Buttons; see the description under *"Toggle Button"* for more details.

Rendering Specification



Palette Menu

Presentation States

State Transistions

[Transistions between the basic presentation states, and any other intermediate states, are outlined against user input, to help describe the behavior of the component.]

Palette Menu

Variations

The graphical label displayed within the Palette Button varies with usage. Use any combinations of Indicator Lamps, Graphical Labels, and/or Text Labels to convey the meaning of the tool.

A sample picture label

An important technique to reduce the busy, cluttered appearance within a particularly large, compact, or complex Palette Menu involves supressing the explicit rendering of each Palette Button background while shown in the neutral state. Look under "*Toggle Button*" for a complete description of this treatment.

Neutral

Neutral

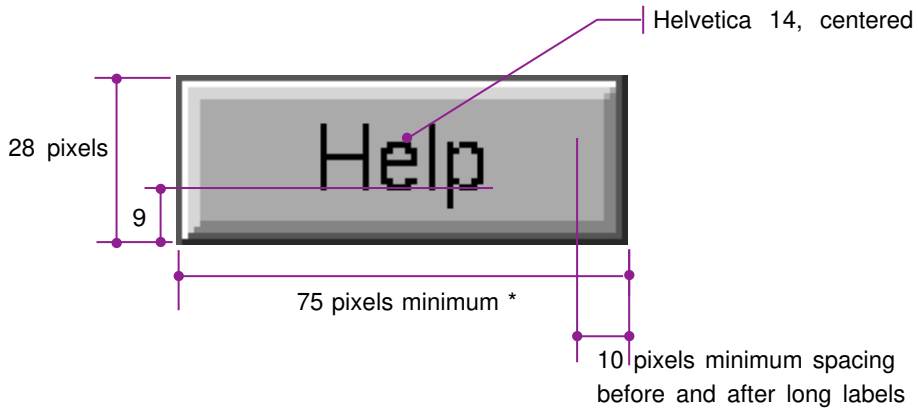
Loc+Sel

Loc+Sel

Push Button

The Push Button consists of a button and a text label. Fixed Push Buttons are preferable to transient menus when the user needs to make several selections at once or is working primarily in a control panel area.

Rendering Specification



*Button width is the greater of 75 pixels or the actual text label width + 2 * 10 pixel spacing.

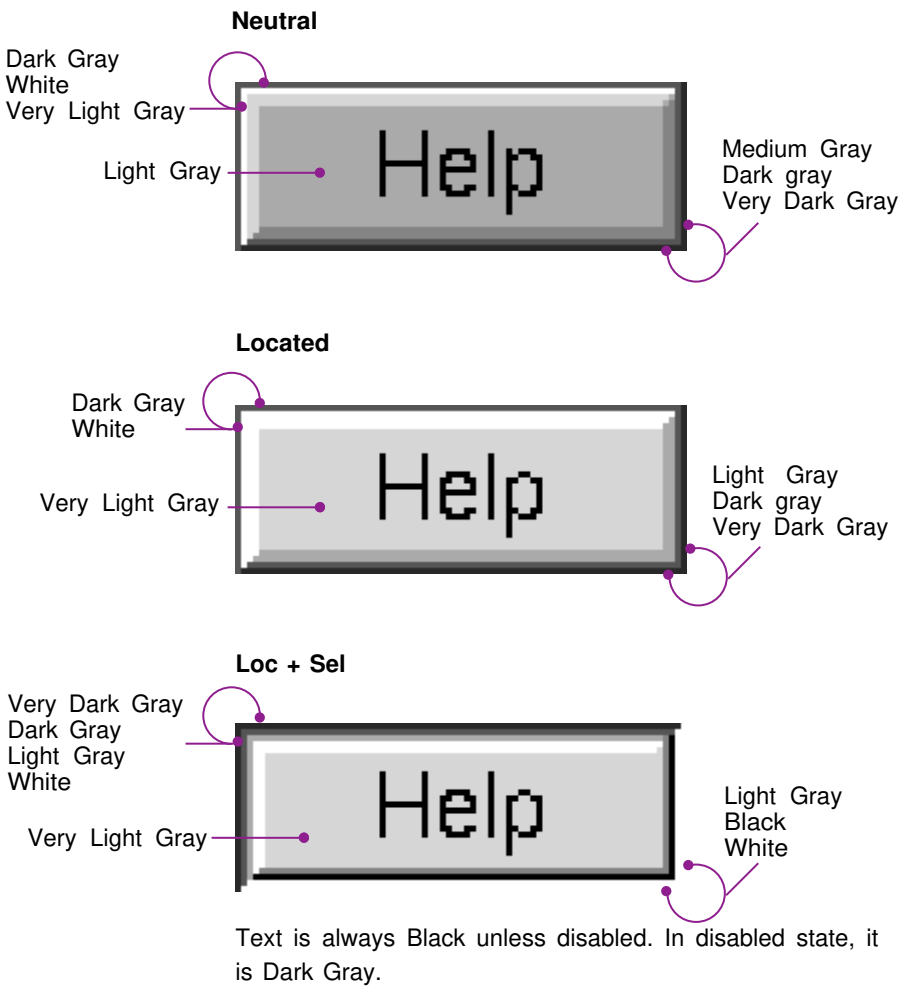
Push Button

Presentation States

State Transistions

[Transistions between the basic presentation states, and any other intermediate states, are outlined against user input, to help describe the behavior of the component.]

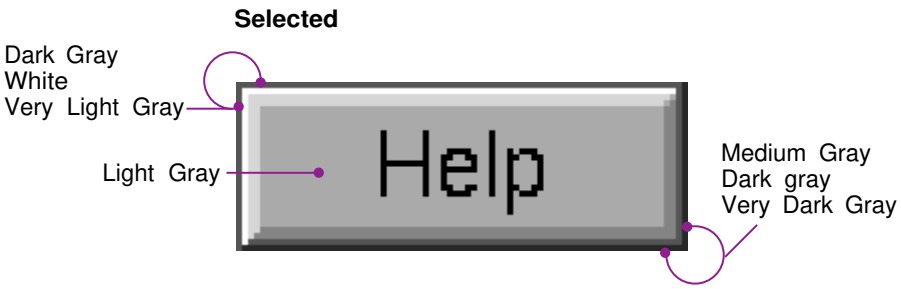
Rendering Details

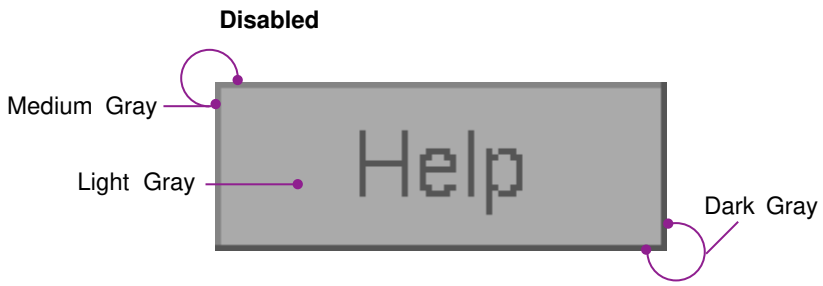


Text is always Black unless disabled. In disabled state, it is Dark Gray.

Push Button

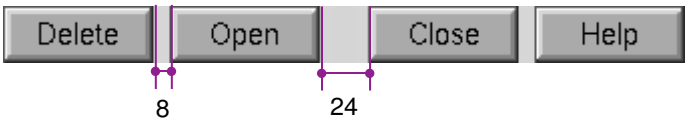
Rendering Details



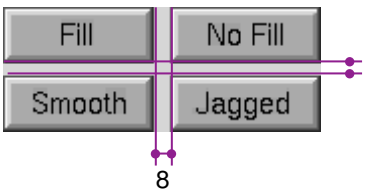


Layout Guidelines

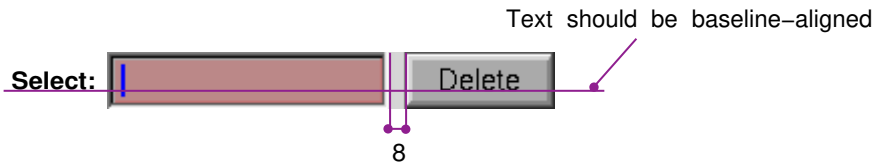
Fixed menus of Push Buttons are suitable for presenting high−visibility commands within a dialog or option box. Related Push Buttons should be grouped into distinct sets, and positioned within the window in a manner which reflects the mainstream workflow. Master commands which affect the entire window context should appear across the very bottom of the window, aligned from right to left starting in the lower right hand corner of the window.



8 pixels horizontally between push buttons,
24 pixels between groups of push buttons.



6 pixels verticaly between groups of push buttons

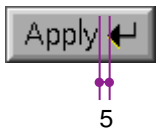


Push buttons and text entry fields are the same height, and can thus be aligned.

Push Button

Variations

One Push Button within a given window can be designated as the "default" Push Button for that window. When the <Return> key is pressed, the default Push Button is executed, providing a speedy shortcut alternative to clicking that Push Button with the Select Mouse Button (LeftMouse). The default Push Button should be clearly labelled with the Return Arrow symbol, to inform the user of the available shortcut.

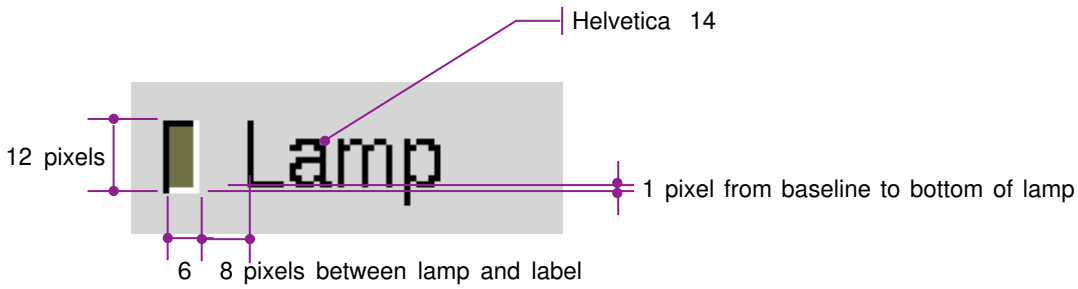


The button designated as the "default" case shows a Return Arrow symbol to the right of the text label. Leave 5 pixels of space between the text label and the arrow symbol, and center the resultant label–symbol pair on the face of the button.

Indicator Lamp

Use Indicator Lamps to provide graphical feedback about underlying state. They can be used alone, set directly into the face of the control panel, or they may be imbedded into other fixtures. Refer to the "*Toggle Button*"section for an example of imbedded Indicator Lamps.

Rendering Specification

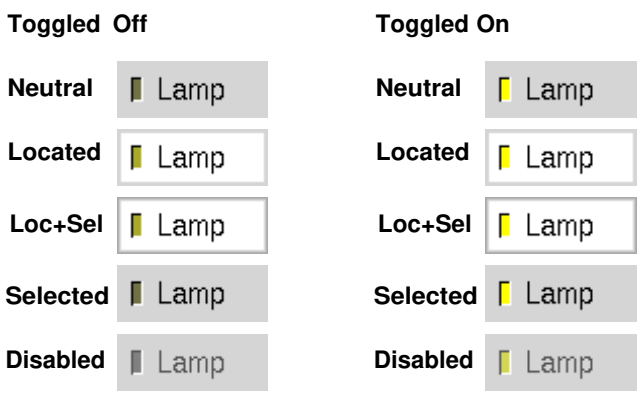


White highlighted area extends 6 pixels above, below, and to left of lamp. Length to right is variable. An extra 2 pixel border should be provided in the Loc + Sel cases (shown at left).

Lamp

Indicator Lamp

Presentation States



[**Note:** Show Presentation States and Rendering Details for the small square– and diamond–shaped variations.]

State Transistions

[Transistions between the basic presentation states, and any other intermediate states, are outlined against user input, to help describe the behavior of the component.]

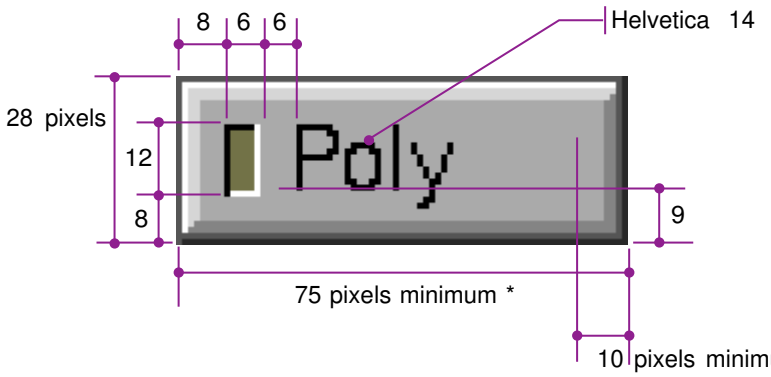
Toggle Button

The Toggle Button appears in the form of a Push Button, with an imbedded Indicator Lamp that provides state feedback.

You may use a Toggle Button instead of a Radio Button, to indicate that one of a range of options or modes is currently in effect. Alternatively, you may use a Toggle Button instead of a Check Button, to indicate whether a given option or mode is currently on or off.

In either usage, Toggle Buttons are most appropriate in a context where other Push Buttons are being employed, as part of a control panel. The Toggle Button is the same size as the Push Button. The embedded lamps are described in more detail in the section *"Indicator Lamps."*

Rendering Specification



*Button width is the greater of 75 pixels or the actual text label width + 20 pixels to the left button edge + 10 pixels to the right edge.

Toggle Button

Presentation States

	Toggle Off	Toggle On
Neutral		
Located		
Loc+Sel		
Selected		
Disabled		

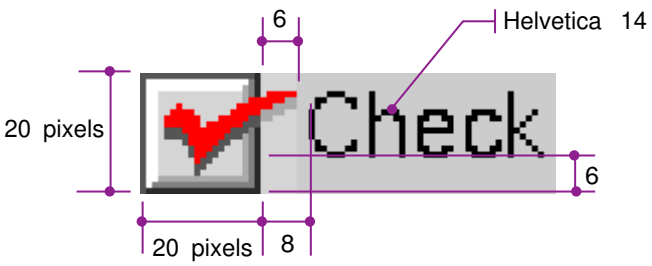
Check Button

Use a Check Button to set or unset the state of a given parameter or option. Each individual Check Button serves as its own field, controlling one independent variable within the application. You may use clusters of Check Buttons to manage a set of related parameters, such as bold, italic, and underline modifiers to a font style, and you may provide any number of these fields within the application.

The Check Button should always be presented with an accurate, up–to–date status (On or Off), whether the related variable has been changed by the user, via the Check Button interface itself, or as a side effect of the underlying application.

The Check Button is composed of a text or graphic label, preceded by the square check–box device itself.

Rendering Specification



Check Button

Presentation States

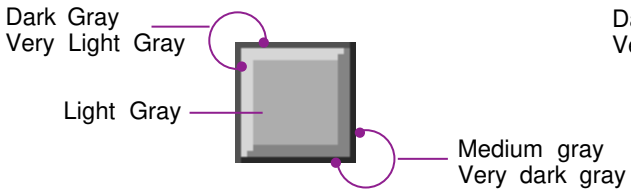
Toggled Off		Toggled On	
Neutral		Neutral	
Located		Located	
Loc+Sel		Loc+Sel	
Selected		Selected	
Disabled		Disabled	

State Transistions

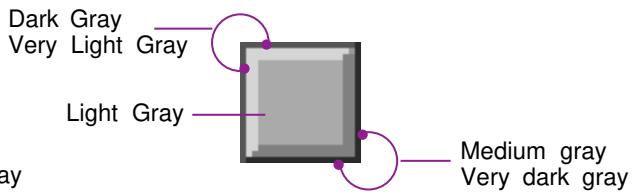
[Transistions between the basic presentation states, and any other intermediate states, are outlined against user input, to help describe the behavior of the component.]

Rendering Details

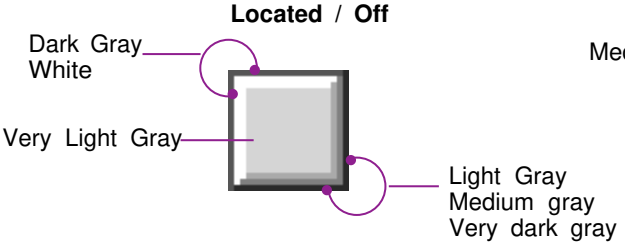
Neutral / Off



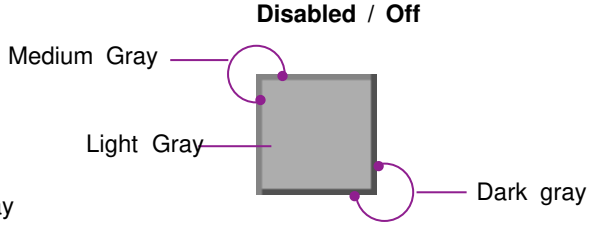
Selected / Off



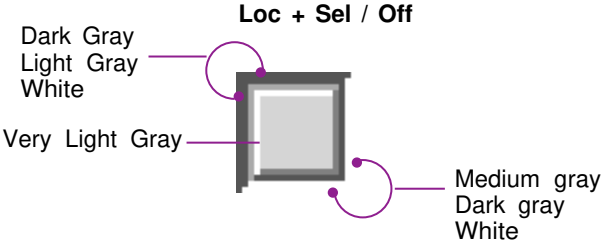
Located / Off



Disabled / Off



Loc + Sel / Off

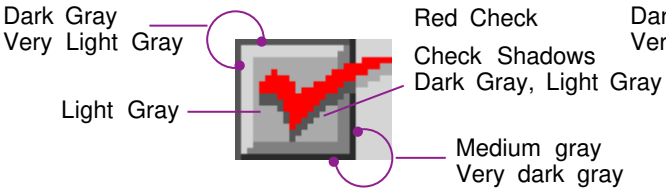


Label text is Black, except for Disabled Off and Disabled On. In those cases, label text is Dark gray.

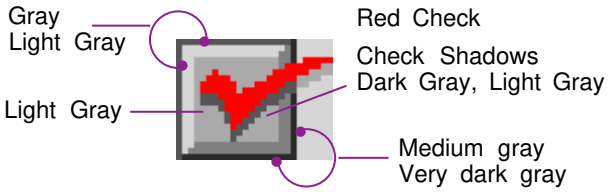
Check Button

Rendering Details

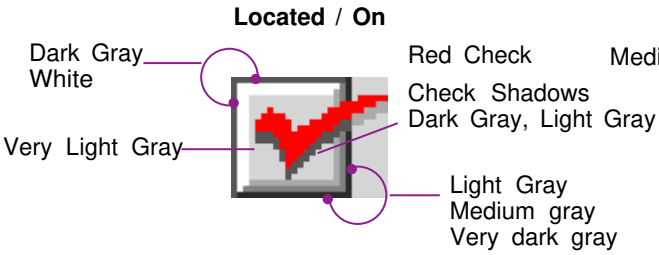
Neutral / On



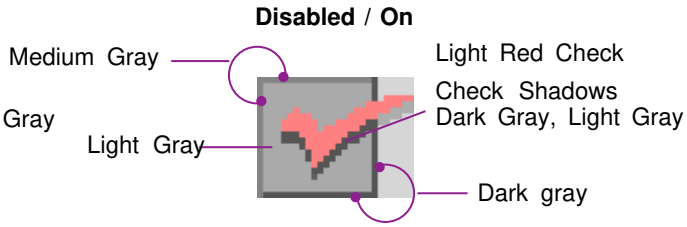
Selected / On



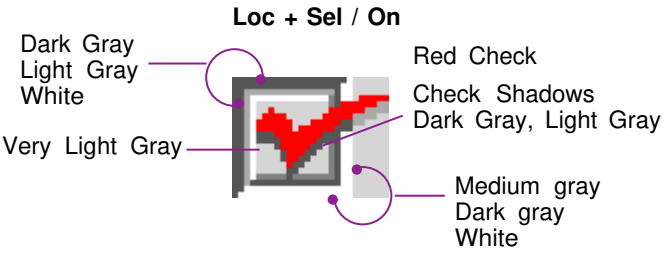
Located / On



Disabled / On



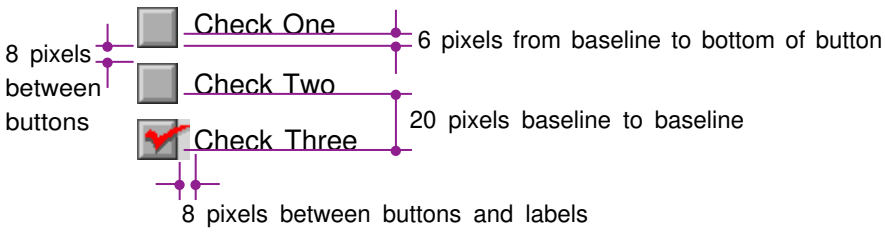
Loc + Sel / On



Label text is Black, except for Disabled Off and Disabled On. In those cases, label text is Dark gray.

Layout Guidelines

Sets of related Check Buttons are typically stacked with the usual 8 pixel space intervening. This standard spacing results in a 20 pixel baseline–to–baseline interval between stacked Check Buttons.



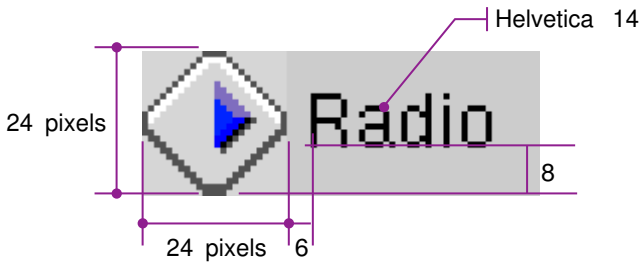
Radio Button

Use a Radio Button to select one option from a group of related, mutually exclusive options. You can provide any number of Radio Buttons within a single field (as a gang of related, alternative settings), and you may provide any number of these fields within the application.

Within a given field, a user can set only one Radio Button at a time to the On state. When users set one Radio Button, they automatically unset all the other Radio Buttons within that field.

The Radio Button is composed of a text or graphic label, preceded by the diamond–shaped button device itself.

Rendering Specification



Radio Button

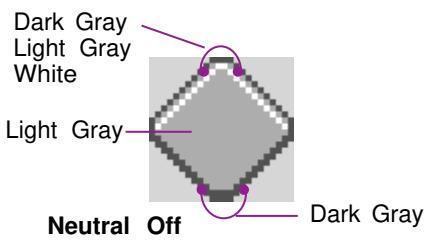
Presentation States

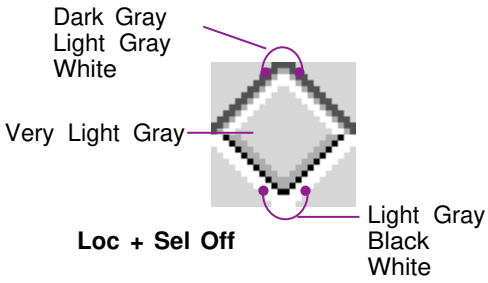
Toggle Off		Toggle On	
Neutral		Neutral	
Located		Located	
Loc+Sel		Loc+Sel	
Selected		Selected	
Disabled		Disabled	

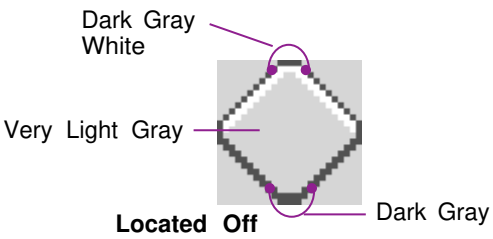
State Transistions

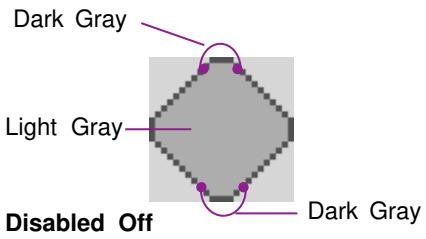
[Transistions between the basic presentation states, and any other intermediate states, are outlined against user input, to help describe the behavior of the component.]

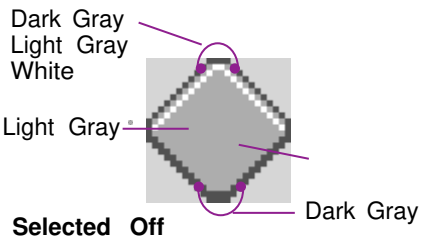
Rendering Details







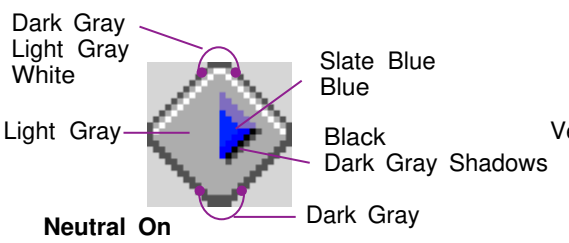


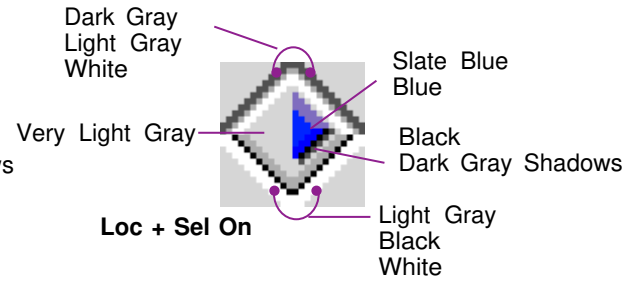


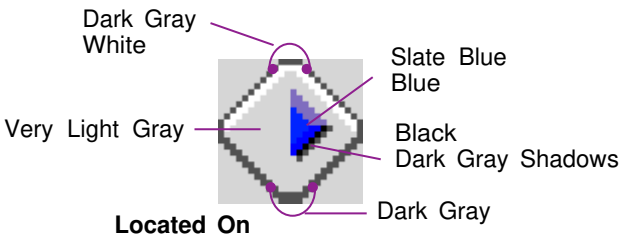
Label text is Black, except for Disabled Off and Disabled On. In those cases, label text is Dark gray.

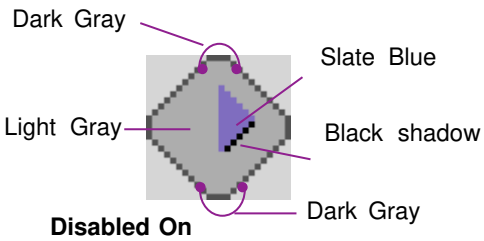
Radio Button

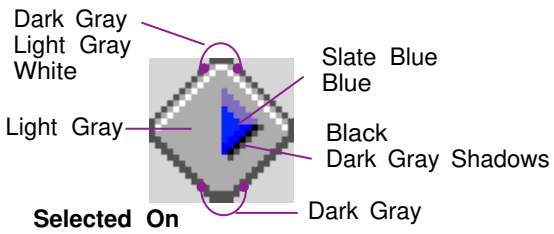
Rendering Details







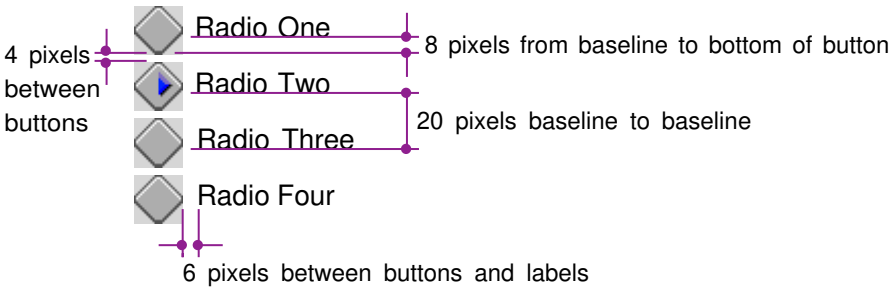




Label text is Black, except for Disabled Off and Disabled On. In those cases, label text is Dark gray.

Layout Guidelines

Due to the height of their diamond shape, Radio Buttons are typically spaced with 4 pixels intervening. This keeps the resultant 20 pixel baseline–to–baseline interval between stacked Radio Buttons aligned with other stacked widgets, such as Check Buttons, which employ the standard 8 pixel vertical spacing.

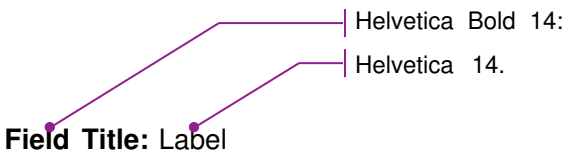


Field Title and Label

You use a Field Title to provide primary identification of the role of a given control or working area. It is used to name fields and groups of buttons which act together as a single field. Field Titles are displayed in Helvetica Bold 14, and are always followed by a colon. Abbreviations should never be used in Field Titles.

Labels provide minor text identification, appearing on the faces of Push Buttons, Option Menu Buttons, and opposite individual Check and Radio Buttons. Labels are also used to present output–only values, captioning tic marks on Dials and Valuators, as well as filling in output–only values to the right a Field Title. Labels are displayed in Helvetica 14.

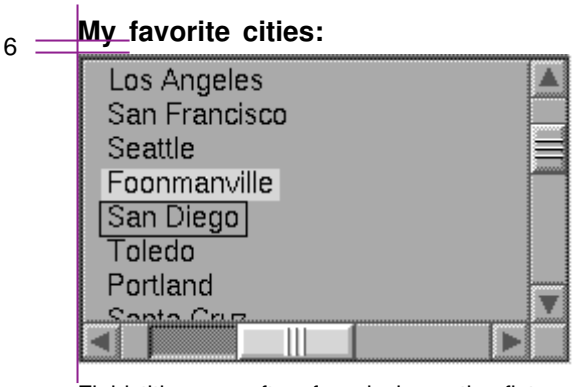
Rendering Specification



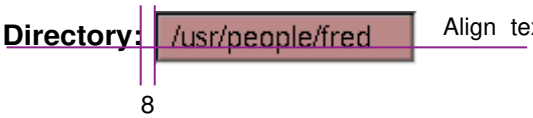
Field Title: Label

Field Title and Label

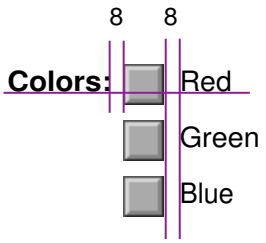
Layout Guidelines



Field titles are often found above the fixture to be identified. Leave 6 pixels vertical distance in this case. Note Left justification of field title and left edge of scrolling list.



They are also often seen to the left of the fixture.
A field title identifying a text entry field.



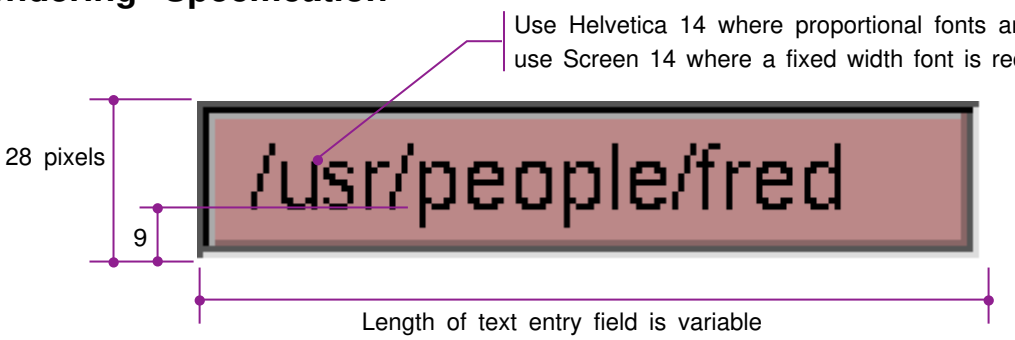
Field title identifies group of check buttons.
Labels identify each individual button.
Note position of labels to right of buttons.
This is true for radio buttons also.

Text Field

Use a Text Field to present text from the application to the user, and to receive text from the user into the application. You can use a Text Field for output only (read–only), or you can use one of the text editing configurations that support input. These include fixed–format text entry fields (for example, numeric–only or restricted character sets), a single–line, horizontally scrolling field suitable for entering filenames and other arbitrarily long "one liners", a compact scrolling field just two lines high, and a full–fledged multi–line text editor.

Use the distinctive "terra cotta" color for any field that supports text entry from the user. This color tells users that they can type input text into the field. Never use this reserved background color for scrolling lists or other output–only fields, as their contents may not be edited free–form directly from the keyboard.

Rendering Specification



Text Field

Presentation States

Text Entry Fields do not use locate or select highlighting in the same manner that most other components do. Instead, highlight the Text Field's internal text selection if and only if the field is selected as the one current keyboard input focus.

With Keyboard Focus: show text insertion cursor and sweep–selection

Without Keyboard Focus: hide text insertion cursor and sweep–selection

Do not display the text selection highlight when the Text Field loses input focus.

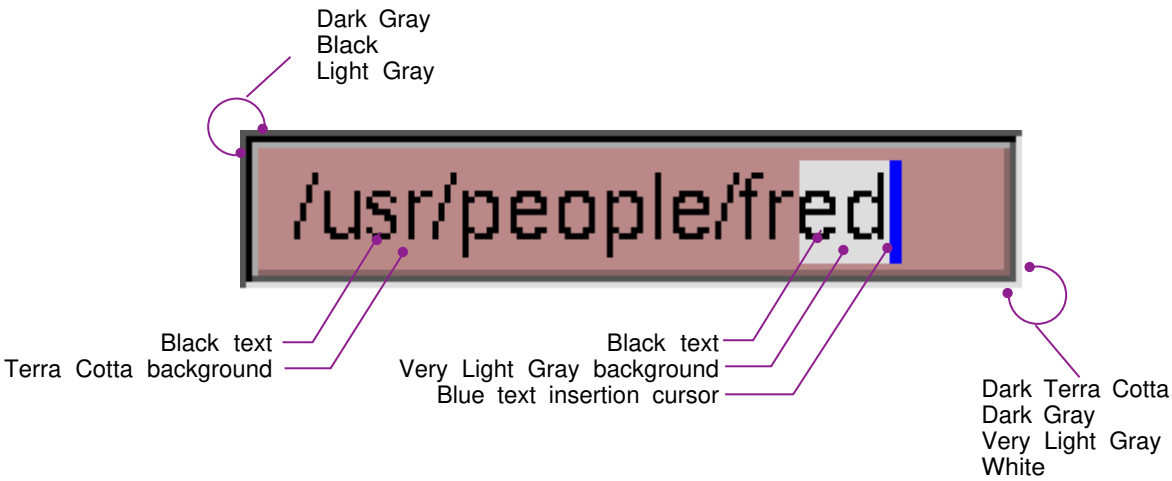
The SGI Style Guide recommends an explicit, "click–to–type" keyboard focus policy. Under this policy, each and every application window can have at most one candidate for the active keyboard input field among its various internal components. As the user makes a given application window the current input focus, that window's candidate for the active keyboard input field gains the keyboard focus. The application may want to assign a convenient text field as its default candidate for input; however, the user can easily assign the keyboard input focus to a new field within that window, by clicking the Select mouse Button (LeftMouse) over the desired field.

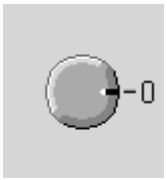
/usr/people/fred

Text Field

Rendering Details

Text displayed within a Text Field should always be black. Do not use colored text, as it interferes with legibility. The text insertion cursor is always blue, and the sweep–selection is always Very Light Gray, to make it the most salient text on–screen.



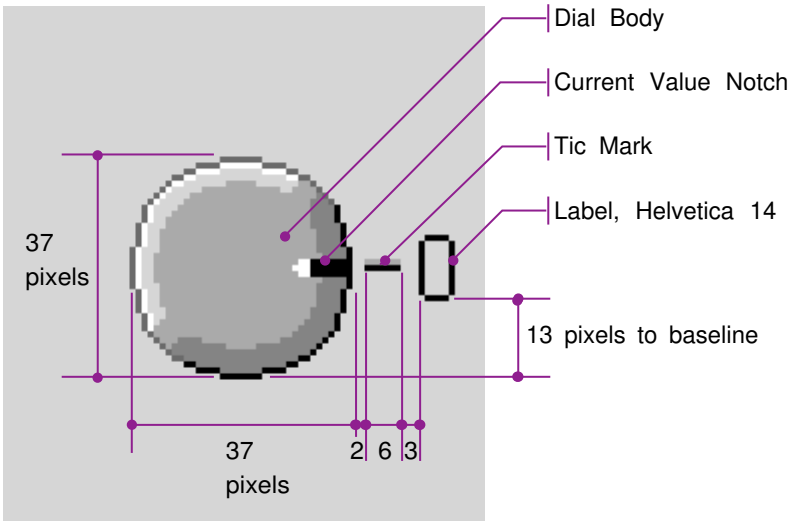


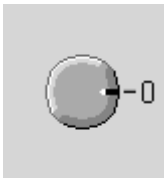
Dial

You can use a Dial just as you would use a Slider, to input a value from within a predefined range. The Dial provides an interface with different advantages from the traditional Slider, which can provide a closer match to your user's needs in some cases. The Dial can be more compact, supporting a larger number of distinct fields within a given size of control panel. The Dial can provide continuous, wrap−around values, appropriate to values such as rotation angles, where values wrap continuously modulo 360. Often, the rotary dial or rotary switch paradigm can be a more natural match to the user's expectations. For instance, the Hue and Contrast knobs for an (on−screen) video monitor feel more familiar as a set of small Dials, rather than a large, clumsy set of Sliders.

Dials can be accompanied by a numeric Text Field, to provide an accurate, up−to−date numeric readout for the Dial's current value. This approach allows users to edit this Text Field to set the Dial field to a specific value.

Rendering Specification





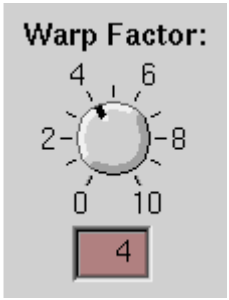
Dial

Presentation States

Neutral

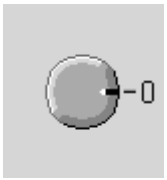
Located

Loc+Sel



Selected Not Applicable

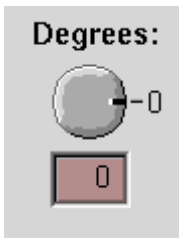
Disabled Not Applicable



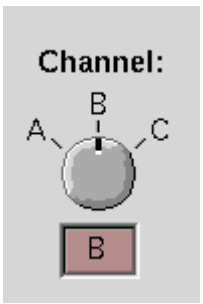
Dial

Variations

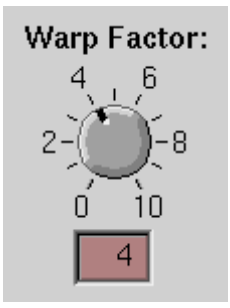
Important variations on the Dial include its use as a continuous, unconstrained rotating knob, a rotary switch which jumps to discrete click–stops, and a valuator which can be turned continuously within a constrained range of values, with support for both free–form placement and snap–to–tic constraints.



To minimize clutter, only a "Home" tic mark is provided. As the dial is adjusted, continuous numeric readout provides feedback between 0 and 360 degrees. Once the dial has been selected, moving the pointer further away from the dial body increases the "radius" of the effective dial, allowing greater precision for converging upon a given value.



A dial may be used to select among discrete choices. Here, the notch of the "rotary switch" should jump cleanly from one choice to the next. This approach is in contrast to the use of a dial for continuous numeric input, where the notch should be plotted at every perceptable position between the tic marks.



In a combined approach, waving the pointer near the dial produces constrained, discrete feedback, making it easy to choose an exact integer value. However, if the pointer is moved further away from the dial, the full range of intermediate values may be found.

Slider

Use a slider to input a value from within a predefined, continuous range. The standard slider consists of a thumb, which slides back and forth within a trough, and includes a set of tic marks and value labels which correspond to numeric values at the extremes and at useful intervals along its length. The length of the trough represents the entire range of represented values. The ridge in the thumb always aligns with the current value plotted along the range of tic marks.

In its typical, recommended configuration, a Slider is accompanied by a numeric Text Field, which provides an accurate, up−to−date numeric readout. Users can edit this Text Field to set the Slider field to a specific value.

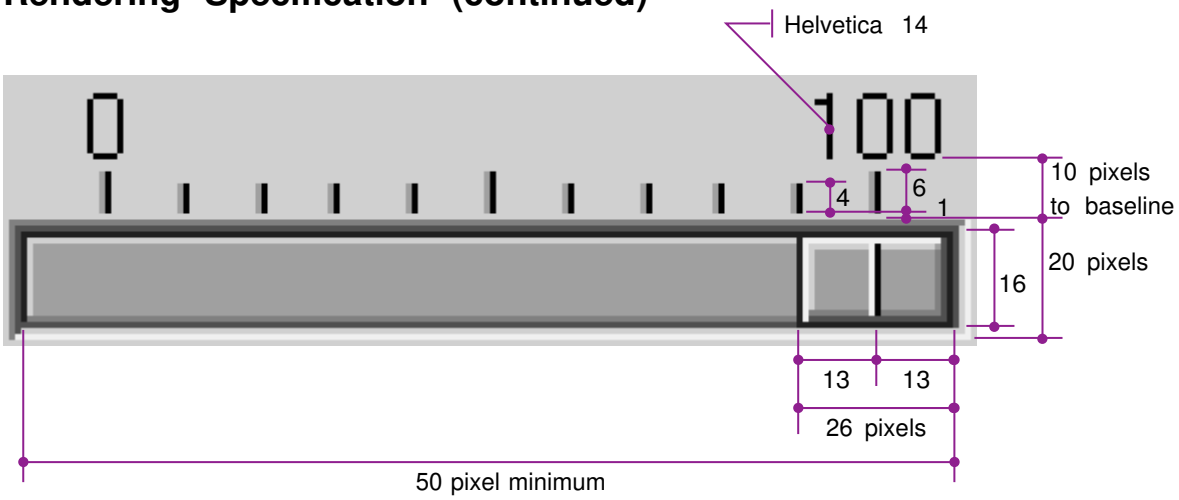
Sliders may be displayed either horizontally or vertically within a control panel.

Rendering Specification

The thumb of the Slider is differentiated by a single ridge, which locates the current value setting. Note that the thumb of a Scroll Bar shows 3 ridges, presenting a distinctive knurled appearance, while offering a familiar visual cue which hints "slide me" to the user.

Slider

Rendering Specification (continued)



Presentation States

Neutral

Located

Loc+Sel

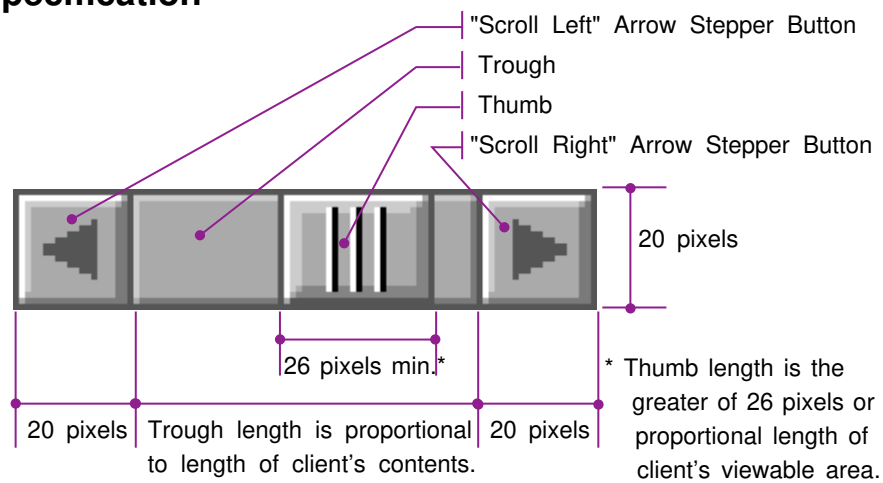
Selected

Disabled

Scroll Bar

You use a Scroll Bar to manage the view of the contents of an associated client component. The standard Scroll Bar consists of a Thumb which slides back and forth within a Trough, accompanied by a pair of Arrow Stepper Buttons. The length of the Trough represents the entire length of the contents of the client, and the proportional position and length of the Thumb represents the subset of the contents currently visible within the client's view. As the Thumb is dragged within the Trough, the view of the client's contents is re–oriented continuously and simultaneously, allowing the user to browse the newly–revealed contents on the fly. The Scroll Bar may be used either horizontally or vertically, or in combination together to scroll the appropriate dimensions of the client's view.

Rendering Specification



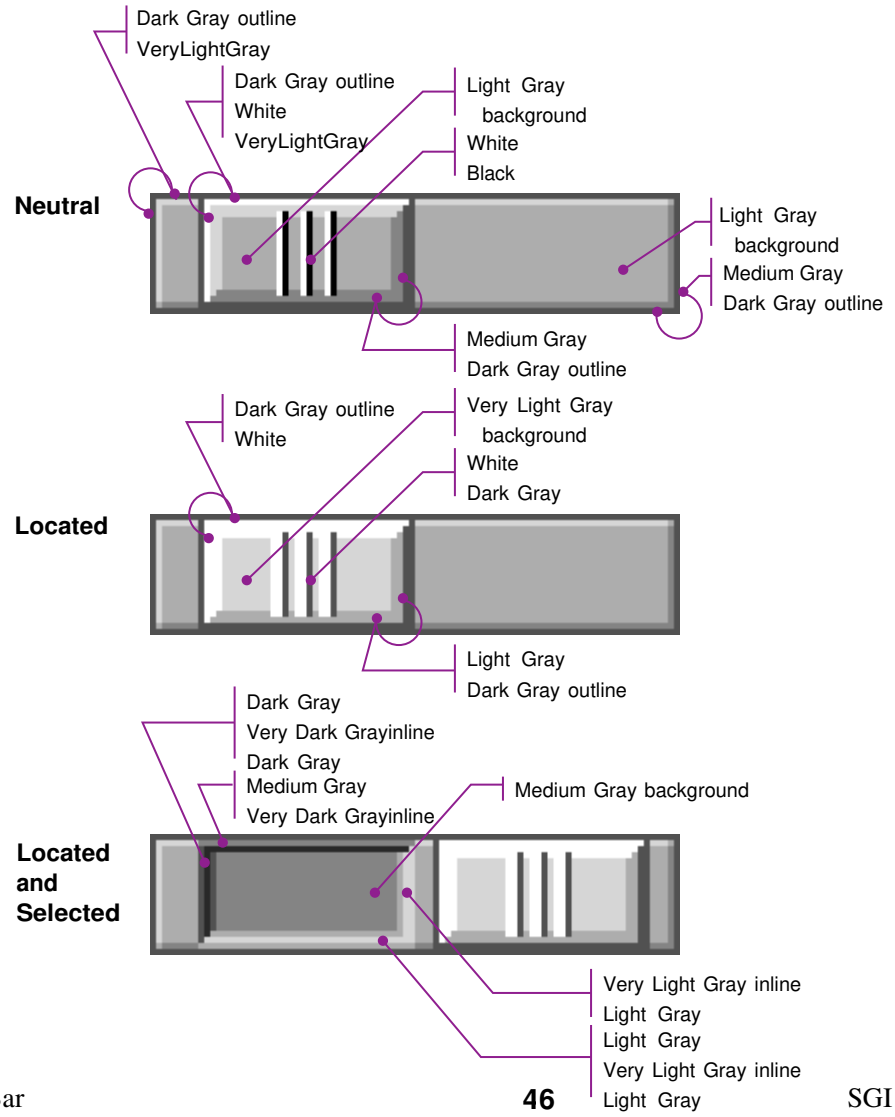
Scroll Bar

Presentation States

Scroll Bar Trough and Thumb		Stepper Arrows	
Neutral		**Neutral**	
Located		**Located**	
Loc+Sel		**Loc+Sel**	
Selected Not Applicable		**Selected**	
Disabled		**Disabled**	

Scroll Bar

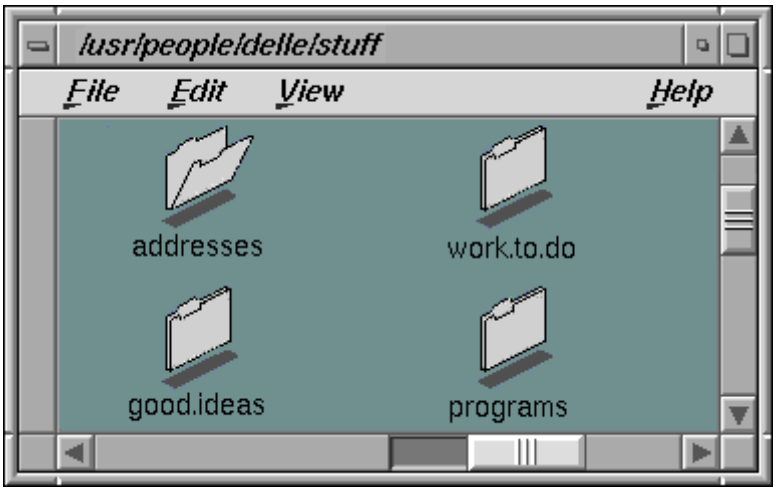
Rendering Details



Layout Guidelines

When assembling the Scroll Bar Trough and its surrounding Arrow Stepper Buttons, be sure to overlap their outermost single–pixel outline. This should also be done when abutting the Scroll Bar with its adjoining client Pane, and when abutting the Scroll Bar with the Window Border Frame.

A horizontal Scroll Bar should be presented along the lower edge of its adjoining client Pane. A vertical Scroll Bar should be presented along the right edge of its adjoining client Pane.

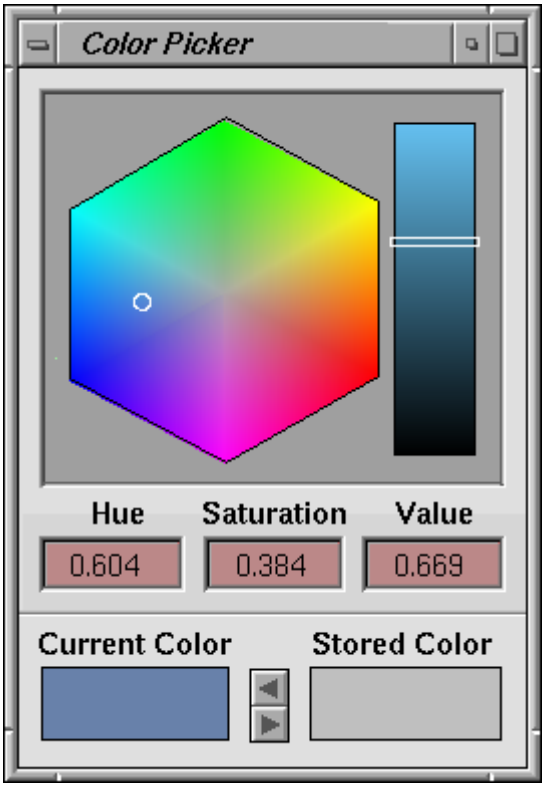


Typical example of horizontal and vertical Scroll Bar usage.

Arrow Button

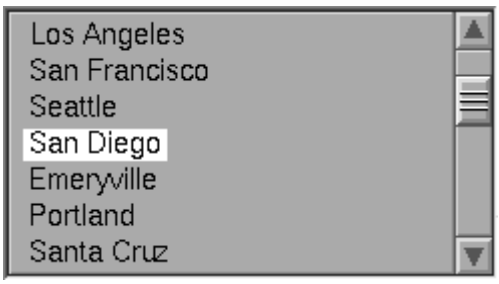
Variations

Arrow Stepper Buttons are most often used to increment an accompanying Scroll Bar or Valuator. When used with a small Text Edit Field or a short Scrolling List, these Arrow Stepper Buttons can be used in lieu of a full Scroll Bar for the sake of compactness, allowing the associated text to be scrolled one line at a time.



An alternative use of Arrow Stepper Buttons in conjunction with a numeric Text Field increments and decrements the numeric value displayed within the field.

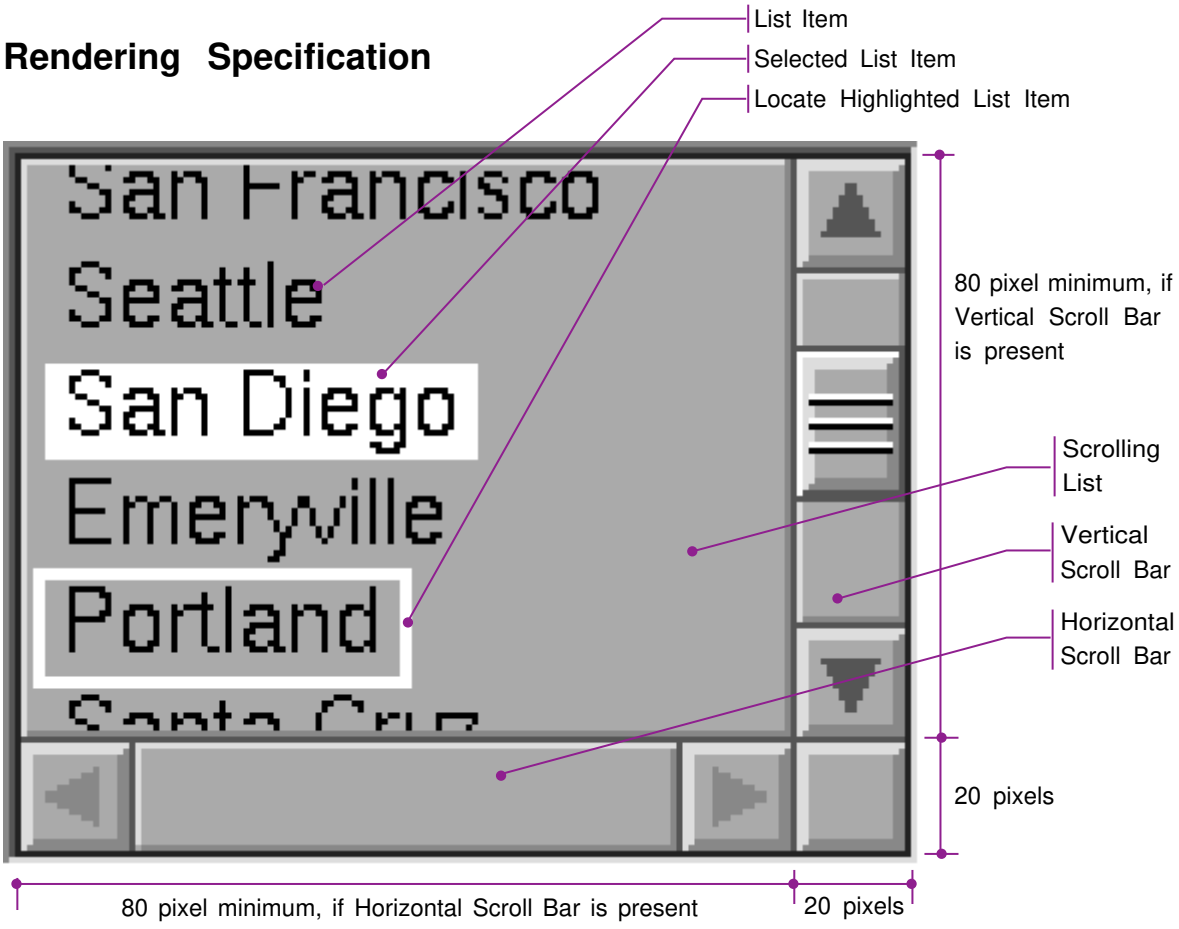
One Shot Arrow Buttons may also be employed as a shorthand for Push Buttons managing a transfer function between two fields. In the example on the left, the lower Arrow Button is used to transfer the color value displayed in the Current Color field, saving it away as the Stored Color. Similarly, the upper Arrow Button transfers the Stored Color value back, replacing the existing Current Color value.

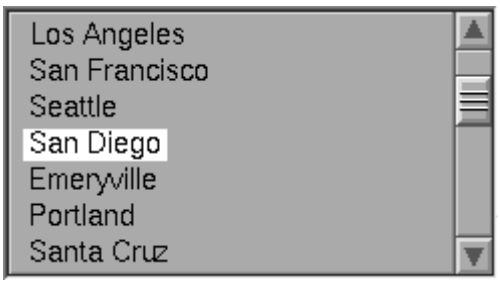


Scrolling List

The Scrolling List is used to present a list of items for browsing and selection. Typically these are text items, although a Scrolling List may contain stacked entries composed of icons, graphics, static text, editable text, or combinations of these on a single line.

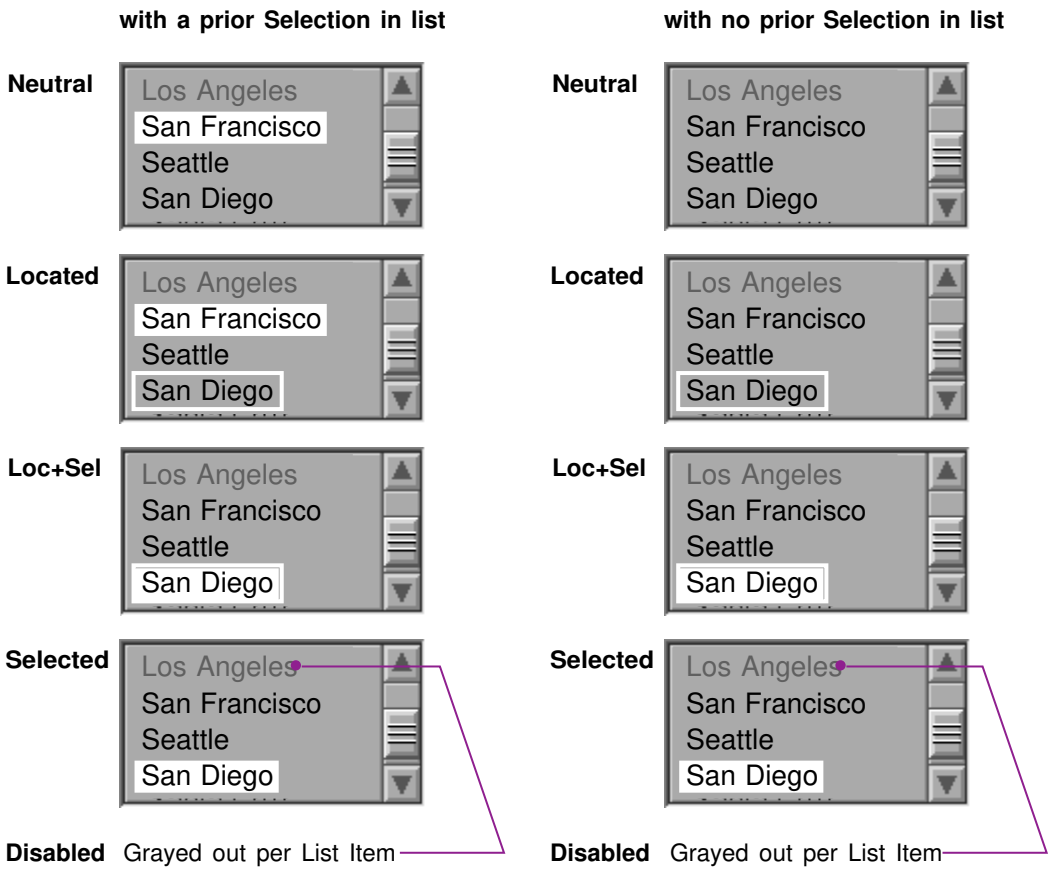
Rendering Specification

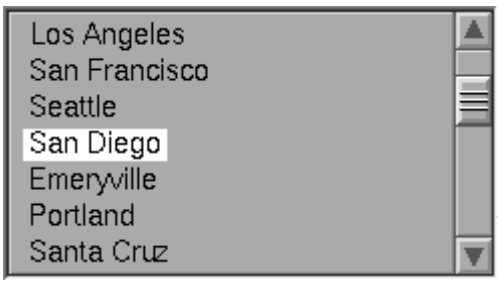




Scrolling List

Presentation States

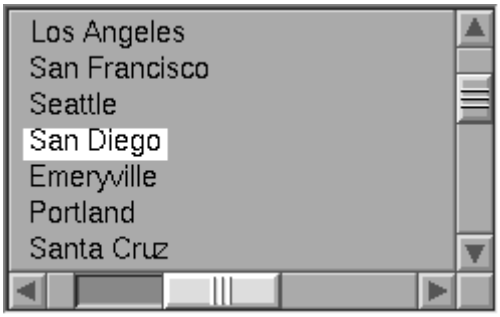




Scrolling List

Variations

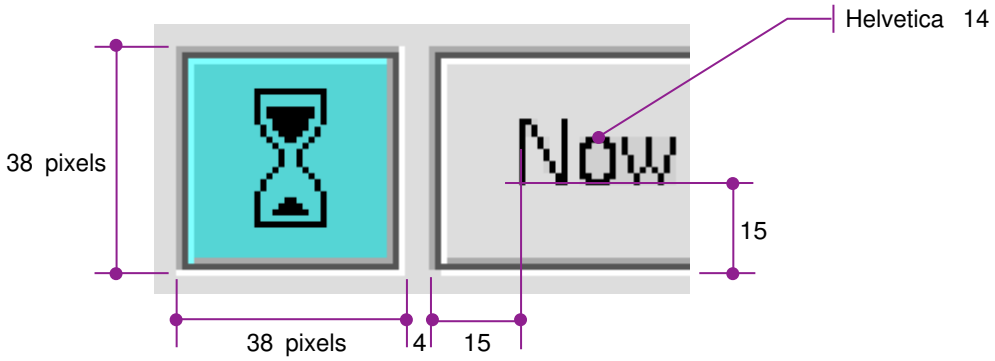
A Scrolling List with only a Vertical Scroll Bar offers a less cluttered presentation, desireable when list items of a know maximum length can be accomodated. Where items of arbitrary width may be encountered, a Horizontal Scroll bar should be added so any and all list items can be read in their entirety.



Dialog Box

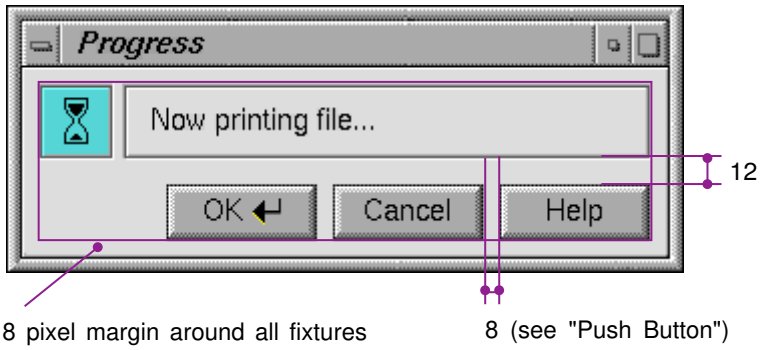
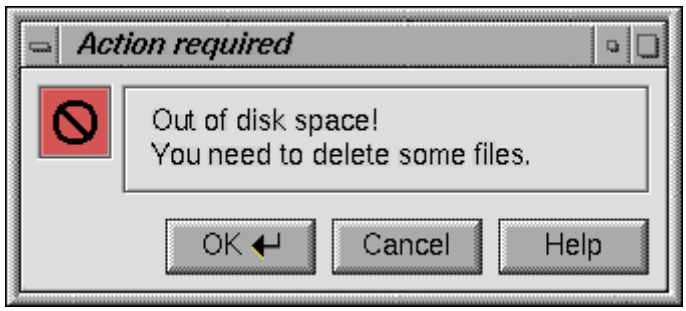
There are five standard types of dialog boxes: Information, Progress, Question, Warning, and Action.

Rendering Specification



Layout Guidelines

The width of the box is variable, depending upon the length of the message. However, the width of the box should not exceed good readable proportions; a 600 pixel maximum is recommended. The message may be broken into more than one line. See "Variations" below for an example of a multiple line message.



Variations