Motif and CDE Style Guide Reference

Preface

- Audience
- Applicability
- Organization
- Related Documents
- Conventions
- Problem Reporting

Motif and CDE Reference Topics

- Sample Topic
- Action (Choice Type)
- Action Message
- Active Window
- Adjustment Techniques
- Application Design Principles (CDE)
- Area Adjust Click Technique
- Area Adjust Swipe Technique
- Area Click Technique
- Area Swipe Technique
- Browse Technique
- Cancel (Action and Choice)
- Cascading (Choice Type)
- Change View (Choice)
- Check Box (Control)
- Choice
- Clear (Action Choice)

Client Area

Clipboard

Close (Choice)

Combination Box (Control)

Combination Text–List Control (Control Type)

Command Area (Area)

Command Box (Control)

Component Activation (CDE)

Controls, Groups, and Models (CDE)

Container (Control)

Context–Sensitive Help

Control

Control Navigation

Copy To (Dialog Choice)

Cursor

Cut, Copy, Paste (Action Choice)

Data Transfer

Default Action

Delete (Action Choice)

Dialog (Choice Type)

Dialog (Window)

Direct Editing

Direct Manipulation

Drag–and–Drop Transfer

Drag/Move (Action Choice)

Drop–Down Combination Box (Control)

Drop–Down List (Control)

Edit (Menu)

Emphasis (Cue)

Exit (Choice)

File Menu

File Selection (Dialog)

Find (Choice and Dialog)

First–Letter Cursor Navigation

Gauge (Control)

Group Box

Group Heading (Label Type)

Help (Menu/Action Choice)

Icon

In–Progress Message

Include (Choice and Dialog)

Information and Message Areas (Area)

Information Message

Input Focus

Input Models (CDE)

Internal Navigation

Keyboard (Device)

Label

List Box (Control)

Margin Selection Techniques

Maximize (Choice)

Menu (Control)

Menu Bar (Menu Type)

Menu Cascade Button (Control)

Menu Guidelines

Message

Minimize (Choice)

Mnemonic

Mouse (Device)

Move To (Dialog Choice)

Multilevel Selection Technique

Multipage Control (Control Type)

Navigation (CDE)

New (Action Choice)

New Window (Choice)

Notebook (Control)

Object

Open (Choice)

Option Menu (Menu Type)

Options (Menu)

Palette Area (Area)

Paned Box (Control)

Persistent Cue

Point Technique

Pointer

Pointer (Predefined)

Pop–Up Menu (Menu Type)

Prefix Completion

Prefix Navigation

Primary Transfer

Primary Window

Properties (Choice)

Pull–Down Menu (Menu Type)

Push Button (Control)

Push Button (Predefined)

Quick Transfer

Radio Button (Control)

Range Adjust Click Technique

Range Adjust Swipe Technique

Range Click Technique

Range Swipe Technique

Restore (Choice)

Sash (Control)

Save/Save As (Action Choice)

Scroll Bar (Control)

Scrolling

Secondary Window

Selected Menu

Selection

Selection (CDE)

Selection Box (Control)

Selection Dialog (Secondary Window)

Selection Models

Selection Modes

Selection Policies

Selection Techniques

Shortcut Key

Size (Choice)

Size Border (Control)

Slider (Control)

Sort (Choice and Dialog)

Spin Box (Control)

Spring–Loaded (Control Type)

Spring Sensitive (Mode)

Stacking Order

Static Text

Status Area (Area)

Tab Group

Tear–Off Menu (Menu Type)

Text–Display Field (Control)

Text–Entry Field (Control)

Text Field (Abstract Control)

Tool (Value Choice Type)

Touch Adjust Click Technique

Touch Adjust Swipe Technique

Touch Swipe Technique

Undo, Redo, Repeat (Action Choices)

Value (Choice Type)

Value Set (Control)

View

View Menu

Viewing Area (Control)

Warning Signal

Window (Menu)

Window Frame

Window Icon

Window Icon Box

Window Management (CDE)

Window Menu

Window Navigation

Window Title

Workspace

Preface

The Style Guide Reference is a supplement to the Style Guide. This reference contains the guidelines for the topics discussed in the style guide.

For specific details on coding an application program, widget, or window manager, refer to the Related Documents section.

Audience

This document is written for five audiences:

Application designers

Any designer who uses Motif as a basis for building an application

Widget designers

Any designer who adds functionality to the existing Motif software

User interface system designers

Any designer who designs an interface for users with specific cultural and accessibility needs

Window manager designers

Any designer who uses Motif to govern the layout and controls that affect application windows

Common Desktop Environment application designers

Any designer who builds an application for the Common Desktop Environment

You should read through the entire Style Guide once to familiarize yourself with all user interface design concepts. If you are already familiar with graphical user interfaces and their design concepts but need specific guidelines, use the following table to go to the topic you are interested in:

For guidelines on...	**Read...**
controls	The (Control) reference pages
menus	The (Menu) or (Menu Type) reference pages
choices	The (Action Choice), (Choice), (Choice Type), or (Dialog Choice) reference pages
an input device	The (Device) reference pages
a specific topic	The Style Guide and refer to the related reference pages in this book

Applicability

This is revision 2.1 of this document. It applies to Version 2.1 of the Motif software system and Version 2.1 of the Common Desktop Environment system.

Organization

This document is organized alphabetically into topics that are covered in the Style Guide. For information on the layout of the reference topics, see the Sample Topic reference page.

Related Documents

For documentation on Motif and CDE style, see the following documents:

- Motif and Common Desktop Environment: Style Guide
- Motif and Common Desktop Environment: Style Guide Certification Checklist

For additional information about Motif, refer to the following documents:

- Motif User's Guide
- Motif Widget Writer's Guide
- Motif Programmer's Guide
- Motif Programmer's Reference
- Motif Release Notes

For additional information about the Common Desktop Environment, refer to the following documents:

- Common Desktop Environment: User's Guide
- Common Desktop Environment: Advanced User's Guide
- Common Desktop Environment: Information Manager Users Guide
- Common Desktop Environment: Application Builder User's Guide
- Common Desktop Environment: ToolTalk Messaging Overview
- Common Desktop Environment: Programmer's Guide
- Common Desktop Environment: Help System Author's and Programmer's Guide
- Common Desktop Environment: Internationalization Programmer's Guide
- Common Desktop Environment: Information System Author's and Programmer's Guide

Conventions

This document uses the following conventions:

Typeface or Symbol	Meaning	Example
`Enter`	A key on the keyboard	When the user presses `Delete`, delete the text.
SELECT, ADJUST, TRANSFER, MENU	A virtual mouse button	The SELECT button selects text.
MB1, MB2, MB3	A physical mouse button (where MB1 is the leftmost button)	Make MB1 the SELECT button.
Push button	A new term	A *push button* is a control that displays a label or graphic that represents an action.
Control	A reference page	For more information, see the Control reference page.
ls	A command	Use **ls –a** to list all files.

Problem Reporting

If you have any problems with the software or documentation, contact your software vendor's customer service department.

Motif and CDE Reference Topics

This section lists all of the fundamental, recommended, and optional guidelines for designing and developing an application with a Motif toolkit.

Sample Topic

NAME

Sample Topic –Reference

Description

This sample topic describes how this chapter presents the reference material.

This section provides a technical definition of the item to be described begins each topic. Topics include guidelines for all controls and menu choices and for some concepts. Many topics include a graphical representation of the item being described.

When to Use

This section includes required, recommended, and optional guidelines on when to use the element described in this topic.

Required

> When the word *Required* precedes the statement, the statement is considered to be fundamental to developing an application that complies with the standard.

Recommended

> When the word *Recommended* precedes the statement, the statement is recommended to create an interface that is easy to use and consistent with other interfaces.

Optional

> When the word *Optional* precedes the statement, the statement is optional. That is, the guideline might be useful to consider when creating your interface, but is neither fundamental nor recommended.

Specific requirements may not apply to your application. For more information, see the Certification Checklist.

Guidelines

This section includes required and recommended guidelines for the topic. However, the specific requirements may not apply to your application. For example, the Selected menu topic includes required guidelines for a Paste choice on the Selected menu. If you do not supply this choice on your Selected menu, the requirement does not apply.

This section may also include guidelines related to specific designer or user tasks, such as providing keyboard activation.

Required

> When the word *Required* precedes the statement, the statement is considered to be fundamental to creating an application that complies with the standard and that is easy to use and consistent with other interfaces.

Recommended

> When the word *Recommended* precedes the statement, the statement is recommended to create an interface that is easy to use and consistent with other interfaces.

Optional

> When the word *Optional* precedes the statement, the statement is considered to be optional. That is, the guideline might be useful to consider when creating your interface, but is neither fundamental nor recommended.

Essential Related Topics

This section lists references to essential related topics. Read these topics to ensure that you understand how the element should be used and how it should interact, or how a menu−bar item relates to the menu structure. For example, because a fundamental guideline in the menu bar topic specifies assigning a mnemonic for the standard menu−bar choices, the mnemonic topic is listed as an essential reference. You should refer to the mnemonic topic for the specific guidelines that apply.

Supplemental Related Topics

This section lists topics that supply additional information related to the element, but that are not critical to understanding the element. For example, the menu bar topic includes a recommendation to provide tear−off menus. If you do not allow the user to tear off menus, you do not need to refer to that topic.

Action (Choice Type)

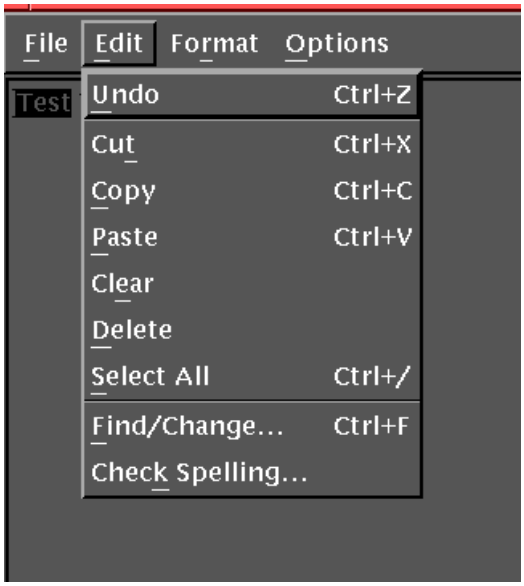
NAME

Action (Choice Type) –Reference

Description

An action choice is a type of choice that invokes an action that does not require further specification of parameters for invocation of the action. Close and Copy are examples of action choices. Action Choices illustrates action choices.

Figure 1 Action Choices



When to Use

Required

If an operation or action is to happen immediately after the activation of a choice, provide an action choice.

Guidelines

Required

When a user activates an action choice, immediately begin to perform the action.

Required

Provide access to all action choices through menus, push buttons, or both.

Required

Do not use an action choice to request further parameters in a secondary window; use a dialog choice instead.

Required

Do not use an action choice to temporarily display a list or menu; use a cascading choice instead.

Required

If an action choice can possibly refer to more than one control in a window, use the following rules to determine which control is affected:

If the action can be applied to the control that has focus emphasis, apply it to that control.

For example, if the focus is in a text−entry field and the user uses a shortcut key to invoke the Paste choice, paste to the text−entry field that has focus.

If the action operates on a selection, apply the action to the primary selection when the following is true:

The primary selection is in the window

No editing operation has been done in any other control in the window since it was selected

If the control with interacted emphasis is in the window and the action can be applied to it, apply it.

The control with interacted emphasis is the last editable control the user interacted with, so, for example, if the user last interacted with a text−entry field, paste to that field.

If the window has a main control and the action can be applied to it, apply it.

For example, a drawing program might have a large drawing area and a small text−entry field for entering identification numbers. The drawing area is the main control; if the previous cases do not apply, paste to it.

If none of the previous cases is true, then invoking the action has no effect except for invoking a warning message.

Essential Related Topics

For more information, see the Choice and Message reference pages.

Supplemental Related Topics

For more information, see the Cascading (Choice Type) , Selection , and Value (Choice Type) reference pages.

Action Message

NAME

Action Message –Reference

Description

An action message is a message displayed in a secondary window that indicates a condition has occurred that requires a response from the user. The user can correct the condition and then continue, withdraw the request, or get help.

There are three kinds of action messages: question messages, warning messages, and error messages.

Question messages

> Gets a response to a question. Display question messages when the user's immediate attention is not required, such as when the user's data will not be lost or deteriorate with time.

Warning messages

> Alerts the user to a possible danger. Display warning messages to give the user one last chance to retract an operation that is potentially destructive or reversible only with great difficulty.

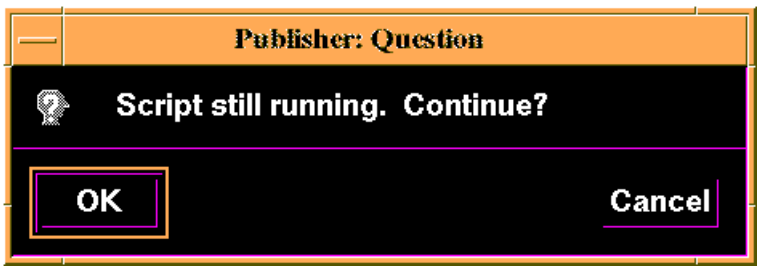
Error messages

> Conveys a message about a critical condition user error. Display error messages when the user's attention is needed to cancel the text or correct the situation.

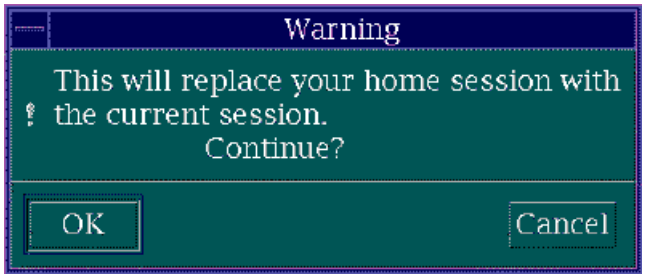
An action message suspends the task until the message is removed. You can display the message window as an application–modal window if user interaction with the application is to be restricted. At the very least, you should be careful about allowing any change of state that would affect the suspended task. Action Message Types illustrates the types of action messages.

Figure 1 Action Message Types

Question Message:



Warning Message:



Error Message

When to Use

Required

Display a question message when the user must respond to a question before continuing the task, but only when the task can be suspended indefinitely without harm and other user activity is unlikely to affect the continuation of the task at a later time.

Required

Display a warning message to alert the user to a possible danger and to allow the user to take some alternative action or to withdraw the task request.

Required

Display an error message when a response is needed before the task can be continued (or if it is likely that other user actions might make the task impossible to continue) and if it is essential that either the situation be corrected or the task canceled.

Guidelines

Required

Display the appropriate symbol to the left of each action message as follows (except in cases of bidirectional language support; see Chapter 11):

Question message

?

Warning message

!

Error message

0

Recommended

In a warning message, clearly explain the possible danger that can occur.

Recommended

In an error message, suggest possible actions that the user can take to correct the situation that caused the message to appear. For example:

```
Entry must be a hexadecimal value between 0 to 9 or A
to F.  Correct entry and choose RETRY.
```

Recommended

In the text of an error message, indicate the possible cause of the error.

Behavior

Required

Display an action message in a secondary window.

Recommended

Use a system–modal secondary window to display the message only if the user's immediate attention is required and if any user actions outside of the message window would make it impossible to resume the task or render it meaningless.

Recommended

Use an application–modal secondary window to display the message if the user can correct the situation without interacting with the application or must correct the situation by interacting through the message window.

Recommended

Use a modeless secondary window to display the message if the user cannot correct the situation by interacting through the message window but must interact with the application while the message remains displayed.

Recommended

When the user cancels a task, leave the objects or data in a form that is meaningful to the user. For example, when the user requests that an object be copied, and an error occurs that causes the user to cancel the task while the object is being copied, remove the partially copied object from the target destination.

Providing Controls in an Action Message

Recommended

Provide controls in the message window that allow the user to correct the situation that caused the message to appear or to request a related alternative action. For example, provide a text–entry field in which the user can correct a value, such as the name of the printer to use for printing a document.

Recommended

Provide a Retry push button that allows the user to continue or retry the task after correcting the situation that caused the message to appear. In an error message about a paper jam on the printer, provide a Retry push button that the user can activate after clearing the jam.

Recommended

Provide a Cancel push button that allows the user to withdraw the task.

Recommended

Provide a Help push button that allows the user to access the Help information about the task. For example, provide a push button labeled Help that explains how to unjam the printer.

Recommended

Provide at least the following choices in a question message:

Yes and No (or other appropriate set of answers)

Cancel (optional, use only if all the other choices perform some action)
Help

Recommended

Provide the following choices in an error message:
One choice or two choices labeled Retry and Cancel
Help

Essential Related Topics

For more information, see the Cancel (Action and Choice) , Information Message , Message , Push Button (Predefined) , and Warning Signal reference pages.

Supplemental Related Topics

For more information, see Chapter 4.

Active Window

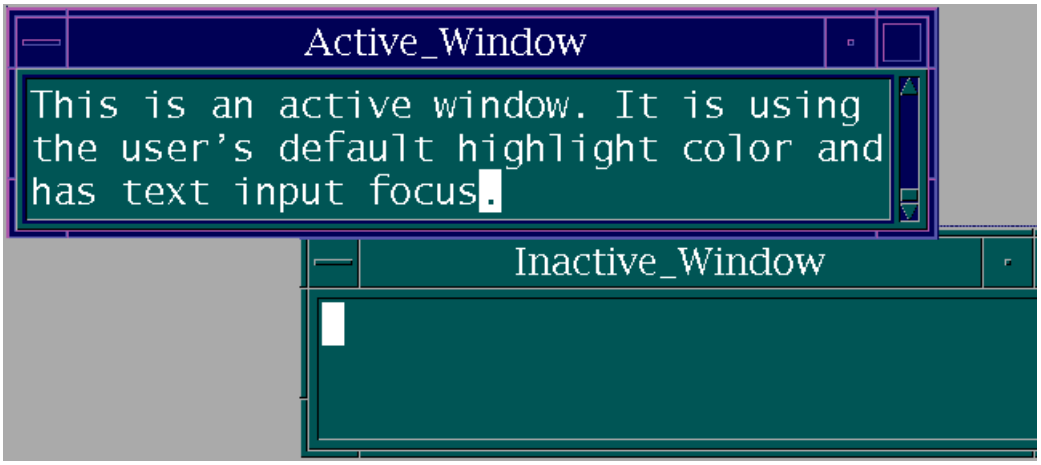
NAME

Active Window –Reference

Description

The active window is the window that currently has the input focus. The active window is indicated by emphasis displayed on its title bar and border, as specified by the operating environment.

Figure 1 Active Window



When to Use

Required

Make the window that can currently receive keyboard input the active window.

Guidelines

Required

If an explicit focus policy is in use, and an active secondary window is closed, move focus to the window that the active window is dependent on.

Required

When a window receives input focus, provide a visible difference in the window's title bar. For example:

Change the color of the active window's title bar and border to the color the user has specified for the operating environment.

Increase the saturation of the title bar to signal the change for monochrome systems and in situations where the user might have difficulty distinguishing colors.

Recommended

When the user opens a window but switches input focus to another window before the requested window is displayed, do not display the previously requested window on top of the window that currently has the focus.

Essential Related Topics

For more information, see the **Input Focus** reference page.

Supplemental Related Topics

For more information see Chapter 5.

Adjustment Techniques

NAME

Adjustment Techniques –Reference

Description

Adjustment techniques are variants of other group techniques. The user uses adjustment techniques to adjust the group of elements in the current selection region. Adjustment techniques have both click and swipe variants, as follows:

Adjust click techniques

The group is adjusted by taking a discrete action at a single adjustment point.

Via the mouse

Click the ADJUST button at that point.

Via the keyboard

Press `Shift Select` at that point (use `Space` or `Ctrl Space` instead of `Select` if there is no `Select` key).

Adjust swipe techniques

The group is adjusted by taking a continuous action between an initial and final adjustment point.

Via the mouse

Press, move, and release the ADJUST button.

Via the keyboard

Navigation augmented by the `Shift` key.

When to Use

Required

If you support a group technique, support its corresponding adjustment technique variants.

Required

Allow an adjustment technique to be used if and only if the previous selection technique used in the scope was:
A point technique
An associated group technique
Another adjustment technique associated with the same group technique
A margin or multilevel technique associated with the same group technique (for example, allow only the range adjust click technique to be used after a point, range [click, swipe, click adjust, or click swipe], or a range margin/multilevel technique)

Guidelines

Effects of Adjustment on Selection

Required

When using an adjustment technique, if the current selection region is selected instead of toggled, the following should occur:
Select all the elements in the new selection region.

Deselect all other elements in the scope.

Required

When using an adjustment technique, if the current selection region is toggled instead of selected, the following should occur:
Elements added to the selection region become toggled, based on the toggling policy.
Elements removed from the selection region have their state determined by the toggling removal policy.
The state of all other elements in the scope are unaffected.

Essential Related Topics

For more information, see the **Area Adjust Click Technique** , Area Adjust Swipe Technique , Range Adjust Click Technique , Range Adjust Swipe Technique , Selection Policies , Selection Techniques , Touch Adjust Click Technique , and Touch Adjust Swipe Technique reference pages.

Supplemental Related Topics

For more information, see the Selection Models and Selection Modes reference pages.

Application Design Principles (CDE)

NAME

Application Design Principles (CDE) -Reference

Description

Application design principles govern window, menu, and dialog layout within CDE.

Guidelines

Main Window Layout

Required

Compose your application with at least one main window.

A main window contains a client area and, optionally, a menu bar, a command area, a message area, and scroll bars. The client area contains the framework of the application.

Required

The default size of the application's main window must be large enough to accommodate a typical amount of data, but should not fill the entire physical display size to minimize visual conflicts with other applications.

Required

Include resize corners in any main window that incorporates a scrolling data pane or list.

Required

If your application has multiple main windows that serve the same primary function, close and iconify each window separately.

Required

If your application has multiple main windows that serve different primary functions, each window should be able to iconify independently of the other windows.

Window Title Layout

Optional

The title of your primary window (the main window your application displays to the user) should be the name of your application. Note that this does not have to be the actual name of the executable invoked by the user.

Carefully consider how the title you choose for your primary window works when it is used in icons and pop−up windows. If the name of the pop−up window is too long, you may remove the application title; however, without the title, users might have difficulty telling which pop−up window belongs with the originating primary window.

Optional

Use initial capital letters for each word in the title (in languages that support capitalization).

Optional

Follow the application name for each property window, as a minimum, with the title Properties and the name of the object it affects.

Optional

Begin the title of each pop−up window with the application title followed by a colon, then the title of the pop−up window. The colon should have a space both before and after it for readability.

Pop−up windows should always indicate which primary window they are associated with (which primary window invoked that pop−up).

Optional

Use a hyphen to denote the current file name when the application has files that can be loaded or saved. The hyphen should have a space before and after it. Only the base name of the file should be displayed, not the entire path.

The hyphen is used to denote specific instances of a window or data. The colon serves to delimit general categories or commands. For example, a file manager might have the following title for a Properties dialog box:

File Manager: Properties − myfile

Recommended

Follow the application name for each command window with the same title that is on the window button or window item that the user chooses to display that window.

Optional

In the case of multiple primary windows, include the application name at the beginning of each window title, and add a name that uniquely identifies that primary window. No separator should be provided for these names (for example, Calendar Manager Multibrowse, Catalog Search, Admintool Databases).

Optional

An abbreviated name for the application may be used on other windows, as long as it is done on all windows.

Menu−Bar Layout

Required

If your application has a menu bar, use a horizontal bar at the top edge of the application, just below the title area of the window frame. A menu bar organizes the most common features of an application. It contains a list of menu topics in cascading buttons; each button is associated with a distinct pull−down menu that contains commands that are grouped by common functionality.

Required

The menu bar should contain only cascading buttons.

Recommended

There are several common menu operations that are considered standard. The standard menu bar entries are File, Edit, View, Options and Help. If your application provides these functionalities, they should be included in the menu bar under the appropriate name.

Present standard menu−bar entries in the following order:

```
File Edit View Options Help
```

You should exclude from your menu bar any item shown in the preceding text if your application does not support the associated function. For example, if your application does not support the ability to display its data in different views, then you should not include a View menu.

You may add application−specific menus in between any of the standard menu items, with the following exceptions:

The File menu, if present, is located in the first menu position on the left.
The Help menu is located on the far right.
If File and Edit are present, they should be next to each other.

Recommended

Applications that are not file−oriented in nature (or that manage files transparently, not exposing this activity to the user) should replace the File menu with one or more application−specific menus.

Replacing the File menu:

Replacement1 <app−label> Selected
Replacement2 <app−label><obj−type>
Replacement3: <obj−type>

You may use Replacement1 if your application has more than one object type. Items on <app−label> would be used for global actions that are not specific to an object type. The items in Selected are actions that pertain to objects that are currently selected and may change, depending on what objects are selected. If nothing is selected, this menu should have a single item that says (none selected). If an item is selected, but there are no items that apply to that object, this menu should have a single item that says (none).

You may use Replacement2 if your application has a single object type. Actions that are global to the application are on <app−label>, and actions that are specific to the object type are on <obj−type>.

You may use Replacement3 if your application has a single object type and does not require an <app−label> menu. For example, a Print Manager might contain a Printer menu.

All other menu−bar guidelines that apply to File−oriented applications also apply to non−File−oriented applications. Thus, the following menu bar would be valid:

```
<app-label> Selected Edit <category1> View <category2> Help
```

Applications that are complex or are domain−specific (for example, an application for medical imaging and diagnosis of cat−scan data) may require other approaches to their menu−bar design. For example:

```
<app-label><category1><category2> Selected Edit <object-type> Options Help
```

Recommended

Place Exit or Close on the first (leftmost) menu of the menu bar.

File Menu Contents

Required

If the user chooses Exit, or in any other manner indicates that the application should be terminated, but if there are changes to the current file that have not been saved, display a dialog box that asks whether the changes should be saved before exiting.

Required

If your application uses a File menu, it must include the choices in **File Menu Choices** , with the specified functionality.

Table 1 File Menu Choices

Mnemonic	Menu Choice	Function
N	New...	Creates a new file. If the current client area will be used to display the new file, clear the existing data from the client area. If changes made to the current file will be lost, display a dialog box asking the user about saving changes.
O	Open...	Opens an existing file by prompting the user for a file name with a dialog box. If changes made to the current file will be lost, display a dialog box asking the user about saving changes.
S	Save	Saves the currently opened file without removing the existing contents of the client area. If the file has no name, display a dialog box prompting the user to enter a file name.
A	Save As...	Saves the currently opened file under a new name by prompting the user for a file name with a dialog box. If the user tries to save the file with an existing name, display a dialog box that warns the user about a possible loss of data. Do not remove the existing contents of the client area.
P	Print (recommended)	Schedules a file for printing. If your application needs specific information to

		print, it displays a dialog box requesting the information from the user. In this case, the menu entry is followed by an ellipsis (Print...).
C	Close (recommended)	Closes the current primary window and its associated secondary windows. If your application uses only a single primary window or multiple dependent primary windows, this action is not supplied
X	Exit	Ends the current application and all windows associated with it. If changes made to the current file will be lost, displays a dialog box asking the user about saving changes.

<*Object–type*> / Selected Menu Contents

Recommended

If your application uses an <*object–type*> menu or a Selected menu, it may include the choices in <object–type> / Selected Menu Choices , with the specified functionality, when the actions are actually supported by your application. Items should be presented to the user in the order listed.

Table 2 <*object–type*> / Selected Menu Choices

Menu Choice	Function
New...	Creates a new instance of the object type. If appropriate, a dialog box is presented allowing the user to specify the values for settings associated with that object.
Move To...	Allows the user to move the selected objects into a folder. Display a file selection dialog box allowing the user to select the desired folder.
Copy To...	Allows the user to copy the selected objects into a folder. Display a file selection dialog box allowing the user to select the desired folder.
Put in Workspace	Allows the user to put a link for the object onto the CDE desktop in the current workspace.
Any of the preceding menu choices should be provided only if the objects managed by your application are able to reside as separate entities outside of your application's main window. For example, a printer object created by a printer management application might be able to be placed in a Folder window and function as an application by itself. Your application should also support drag and drop as a method for performing any of these actions.	
Delete	Removes the selected objects. A confirmation dialog box should be presented to the user before the object is actually deleted.
Properties	Displays a Properties window that shows the current values for settings associated with the selected object.
<default action>	This choice should enact the default action for the selected object. The Open choice is a typical default.

Edit Menu Contents

Optional

If your application uses an Edit menu, it may include the choices in Edit Menu Choices , with the specified functionality, when the actions are actually supported by your application.

Table 3 Edit Menu Choices

Mnemonic	Menu Choice	Function
U	Undo	Reverses the most recently executed action.
t	Cut	Removes the selected portion of data from the client area and puts it on the clipboard.
C	Copy	Copies the selected portion of data from the client area and puts it on the clipboard.
k	Copy Link	Copies a link of the selected portion of data from the client area and puts it on the clipboard.
P	Paste	Pastes the contents of the clipboard into the client area.

L	Paste Link	Pastes a link of the data represented by the contents of the clipboard into the client area.
e	Clear	Removes a selected portion of data from the client area without copying it to the clipboard and does not compress the remaining data.
D	Delete	Removes a selected portion of data from the client area without copying it to the clipboard.
S	Select All	Sets the primary selection to be all the elements in a component of the client area.
l	Deselect All	Removes from the primary selection all the elements in a component of the client area.
a	Select Pasted	Sets the primary selection to the last element or elements pasted into a component of the client area.
R	Reselect	Sets the primary selection to the last selected element or elements in a component of the client area. This action is available only in components that do not support persistent selections and only when the current selection is empty.
m	Promote	Promotes to the primary selection the current selection of a component of the client area. This action is available only for components that support persistent selections.

Recommended

If your application does not provide an <object–type> or Selected menu, but allows the user to select data within the window and manage settings for the selected data, then provide a Properties... choice as the last item in the Edit menu.

View Menu

Recommended

If your application provides a View menu, include only those functions that affect the way the current data is presented. Do not include any option that alters the data itself.

Options Menu

Recommended

If your application has global settings that control the way the application behaves, provide an Options menu from which these can be set.

Help Menu Contents

Required

If your application includes a Help menu, include the set of choices in Help Menu Choices in CDE , with the specified functionality, when the actions are actually supported by your application.

Table 4 Help Menu Choices in CDE

Mnemonic	Menu Choice	Function
V	Overview	Provides general information about the window from which help was accessed or about the application overall. Place a separator after this choice.
I	Index (optional)	Provides an index that lists the topics for all help information available for your application.
C	Table of Contents (recommended)	Provides a table of contents that lists the topics for all help information available for your application.
T	Tasks (recommended)	Provides access to help information that indicates how to perform different tasks within your application.
R	Reference (recommended)	Provides access to reference information.

L	Tutorial (optional)	Provides access to your application's tutorial.
K	Keyboard (optional)	Provides information about your application's use of function keys, mnemonics, and keyboard accelerators. Also provides information on general CDE use of such keys.
M	Mouse (optional)	Provides information about using a mouse with your application.
M	Mouse and Keyboard (optional)	Provides information about your application's use of function keys, mnemonics, keyboard accelerators, and using a mouse with your application. Also provides information on general CDE use of such keys. Use, rather than separate, mouse and keyboard choices if this information is best presented together.
O	On Item (recommended)	Initiates context–sensitive help by changing the shape of the pointer to the question mark pointer. When the user moves the pointer to a component and presses the SELECT button, any available context–sensitive help for the component is presented. Set off with separators on both sides.
U	Using Help	Provides information on how to use the CDE Help Viewer. Set off with separators on both sides.
A	About *application name*	Displays a dialog box that indicates, minimally, the name and version of your application, and displays its icon or some other signature graphic for your application.

Attachment Menu Contents

Recommended

If your application uses an attachment menu, it may include the choices in Attachment Menu Choices in CDE , with the specified functionality, when the actions are actually supported by your application.

Table 5 Attachment Menu Choices in CDE

Menu Choice	Function
Add File...	Selects files and other items to be attached. Display a file selection box that allows the user to select the desired files to attach. The default button in the file selection box is Attach.
Save As...	Saves the currently selected attachments. The user is prompted with a file selection dialog box for indicating where in the file system the attachments are to be saved. When multiple attachments are selected, the name field is inactive and the current names of the attachments are used as the name of the new file. This menu item is active only when one or more attachments are selected.
Rename...	Renames the attachment icon. The application should provide in–line renaming of attachment icons, such as File Manager uses. If the application cannot provide in–line renaming, then Rename allows the user to rename an attachment by displaying a dialog box that requests the name from the user. This menu item is active only when a single attachment is selected.
Delete	Deletes attachments from the attachment list. This menu item is active only when an attachment is selected.
Select All	Selects all the attachments in the attachment list.

Pop–Up Menus

Recommended

If your application provides functions that apply to a data pane and not any specific element therein, then provide a pop–up menu that contains the frequently used data pane functions and that is accessible by pressing the MENU button when the mouse pointer is over the background of the pane or a nonselectable element within the pane.

Recommended

Your application should provide a pop–up menu for any element that is selectable within its data pane.

Pop–up menus provide access to frequently used functions and should be used pervasively throughout the CDE desktop environment. A pop–up menu may contain a collection of options that appear in different menus available from the menu bar. For example, it may contain items from both the File and Edit menus.

Recommended

When a pop–up menu is displayed over an unselected object, apply any action selected from the pop–up menu to that object only, and not to any other objects that might currently be selected. This helps to protect the user from inadvertently applying an action to objects that the user may not realize are currently selected. Pressing the menu button invokes a pop–up menu pertinent to the object under the mouse cursor whether it is selected to not; if the object under the mouse cursor and other objects are selected, the pop–up menu is pertinent to the selected set.

Recommended

Every pop–up menu in your application should have a title that indicates the function the menu performs or the element on which it operates.

Recommended

Make the functions accessible from within your application's pop–up menus also accessible from buttons displayed within the window or menus accessed through the menu bar.

Because pop–up menus are hidden, they should provide redundant access only to functions available from more visible controls within the application's windows.

Optional

If your application uses any of the common pop–up menu actions, the actions should function according to the specifications in Pop–Up Menu Choices in CDE .

Table 6 Pop–Up Menu Choices in CDE

Menu Choice	Function
Properties	Displays a Properties dialog box that the user can use to set the properties of the component.
Undo	Reverses the most recently executed action.
Primary Move	Moves the contents of the primary selection to the component. This action is available only in editable components.
Primary Link	Places a link to the primary selection in the component. This action is available only in editable components.
Cut	Cuts elements to the clipboard. If the menu is popped up in a selection, cuts the entire selection to the clipboard.
Copy	Copies elements to the clipboard. If the menu is popped up in a selection, this action copies the entire selection to the clipboard.
Copy Link	Copies a link of elements to the clipboard. If the menu is popped up in a selection, copies a link to the entire selection to the clipboard.
Paste	Pastes the contents of the clipboard to the component. This action is available only in editable components.
Paste Link	Pastes a link of the contents of the clipboard to the component. This action is available only in editable components.
Clear	Removes a selected portion of data from the client area without copying it to the clipboard. If the menu is popped up in a selection, deletes the selection.
Delete	Removes a selected portion of data from the client area without copying it to the clipboard. If the menu is popped up in a selection, deletes the selection.
Select All	Sets the primary selection to be all of the elements in the collection with the pop–up menu.
Deselect All	Deselects the current selection in the collection with the pop–up menu.
Select Pasted	Sets the primary selection to be the last element or elements pasted into the collection with the pop–up menu.
Reselect	Sets the primary selection to be the last selected element or elements in the component with the pop–up menu. This action is available only in components that do not support persistent selections and only when

	the current selection is empty.
Promote	Promotes the current selection to the primary selection. It is available only in components that support persistent selections.

Recommended

Pop−up menus for selectable objects should include the set of choices in Pop−Up Menu Choices For Selectable Objects in CDE , with the specified functionality, when the actions are actually supported by your application.

Table 7 Pop−Up Menu Choices For Selectable Objects in CDE

Menu Choice	Function
Move To...	Allows the user to move the selected objects into a folder. A file selection dialog box is displayed that allows the user to select the desired folder.
Copy To...	Allows the user to copy the selected objects into a folder. A file selection dialog box is displayed that allows the user to select the desired folder.
Put in Workspace	Allows the user to put a link for the selected objects onto the CDE desktop in the current workspace.
Delete	Deletes the selected object. A confirmation is displayed to the user before actually removing the object.
Properties (recommended)	Displays a dialog box that indicates the current settings for attributes associated with the selected object.
Help (recommended)	Displays a help window that pertains to objects of the type selected.

Optional

Organize choices within your pop−up menus in the following manner:

<choices that manage the object such as Open, Save, or Properties>

−−−−−−−−− separator −−−−−−−−−−−−−−−−

<standard edit menu choices such as Cut, Copy, and Paste>

−−−−−−−−−− separator −−−−−−−−−−−−−−−

<other choices>

Required

When a pop−up menu is popped up in the context of a selection, any action that acts on elements should also act on the entire selection.

Dialog Boxes

Required

Display an information dialog box such that it does not interrupt the user's interaction with your application.

An information dialog box conveys information to the user that does not require immediate attention so it does not need to be modal.

Menu Design

Recommended

If the selection of a menu item will result in the user being queried for more information, such as through the posting of a file selection dialog, the menu item should be followed by an ellipsis (...). This requirement does not apply to menu items that will result in a simple warning or confirmation dialog being displayed.

The use of an ellipsis helps set the user's expectation for the behavior of the interface. When selecting an item without an ellipsis, the user can expect an immediate result.

Recommended

Menus accessed from within your application should contain at least two menu items.

No menu should contain only one item. If your application provides a menu with only one item, move that item into another menu or make it a button within the window. The longer the menu, the more effort is needed for the user to access choices near the bottom. If your menu has a lot of choices, break it up into two or more menus, or group some items into submenus.

Optional

Submenus accessed from within your application should contain at least three menu items.

Submenus may be used to group like items into a single secondary cascading menu where putting the items into the primary cascading menu would make it too long. However, if your submenu contains only two options, consider removing the secondary cascading menu and putting the options into the primary cascading menu since it takes more effort for the user to access options located in a submenu.

Recommended

No menu in your application should contain more than 15 choices.

If your menu has many choices, consider breaking it up into two or more menus, or grouping some items into submenus.

Optional

If your application contains a menu that is expected to be accessed frequently, then provide a tear−off menu option.

The user should be able to tear off frequently accessed menus so that these can remain posted on the desktop.

Optional

Provide keyboard accelerators where appropriate.

If specific menu items within a menu are expected to be used frequently, not the menu as a whole, then provide keyboard accelerators for these items and display the keyboard accelerators in the associated menu to the right of the item to which they relate.

Recommended

The labels used for items in the menu bar should not appear as options within the menus themselves.

The names of items in the menu bar serve as titles for the options the menu contains. The name of the menu bar item should provide a term that accurately describes the concept of the category relating all of the menu items and should not be used as the name of any item within the menu itself.

Required

Dim (make insensitive) any menu choice that is not currently an appropriate selection.

Dimmed controls cannot be activated by the user and should appear only when the inactive state is short−term (that is, there is something the user can do within the application or the desktop environment to make the control become active). When the control is persistently inactive (because of the current configuration of the application or system, or a particular set of companion software is not currently installed), remove the control rather than dim it.

Recommended

If a menu item is used to indicate a selection state, use a check box or radio button to indicate the state of the item. Use a check box if a single item is used to represent on or off states, and use radio buttons for multiple adjacent menu items in which only one of the items may be selected.

Required

If radio buttons are used in a menu, use separators between each set of radio buttons and other menu items.

Recommended

If a check box or radio button is used on a menu item, display it as either selected or not selected. It should not disappear when in the unselected state.

Required

If your application uses a tear–off choice in a menu, make the tear–off choice the first element in the menu.

Required

Make all menus wide enough to accommodate their widest elements.

Dialog Box Design

Recommended

The title of dialog boxes used within your application should adhere to the conventions listed in Dialog Box Title Conventions .

Table 8 Dialog Box Title Conventions

Window Usage	Window Title Format
Message	<app or object name>: <action or situation>
Progress	<app or object name> : <action> in Progress
Action (Command)	<app name>: <action>
Object Properties	<app name>: <object–type> Properties
Application Options	<app name>: <type> Options

Required

Every dialog box should have at least one button that either performs the dialog box action and dismisses it or dismisses the dialog box without taking any action.

Recommended

If your application uses common dialog box actions, include the specified functionality and labels listed in Dialog Box Choices .

Table 9 Dialog Box Choices

Dialog Choice	Function
Yes	Indicates an affirmative response to a question posed in the dialog box.
No	Indicates a negative response to a question posed in the dialog box.
OK	Applies any changes made to components in the dialog box and dismisses the dialog box.
<command>	Applies any changes made to components in the dialog box, performs the action associated with the <command>, and dismisses the dialog box. <command> should be used in lieu of OK, Yes, or No as a button label when it provides more meaning to the user as to the action that will be performed when that button is clicked.
Apply	Applies any changes made to components in the dialog box and does not dismiss it.
Retry	Causes the task in progress to be attempted again.
Stop	Ends the task in progress at the next possible break point.
Pause	Causes the task in progress to pause.
Resume	Causes a task that has paused to resume.
Save As Defaults	Saves the current settings as the default settings that will appear the next time the window is displayed. The settings are not applied to any selected object and the dialog box is not dismissed. A Save As Defaults button should be provided if it is expected that a user would want to use different default values for a set of controls within a dialog box than those that you provide as the factory settings. For example, a Save As Defaults button might be provided in a New <object–type> window, allowing the user to indicate that whenever a new instance of that object type is created, the current values should be displayed as the default settings instead of the values given by the application.
Reset	Cancels any changes that have not yet been applied by your application. The controls within the dialog box are reset to their state since the last time the dialog box action was applied. If no changes have been applied within the current invocation of the dialog box, the controls are reset to the state when the dialog box was first displayed.
Reset to Factory	Cancels any changes that have not yet been applied. Components in the dialog box are reset to their default state and value as specified by

	the vendor that delivered the application (that is, the controls are restored to the original factory settings).
Cancel	Dismisses the dialog box without performing any actions not yet applied.
Help	Provides help for the dialog box.

Recommended

Dim any visible control that is not currently active or whose setting is currently invalid.

When the control is persistently inactive (because of the current configuration of the application or system, or a particular set of companion software is not currently installed), the control should be removed rather than dimmed.

Optional

Keep the size of your dialog boxes to a minimum. Remember that on low−resolution displays, dialogs may take up most of the screen real estate, and may even run off the edge of the screen if not designed correctly.

Optional

Avoid complexity in your dialog boxes. If your dialog box must support many functions, consider using an expandable dialog box, or use more than one dialog in a nested fashion.

Optional

Avoid the use of resize handles in your dialog box. However, you may use resize handles when resizing is useful in allowing users to see more information; for example, when your dialog contains a scrolling list that is likely to be quite long, and users will frequently need to search the list.

Optional

Every dialog box in your application should have exactly one default button that is activated when the Return key is pressed.

The default button should be associated with the most likely response from the user and should not be potentially destructive or irreversible. Some applications may have dialog boxes that do not reveal a default button until a specific set of fields has been filled or otherwise manipulated.

Optional

If a dialog box displayed by your application has controls that are considered to be advanced features, use an expandable dialog box, or use a multiple page dialog box that provides a <category> option menu that allows a user to navigate to each page.

Required

If your application provides settings that control the behavior of the application, display these settings in an application properties window that is accessible from an Options menu.

Controls that relate to advanced features should not be displayed with the set of options initially displayed to the user. The typical user should be presented with only those options that are necessary to use the basic functionality of the application. Users looking to access advanced functionality within the dialog box may use the <Category> option button (see Figure 7−1). If the number of advanced controls is few, or the settings for these controls are highly related to the settings of basic controls displayed in the dialog box (that is, the settings of the advanced controls change when the user changes settings for basic controls), you might choose to provide an expandable dialog box.

Property Windows

Required

If your application provides settings that control the behavior of the application, display these settings in an application properties window that is accessible from an Options menu.

Recommended

If your application manages objects and allows the user to see or modify settings for these objects, display these settings in an object properties window that is accessible from a Properties... choice in the Edit, <object−type>, or Selected menus, as well as from the pop−up menu associated with the object.

Recommended

If your application provides access to a Properties or Options window, include the following set of buttons in the order listed, with the specified functionality, when supported by your application:

OK

Applies any changes made to components in the dialog box and dismisses it. OK may be replaced by a more appropriate label, for example, Add. The alternate label should be a verb phrase.

Apply (Optional)

Applies any changes made to components in the dialog box and does not dismiss it.

Reset

Cancels any changes that have not yet been applied by your application. The controls within the dialog box are reset to their state since the last time the dialog box action was applied. If no changes have been applied within the current invocation of the dialog box, the controls are reset to the state when the dialog box was first displayed.

Reset to Factory (Optional)

Cancels any changes that have not yet been applied. Components in the dialog box are reset to their default state or value as specified by the vendor that delivered the application (that is, the controls are restored to the original factory settings).

Cancel

Dismisses the dialog box without performing any actions not yet applied.

Help

Provides help for the dialog box.

Recommended

If your application provides a Properties window that displays settings for a selected object, make the Properties window track the current selection and modify the state of any controls to accurately reflect the properties of the currently selected object.

File Selection Dialog Box

Optional

If your application allows the user to open or save files, then use the standard CDE file selection dialog box to allow the user to select specific files and directories.

All user interactions with the file system should be facilitated by providing a point−and−click style of choosing files and directories. The user should never be forced to memorize and type in file paths. The user must be able to explore the contents and structure of the file system by using scrolling lists. The expert user, however, should be able to directly enter a complete file path, as well as able to use relative paths and environment variables such as *$HOME*.

The labels and contents of the standard file selection dialog box may be modified as appropriate to make clear the particular context in which it is being used within your application.

Recommended

If your application allows the objects it manages to exist as separate entities within folders or toolboxes within the desktop environment, provide a Copy To menu option or button that displays a file selection dialog box that allows the user to select the desired folder in which an icon for the object should be placed.

Recommended

The file selection dialog box should not display hidden (dot) directories or files, unless your users depend on these types of files. If your application does support displaying hidden files, supply a check box that allows the user to toggle between showing and not showing hidden files or to toggle between showing and hiding files at a global level in your application.

Recommended

The file selection dialog box should not show the full pathnames for files and directories, but should show only the relative names, except for the directory text field.

The global CDE setting should be:

XmFileSelectionBox.fullPathMode: false

Unless your application overrides this behavior, your file selection dialog box should not show full pathnames in the list boxes.

Required

The file selection dialog box should recall the directory location that was previously set by the user.

For example, if the user brings up Save As and navigates to /users/jay/letters to save the file, the next time the user brings up Save As, the file selection box should be in the directory /users/jay/letters. The directory, however, should not be retained once the user has closed the primary window. When the user brings up Save As for the first time, it should resort to the default directory.

About Dialog Box

Optional

The About dialog box should contain a minimum set of information about the application that is visible in a single text pane.

That minimum set should be:

Application name
Version number
Release date
Copyright

Required

Include a Close button in the About dialog box. Other buttons are optional, such as Help and More.

Recommended

Include information about the operating system or other aspects required to run the application, for example, Common Desktop Environment 2.1.

Optional

Include a More Information dialog box for additional information such as development team credits, licensing, client, or xhost information.

Dialog Box Layout

Optional

Place controls within your dialog box in a left−right, top−down layout based on the order in which the user is expected to fill out or choose options within the dialog box.

Required

Push buttons that affect the dialog box as a whole, either by modifying its contents or layout, invoking the action of the dialog box, or dismissing the dialog box, should be located at the bottom of the dialog box.

There should be only one row of buttons at the bottom of a dialog box. If your application has dialog boxes that contain several global buttons, you may need to create two or more rows of buttons at the bottom of the dialog box. The last row should contain the standard dialog box buttons (OK, Reset, Cancel, and Help). If a dialog box contains buttons that are not related to the dialog box as a whole, but relate to a specific control within the dialog box, the buttons should be located with the control to which they relate.

Required

If your application provides an Apply button within a dialog box, also provide an OK button or <command> button that performs the

dialog box action and then dismisses it.

Optional

Do not use cascading buttons within dialog boxes unless there is absolutely no other design alternative that can be used without a negative impact on the layout of your dialog box.

In general, use cascading buttons only within menus and menu bars. Avoid their use in all other locations unless absolutely necessary.

Designing Drag and Drop

Recommended

Provide a drag−and−drop method for all objects represented as icons. Provide a drag−and−drop method for all elements that the user can directly manipulate.

Recommended

Any basic function that your application supports through drag and should also be supported drop through menus, buttons, or dialog boxes.

Drag and drop is considered an accelerator to functionality that is accessible through other user interface controls that your application supports. There should be no basic operation that is supported solely through drag and drop.

Recommended

Use an icon graphic in a dialog box or window to indicate that objects within the dialog box or window can be dragged. Use the same icon graphic used to represent the draggable object in File Manager. Place the icon adjacent to any display of the contents of the object, if such a display exists. If there is no such display, place the icon in the upper right corner of the dialog box or window, unless a more suitable placement is determined. The icon should be 32x32 in size and have a label under it. The label should indicate what kind of object the icon graphic represents. The icon graphic should also be used as the source indicator in the drag icon.

Required

During a drag operation, change the current pointer to a drag icon.

Recommended

During a drag operation, change the current drag cursor to include a source indicator.

Recommended

During a drag operation, change the current drag cursor to indicate invalid drop zones. Use the standard CDE cannot pointer.

The user must receive feedback as to where an object can and cannot be dropped. Minimally, you should provide feedback as to what are invalid drop zones. Preferably, feedback for valid drop zones is enhanced by use of animation, recessing of the target drop zone, and other such drag−over effects.

Recommended

During a drag operation, change the drop zone feedback to indicate a valid drop zone.

Preferably, feedback for valid drop zones is enhanced by use of animation, recessing of the target drop zone, and other such drag−over effects.

Required

When the user presses `Cancel`, end a drag−and−drop operation by canceling the drag in progress.

Required

When the user releases the TRANSFER button (or the SELECT button) when not over a drop target, end a drag−and−drop operation.

Optional

Any cursor change or drag−over effect your application uses should occur within .2 seconds of the mouse pointer reaching the target area

and should not interfere, in any noticeable way, with the interactive performance of the drag operation.

Recommended

In a collection that supports copy, move, or link operations that can be performed by dragging, the feedback presented to the user during the drag operation should indicate whether a single object or multiple objects are being manipulated.

Feedback provided during the drag operation should ensure that the user feels confident that the desired set of objects is being dragged. The drag icon used for multiobject drag operations should integrate the feedback used to indicate whether the operation is a move, copy, or link.

Required

After a successful transfer, place the data in the drop zone and remove any transfer icon.

Required

If your application removes data upon the completion of a drag−and−drop operation, do so only if the drag−and−drop transfer has completed successfully.

If a drag−and−drop operation has been canceled or failed, do not remove the data or object that was the source of the drag.

Required

After a failed transfer, keep the data at the drag source and do not place it in the drop zone. Remove any transfer icon.

Recommended

If the user drops an object at an inappropriate drop zone within your application's window, your application should participate in the display of a snap−back effect and also post an error dialog box that indicates the reason the drop was disallowed.

The error message should state the context (for example, running action A on object B), what happened (for example, could not connect to system X), and how to correct the problem (for example, press the Help button to obtain information on diagnosing remote execution problems).

Recommended

Applications that accept only single items should reject all multiple−item drops.

Recommended

If your application supports drag and drop as a means of loading a file into the application, have the application respond to this operation in a manner similar to when the file is loaded through more conventional means, such as choosing Open from the File menu.

As an accelerator, drag−and−drop loading of files should provide the same kind of feedback and behavior as choosing Open from the File menu. For example, if changes to a currently loaded file have not yet been saved, display a message dialog box asking whether the changes should first be saved before loading the new file.

Required

If your application provides any drag−and−drop help dialog boxes, include a `Cancel` button for canceling the drag−and−drop operation in progress.

Attachments

Recommended

Drag and drop should not be the only method for attaching objects.

Recommended

Double−clicking is a shortcut for selecting the attachment and choosing the Open menu item for attachments and should never be the only way to access attachments.

Recommended

When the user attempts to drop something into the attachment list that is not attachable, then the drop should fail and the item snaps back to its source.

Recommended

When the user has one or more attachments open for editing and attempts to do any operation that would result in potentially losing the user's edits, the user should be clearly warned and given the opportunity to save changes.

Recommended

When the user chooses something to attach from the file selection dialog box that is not an attachable item, display an error message explaining why the chosen item cannot be attached. For example:

```
The folder "My.Stuff" cannot be attached because it is a folder.
Only documents, applications, and scripts can be attached.
```

Installation

Required

Install applications in folders in the Application Manager, not directly on the Front Panel or subpanels. For consistency, only CDE desktop components will install in the Front Panel or subpanels. Users may choose to rearrange their Front Panel, but applications should not do this without user consent.

Interaction

Required

Display a warning dialog box that allows the user to cancel the destructive action about which the dialog box is providing a warning. The user needs to have a way to cancel an operation that can cause destructive results.

Required

When your application displays a dialog box, place the input focus at the first text field into which the user is allowed to type an entry, or at the first control within the dialog box with which the user should interact.

Input focus should always be placed at a predictable and intuitive location. Do not force the user to set focus at the control most likely to be used when the window is displayed.

Recommended

As the user presses `Tab` within dialog boxes of your application, move the input focus to different controls within the window in a left–right, top–down order.

Required

There should always be exactly one control within any window of your application that has the input focus if the window in which it resides has the input focus.

Optional

When a text field within your application does not have the input focus, do not display the text cursor within that field.

Although use of inactive text cursors is allowed within the Motif style, it is better to hide the text cursor on focus out rather than display the inactive text cursor. This makes it easier for the user to quickly scan the screen or a window and determine which text field currently has focus.

Optional

Provide keyboard mnemonics for all buttons, menus, and menu items displayed within the application.

Optional

Provide keyboard accelerators for those functions that are expected to be used frequently by the user.

Required

Dialog boxes should never block input to other applications within the desktop (that is, they are not system modal) unless it is essential that the user perform no other action on the desktop until the user responds to the dialog box.

Required

Dialog boxes should never block access to other functionality within the application (application modal) unless it is essential that the state of the application remains unchanged until the user responds to the dialog box.

Required

If your application does not use the values of global environment settings, such as multiclick timeout intervals, drag thresholds, window color settings, mouse left– or right–handedness, and so on, but instead uses its own values for these settings, then provide one or more Options dialog boxes that allow the user to change the values for these settings.

In general, you should not override the value of settings treated as global environment settings. The user controls these settings through the CDE Style Manager. If you choose to ignore these settings and specify your own settings, then your application will behave inconsistently with other applications in CDE. If you nevertheless choose to provide your own values, then you must provide the user with a way to make your settings consistent with the rest of the desktop.

Visuals

Recommended

Design any icons or graphics that your application displays to be distinguishable on low– (640x480), medium– (800x600), and high–(mega–pixel) resolution displays. Alternatively, provide different sized visuals for low–, medium–, and high–resolution displays.

Desktop system configurations are including more high–resolution monitors. The user must be able to discern any visuals that your application uses. The embedded base, however, still contains many standard VGA monitors. Your application's visuals must display well on these systems and should not appear overly large.

Recommended

Design any icons or graphics that your application displays to display on black–and–white and gray–scale monitors, as well on low–color (16) systems.

Recommended

Use icons to represent only objects and applications.

Recommended

Use only the palette of 22 colors for icons.

The CDE icon palette was chosen to maximize attractiveness and readability without using an unnecessary number of colors. Use of additional colors may cause undesirable color shifting on the display.

Recommended

Design icons for international use.

Do not use text, symbols, humor, animals, and other items that may be interpreted differently in other cultures.

Recommended

Left–allign 16x16 and 32x32 icons; place any empty bits on the right side of the bounding box.

Recommended

Center 48x48 icons in the bounding box.

Toolbars

Required

If you use a tool bar, it should appear only in windows with a menu bar.

Required

Tool bars should contain only those operations that are already available to the user in your application's menus. All items in a tool bar should be redundant.

Required

When an action represented by a tool bar icon is unavailable to the user, make that icon insensitive, with the associated stippled appearance. If a menu item is made insensitive, make the corresponding tool bar item insensitive as well.

Recommended

Give users the option to hide the tool bar.

Required

Place the tool bar container directly under the menu bar. It should have the same width as the window, as well as similar height to the menu bar.

Recommended

If you use a tool bar in your application, provide a status line in the same primary window as the tool bar.

This status line should provide immediate feedback to the user as to the purpose of the button that the mouse is currently over or that has the keyboard focus. When the arrow is over a tool bar icon, the status line should display a brief definition of what the icon represents or what will happen when the user clicks the icon.

Recommended

Provide labels under tool bar icons. These labels should serve to explain the purpose of the icon.

Recommended

All pixmaps in the tool bar should be the same size. This ensures that all the tool bar buttons are the same size.

Recommended

Drawn buttons in the tool bar should be the same width and height. Similar or related items should be grouped, and groups should be evenly spaced across the tool bar.

Expandable Windows

Recommended

The primary pane of the dialog box or window should contain all of the controls needed to complete the task. This should include all critical and frequently used functionality.

Recommended

It is assumed that infrequently used features are placed in the secondary pane. The core functionality of the application should not depend on any controls placed in secondary panes.

Required

Align command buttons along the bottom of the dialog box. When the window is expanded to show a secondary pane, then move the buttons to the bottom of the secondary pane.

Recommended

If important controls must be placed in the secondary pane, specify that the window in question should be displayed in its expanded state by default. The user should still be able to shrink the window by pressing the Contract button.

Recommended

The secondary pane should expand in the direction most consistent with the user's expectations, the reading pattern of the language in which it will be displayed, and the content of the information displayed.

Recommended

If possible, the panes should have the same default width.

Required

Separate the primary pane from the secondary pane with a separator.

Required

If a window is resizable, allocate any sizing changes to the pane that contains scrolling lists or text fields whose displayed length is less than their stored length. If both panes contain scrollable controls, distribute size changes evenly between the two panes. If neither pane contains scrollable controls, the window should not be resizable.

Required

The expandable window should have one button that changes its label based on the state of the window.

Required

The expand button should have two labels that reflect the two states of the expandable window accurately. The current label indicates to the user what will happen if the user clicks the button.

Examples of possible labels are Basic and Options, Expand and Contract, and More and Less.

Optional

The expand button may contain a graphic in addition to the label. This graphic should indicate the direction in which the window will expand or contract.

Recommended

Place the button in the lower left−hand corner of the window or dialog box for expansion in the vertical direction and in the lower−right hand corner for expansion in the horizontal direction.

Required

If the window or dialog box contains a scrolling list positioned to the far right side of the pane, do not align the drawn button with the scroll bar. For example, the button should be aligned with the list, not the scroll bar.

Required

Applications must remember the state of each window or dialog box (expanded or not expanded) independently (not collectively). The state should be changed only by the user and should always be preserved until explicitly altered by the user.

Recommended

Applications should remember the state of each expandable window or dialog box across sessions so that the user does not have to manually configure the expandable windows each time the application is run.

If appropriate, provide a mechanism (as an option) to allow the user to set the state of an expandable window globally for the application. This would be part of the application's Options.

Messages

Recommended

When creating messages, do not assume that the user has any expert knowledge about computer systems in general or the UNIX system in particular.

Recommended

Error messages should indicate the possible cause of the error and indicate the possible actions the user can take in response.

Optional

Use audio feedback, in addition to any messages displayed, to signal error conditions and events.

Optional

Avoid relying on error messages from the kernel and library routines. Error messages from kernel and library routines are normally not seen by the user, and even when the user does see them, they are usually too low−level and cryptic to be understood by nonprogrammers. Applications should check for error conditions and use an error dialog box to present an appropriate error message in terms of the user's actions and intentions.

Recommended

Display a confirmation or warning message dialog box when an action instigated by the user will be irreversible and potentially destructive.

Optional

Urgent conditions that require immediate attention by the user, no matter which application or desktop service the user is currently accessing should be brought to the user's attention with audiovisual notification. Invoke the alarm signal in the current workspace regardless of the workspace in which the application resides.

Recommended

Use footer messages only to communicate status, progress, or information (help) messages. Do not use the footer to present error messages.

Recommended

Provide a Help button in all message dialog boxes, except those that contain self−explanatory messages.

Recommended

Use the appropriate dialog box style to display messages.

Optional

Use an information dialog box to display status, completion of activity, or other informative types of messages to which the user need not necessarily respond other than to acknowledge having read the message.

Minimally, information dialog boxes should have an OK button so that the user can dismiss the dialog box. If there is additional information available or other references for the topic to which the message relates, include a Help button.

Optional

Use an error dialog box to display error messages to the user. State what the error is and specify why it occurred. Include a Help button so that the user can get additional information, unless the message is self−explanatory. Also include an OK button that dismisses the dialog box.

A Cancel button is not required for error dialog boxes unless the error resulted in the suspension of an activity that was in progress. In this case, the message should indicate whether the user has the option to continue the activity or to stop it, and the buttons for the dialog box should be Continue, Cancel, and Help. In general, error dialog boxes should not be modal unless it is critical that the user not continue interacting with the application until the user has acknowledged having read the error message.

Optional

Use a question dialog box to ask questions of the user. Clearly word the question to indicate what a Yes response or a No response means. Include the buttons Yes, No, and Help. Help should provide additional information as to what the application will do in response to a Yes or No choice.

Optional

Use a warning dialog box to communicate the consequences of an action requested by the user that may result in the loss of data, system or

application accessibility, or some other undesirable event. Present the dialog box before the action is performed and offer the user the opportunity to cancel the requested operation. Include the buttons Yes, No, and Help, or Continue, Cancel, and Help. Help should provide additional information on the consequences of performing the action requested.

Optional

Use a working dialog box to display in−progress information to the user when this information is not displayed in the footer of your application's window. Include a Stop button that allows the user to terminate the activity. The operation is terminated at the next appropriate breakpoint, and a confirmation might be displayed asking whether the user really wants to stop the activity. The confirmation message might state the consequences of stopping the action.

Optional

Write error messages to the CDE error log when it is not appropriate to display these to the user in a message dialog box, but when the message may nevertheless be useful in diagnosing problems.

Messages written to the error log should provide additional information about the error and should state the context in which the error occurred.

Optional

Informational messages should be left aligned and displayed in a light font in keeping with their unobtrusive nature. Note that the margin where informational messages are displayed should not accept mouse focus.

Optional

Progress messages should normally be displayed only while the operation is in progress. Remove notices and other information that is no longer valid within a few seconds to avoid confusion about whether or not the information is current.

Work−in−Progress Feedback

Recommended

If any command chosen by the user is expected to take longer than 2 seconds to complete, but less than 10 seconds, display the standard busy pointer as feedback that the command is executing.

Display the busy pointer within 0.5 seconds of execution of the command.

Recommended

If any command chosen by the user is expected to take longer than 10 seconds to complete, display a working dialog box or other feedback of similar character that indicates that the application is working on the request. The feedback should reveal progress toward completion of the activity.

If an activity is expected to take a significant amount of time (10 seconds or more), display feedback stronger than the busy pointer. Displaying the busy pointer for long amounts of time may lead the user to conclude that the application has "hung." Display a progress indicator in these scenarios that indicates that the application is still functioning and is working on the user's request. The progress indicator should show how much of the activity has been completed and what amount remains.

Recommended

When your application displays work−in−progress feedback to the user, do not block access to other applications and services within the desktop environment.

Multitasking should always be supported and, as such, your application should allow the user to access other services while it is busy performing some activity. Preferably, the user is also able to access other features within your application even though it is currently working on another request. When this is supported, your application should display an enhanced busy pointer that indicates that the application is busy but still willing to accept input.

Essential Related Topics

For more information, see the Control , Menu (Control) , Menu Bar (Menu Type) , Message , Object , View , and Window Navigation reference pages.

Supplemental Related Topics

For more information, see the Warning Signal reference page.

Area Adjust Click Technique

NAME

Area Adjust Click Technique –Reference

Description

The area adjust click technique is an adjustment selection technique in which an area of elements already identified as the current selection region can be adjusted by identifying an adjustment point.

Mouse–based area adjust click technique

Click the ADJUST button at the adjustment point.

Keyboard–based area adjust click technique

With the cursor at the adjustment point, press the `Shift Select`, `Shift Space` (unless in text), or `Ctrl Shift Space` keys.

When to Use

Recommended

Support the area adjust click technique as an alternative to the area adjust swipe technique when the area adjust swipe technique is too cumbersome to use.

Guidelines

Required

When using the area adjust click technique, and the adjustment policy is to reselect:

Determine the new selection region to be the area determined by the anchor point and the adjustment point.
Enlarge the new selection region to include the anchor element or anchor region, if indicated by the anchor inclusion policy.
Enlarge the new selection region, if indicated by the end–point inclusion policy, to include the element, if any, at the adjustment point.

Required

When using the area adjust click technique, and the adjustment policy is to enlarge, make the new selection region larger than the following:

The current selection region
The region determined using the reselect adjustment policy

Required

When using the area adjust click technique, and the adjustment policy is to balance:

Move (if necessary) the anchor point so that it is at the corner of the current selection region farthest from the adjustment point.
Determine the new selection region to be the area determined by the new anchor point and the adjustment point.
Enlarge the new selection region to include the anchor element or anchor region, if the anchor point is unchanged and if indicated by the anchor inclusion policy.
Enlarge the new selection region, if indicated by the end–point inclusion policy, to include the element, if any, at the adjustment point.

Required

If the current selection region does not include an anchor element, or if the anchor element is no longer in the selection region, identify the anchor element to be the element, if any, in the region nearest the anchor point.

Required

After a mouse−based area adjust click technique is used, place the active cursor as follows:

If it is a text or graphics cursor, at the point at which the ADJUST button was clicked

If the selection region contains one or more elements, on some element within the selection region

If neither of the above, on the currently cursored element in the selection scope

Essential Related Topics

For more information, see the **Adjustment Techniques** , Selection Policies , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Area Adjust Swipe Technique , Area Click Technique , Area Swipe Technique , Range Adjust Click Technique , Selection Models , Selection Modes , and Touch Adjust Click Technique reference pages.

Area Adjust Swipe Technique

NAME

Area Adjust Swipe Technique –Reference

Description

The area adjust swipe technique is an adjustment selection technique in which an area of elements already identified as the current selection region can be adjusted by identifying initial and final adjustment points in a single continuous action. The following describes the steps for mouse–based and keyboard–based techniques:

Mouse–based area adjust swipe technique

Press the ADJUST button at the initial adjustment point.
Move the mouse to the final adjustment point.
Release the ADJUST button at that point.

Keyboard–based area adjust swipe technique

With the cursor at the initial adjustment point, hold `Shift` down while pressing a navigation key to move the cursor to the final adjustment point.

When to Use

Required

Support the keyboard–based area adjust swipe technique when the last operation within the scope used an area–based selection technique.

Guidelines

Required

When using the area adjust swipe technique, and the adjustment policy is to reselect:

Determine the new selection region to be the area determined by the anchor point and the final adjustment point.
Enlarge the new selection region to include the anchor element or anchor region, if indicated by the anchor inclusion policy.
Enlarge the new selection region, if indicated by the end–point inclusion policy, to include the element, if any, at the final adjustment point.

Required

When using the area adjust swipe technique, and the adjustment policy is to enlarge, make the new selection region larger than the following:

The current selection region
The region determined using the reselect adjustment policy

Required

When using the area adjust swipe technique, and the adjustment policy is to balance:

When using the keyboard–based technique, and the final adjustment point is not within the current selection region, move (if necessary) the anchor point so that it is at the end of the current selection region farthest from either the initial or the final adjustment point. (The use of the final adjustment point is recommended.)
When using the keyboard–based technique, and the final adjustment point is within the current selection region, or when using the mouse–based technique, move (if necessary) the anchor point so that it is at the end of the current selection region farthest from the initial adjustment point.
Determine the new selection region to be the area determined by the new anchor point and the final adjustment point.
Enlarge the new selection region to include the anchor element or anchor region, if the anchor point is unchanged and if indicated by the anchor inclusion policy.

Enlarge the new selection region, if indicated by the end−point inclusion policy, to include the element, if any, at the final adjustment point.

Required

If the current selection region does not include an anchor element, or if the anchor element is no longer in the selection region, identify the anchor element to be the element, if any, in the region nearest the anchor point.

Required

After a mouse−based area adjust swipe technique is used, place the active cursor as follows:

If it is a text or graphics cursor, at the point at which the ADJUST button was released
If the selection region contains one or more elements, on some element within the selection region
If neither of the above, on the currently cursored element in the selection scope

Essential Related Topics

For more information, see the **Adjustment Techniques** , Selection Policies , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Area Adjust Click Technique , Area Click Technique , Area Swipe Technique , Range Adjust Swipe Technique , Selection Models , Selection Modes , and Touch Adjust Swipe Technique reference pages.

Area Click Technique

NAME

Area Click Technique –Reference

Description

The area click technique is a group selection technique in which a rectangular area of elements can be selected or deselected by indicating each corner of the area in discrete actions. The following describes the steps for mouse–based and keyboard–based techniques:

Mouse–based area click technique
: Click the SELECT button at the initial corner of a rectangular area.
Move the mouse to the opposite corner of the area.
Click the ADJUST button at that opposite corner.

Mouse–based area click technique, forcing toggle mode
: Click `Ctrl` SELECT at the initial corner of a rectangular area.
Move the mouse to the opposite corner of the area.
Click the ADJUST button at that opposite corner.

Keyboard–based area click technique
: Press `Select`, `Space` (unless in text) or `Ctrl Space` with the cursor at the initial corner of a rectangular area.
Use keyboard navigation to move the cursor to the opposite corner of the area.
Press `Shift Select`, `Shift Space` (unless in text) or `Ctrl Shift Space`.

When to Use

Required

Support the area click technique only when a common selection in the scope consists of a set of spatially adjacent elements.

Required

Support the area click technique only when discontiguous selections are allowed.

Required

Support the keyboard–based area click technique in normal mode only when a text cursor is used for all navigation between the two corners of the area. The result of attempted use in other cases is undefined.

Guidelines

Required

Define the current selection region to consist of the specified area as follows:
: Enlarged to include the element, if any, on which the technique was initiated
Enlarged, if indicated by the end–point inclusion policy, to include the element, if any, on which the technique finished

Mouse–Based Techniques

Required

After a mouse–based area click technique is used:
: Identify the anchor point to be the point at which the SELECT button was clicked.
Identify the anchor element to be the element, if any, within the selection region nearest to the anchor point.

Required

After a mouse−based area click technique, place the active cursor as follows:

At the point at which the ADJUST button was clicked, if it can be placed there

If using an element cursor, on some element within the region

If there are no elements in the region, where the cursor previously was in the scope

Required

When using the area click technique in select mode:

Select all the elements in the selection region.

Deselect all other elements in the scope.

Required

If currently in select mode, clicking `Ctrl` SELECT instead of SELECT causes the area click technique to act as if toggle mode were in use.

Required

When using the area click technique in toggle mode:

Toggle all elements in the selection region, based on the toggling policy.

Do not change the selection state of the other elements in the scope.

Keyboard−Based Techniques

Required

After a keyboard−based area click technique is used:

Identify the anchor point to be the point at which `Select`, `Space` (unless in text), or `Ctrl Space` was pressed.

Identify the anchor element to be the element, if any, within the selection region nearest to the anchor point.

Required

When using the keyboard−based area click technique in normal mode:

Select all the elements in the selection region.

Deselect all other elements in the scope.

Required

When using the keyboard−based area click technique in add mode:

Toggle all elements in the selection region, based on the toggling policy.

Do not change the selection state of the other elements in the scope.

Essential Related Topics

For more information, see the **Adjustment Techniques** , Selection Policies , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Area Adjust Click Technique , Area Click Technique , Area Swipe Technique , Range Adjust Swipe Technique , Selection Models , Selection Modes , and Touch Adjust Swipe Technique reference pages.

Area Swipe Technique

NAME

Area Swipe Technique –Reference

Description

The area swipe technique is a group selection technique in which a rectangular area of elements can be selected or deselected by indicating the corners of the area in a single continuous action. The following describes the steps for mouse–based and keyboard–based techniques:

Mouse–based area swipe technique
: Press the SELECT button at the initial corner of a rectangular area.
Move the mouse to the opposite corner of the area.
Release the SELECT button at that opposite corner.

Mouse–based area swipe technique, forcing toggle mode
: Press `Ctrl` SELECT at the initial corner of a rectangular area.
Move the mouse to the opposite corner of the area.
Release the SELECT button at that opposite corner.

Keyboard–based area swipe technique

: Hold `Shift` down while pressing a navigation key. The area is determined by the initial and final cursor positions.

When to Use

Required

: Support the area swipe technique only when a common selection in the scope consists of a set of spatially adjacent elements.

Required

: Support the area swipe technique only when discontiguous selections are allowed.

Guidelines

Required

: Define the current selection region to consist of the specified area as follows:
Enlarged, if indicated by the end–point inclusion policy, to include the element, if any, at the initial corner of the area
Enlarged, if indicated by the end–point inclusion policy, to include the element, if any, at the opposite corner of the area

Mouse–Based Techniques

Required

: After a mouse–based area swipe technique is used:
Identify the anchor point to be the point at which the SELECT button was pressed.
Identify the anchor element to be the element, if any, within the selection region nearest to the anchor point.

Required

: After a mouse–based area swipe technique, place the active cursor as follows:
At the point at which the SELECT button was released, if it can be placed there
If using an element cursor, on some element within the region
If there are no elements in the region, where the cursor previously was in the scope

Required

When using the area swipe technique in select mode:
Select all the elements in the selection region.
Deselect all other elements in the scope.

Required

If currently in select mode, pressing `Ctrl` SELECT instead of SELECT causes the area swipe technique to act as if toggle mode were in use.

Required

When using the area swipe technique in toggle mode:
Toggle all elements in the selection region, based on the toggling policy.
Do not change the selection state of the other elements.

Keyboard−Based Techniques

Required

After a keyboard−based area swipe technique is used:
Define the current selection region to consist of the identified range.
Identify the anchor point to be the initial point at which the `Shift` navigation key was pressed.
Identify the anchor element to be the element, if any, within the selection region nearest to the anchor point.

Required

When using the keyboard−based area swipe technique in normal mode:
Select all the elements in the indicated area.
Deselect all other elements in the scope.

Required

When using the keyboard−based range swipe technique in add mode:
Toggle all elements in the indicated range, based on the toggling policy.
Do not change the selection state of the other elements in the scope.

Essential Related Topics

For more information, see the **Selection Modes** , Selection Policies , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Area Adjust Click Technique , Area Adjust Swipe Technique , Area Click Technique , Range Swipe Technique , Selection Models , and Touch Swipe Technique reference pages.

Browse Technique

NAME

Browse Technique –Reference

Description

The browse technique is an individual selection technique in which the last element touched by the pointer or cursor is selected or deselected. The following describes the steps for mouse–based and keyboard–based techniques:

Mouse–based browse technique
: Press the SELECT button anywhere within the scope.
Release the SELECT button on an element in the scope.

Keyboard–based browse technique

: Use the navigation keys to move a cursor to an element in the scope.

When to Use

Required

: Support the mouse–based browse technique only in select mode, and only when at most one element in a scope can be selected.

Required

: Support the mouse–based browse technique only in normal mode, and only when an element cursor is being used.

Guidelines

Mouse–Based Browse Technique

Required

: When using the browse technique in select mode, releasing the SELECT button on a selectable element does the following:
Selects that element
Deselects all other elements in the scope
Places the active cursor on that element

Recommended

: When using the browse technique, moving the pointer over a selectable element while pressing the SELECT button provides additional feedback about the element.

Keyboard–Based Browse Technique

Required

: Using a navigation key to move the active cursor within a selection scope in normal mode that results in an element cursor being placed on a selectable element does the following:
Selects that element
Deselects all other elements in the scope
Identifies that element as an anchor element for later use in adjustment of the selection
Defines the current selection region to consist of the area taken up by that element

Required

Using a navigation key to move the active cursor within a selection scope in normal mode that results in an element cursor being placed on a selectable element identifies a point as an anchor point for later use in adjustment of the selection. That point is one of the following:

- The pointer position (mouse–based technique)
- The cursor position (keyboard–based technique with a text or graphics cursor)
- The center of the element (keyboard–based technique with an element cursor)

Essential Related Topics

For more information, see the **Adjustment Techniques** , Selection Modes , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Point Technique , Selection Models , and Selection Policies reference pages.

Cancel (Action and Choice)

NAME

Cancel (Action and Choice) –Reference

Description

Cancel includes the following definitions:

- An action that terminates the current task or user interaction or exits from a special mode and restores the application state, if possible, to that preceding the start of the task, user interaction, or interface mode.
- An action choice that invokes the cancel action.

When to Use

Recommended

Support a cancel action if a task can be interrupted by the user or the operating environment before irreversible changes are made.

Recommended

Support a cancel action if the user enters a mode that would normally terminate with an irreversible change.

Recommended

Support a cancel action along every step of a dialog before the final committing step.

Recommended

Provide a Cancel choice to close secondary windows other than in–progress messages.

Required

Do not provide a Cancel choice within an in–progress message. Provide a Close or Stop choice instead.

Guidelines

Required

If the user invokes the cancel action while a requested task is in progress and is waiting for some input, stop the task and, if possible, undo or do not apply any changes.

Required

If the user invokes the cancel action while a requested task is in progress but is not waiting for some input, stop the task and confirm the cancellation with a warning message, unless the operating environment specifies that confirmation is not necessary.

Required

If the user invokes the cancel action while some multistep dialog is in progress, terminate the dialog, and do not continue the task that initiated the dialog. If possible, undo any changes to the application state that may have been made during the dialog.

Required

If the user invokes the cancel action while some modal interaction is in progress, stop the current interaction, exit from the mode, return to the normal mode of operation specified by the operating environment, and do not apply any changes.

Shortcut Keys

Required

Make `Cancel` the shortcut key for the Cancel choice.

Optional

Make `Stop` the shortcut key for the Cancel choice.

Essential Related Topics

For more information, see the **Close (Choice)** reference page.

Supplemental Related Topics

For more information, see the Action Message and Direct Manipulation reference pages.

Cascading (Choice Type)

NAME

Cascading (Choice Type) -Reference

Description

A cascading choice is a type of choice that, when activated, displays a cascaded control.

When to Use

Required

Use a cascading choice to display a cascaded control.

Guidelines

Activation and Unavailability

Required

When a user activates a cascading choice, display its associated cascaded control, even when all choices and controls in the cascaded control are unavailable.

Required

Do not display unavailable emphasis on cascading choices.

Pressing a Cascading Choice

Required

If a cascading choice is used to display a spring−loaded control, pressing (but not clicking) the SELECT button on the choice activates it and displays the control if it is not already displayed.

Recommended

If a cascading choice is used to display a menu, then pressing (but not clicking) the MENU button on the control activates it and displays the menu if it is not already displayed.

Releasing a Cascading Choice

Required

If the SELECT or MENU button is pressed, and the pointer is over a cascading choice whose associated cascaded control is not displayed (for example, because the mouse button was pressed elsewhere and moved to the cascading choice), then releasing the mouse button has no effect.

Required

Releasing the SELECT button over a cascading choice whose associated cascaded control is currently displayed leaves the cascaded control displayed if it was not displayed when the SELECT button was pressed.

Required

Releasing the MENU button over a cascading choice whose associated cascaded menu is currently displayed leaves the cascaded control

displayed if it was not displayed when MENU was pressed.

Clicking to Display a Choice

Required

Clicking the SELECT button on a cascading choice activates it if its associated cascaded control is not currently displayed.

Required

If activating a cascading choice displays a menu, then clicking the MENU button on the choice activates it if the menu is not currently displayed.

Removing a Cascaded Control

Required

Clicking the SELECT button on a cascading choice removes the associated cascaded control if it is currently displayed.

Required

If a menu is associated with a cascading choice, then clicking the MENU button on the choice removes the associated cascaded menu if it is currently displayed.

Required

Removing a control removes all of its cascaded descendants.

Essential Related Topics

For more information, see the **Choice** , Menu (Control) , and Spring–Loaded (Control Type) reference pages.

Supplemental Related Topics

For more information, see the Emphasis (Cue) reference page.

Change View (Choice)

NAME

Change View (Choice) –Reference

Description

Change View is a cascading choice that displays a menu from which the user can choose the view to be presented within the window.

When to Use

Required

Provide a Change View choice in the View menu if your application allows a user to switch among three or more views.

Guidelines

Required

Make Change View a cascading choice; its cascaded menu should list the available views.

Required

Indicate the current view active in the window by displaying that view's choice in the cascaded menu with unavailable emphasis and, optionally, by including a check graphic next to it.

Required

Do not open a new window as a result of activating Change View; instead, display the new view of the element in the current window.

Required

Do not change the underlying data being viewed when the user changes the view.

Essential Related Topics

For more information, see the View and View Menu reference pages.

Supplemental Related Topics

For more information, see the New Window (Choice) reference page.

Check Box (Control)

NAME

Check Box (Control) –Reference

Description

A check box is a control for displaying a value choice that has two clearly distinguishable states and may have a third, indeterminate state. If a selection contains elements that have two different states (for example, a text selection containing both bold and normal text), the check boxes can show an indeterminate state. You typically use check boxes in a group to provide a tab group of alternatives or to set options in an application. Check boxes are also known as check buttons.

Figure 1 Check Box

When to Use

Required

Use a check box to display an individual value choice that can be set to on or off.

Required

Use a group of check boxes for value choices that are not mutually exclusive and can each be set to on or off.

Recommended

Use a check box only if the user will clearly understand the meaning of the choice when it is set or unset. For example, provide a check box next to the Bold value choice to indicate that the selected text is rendered in a bold font.

Recommended

When a value choice is displayed in a menu other than an option menu and is part of a group of choices that are not mutually exclusive, use a check graphic to the left of the value choice to indicate whether it is set or unset. For information on bidirectional and vertical language support, see Chapter 11.

Guidelines

Recommended

When a user activates a check box from a group of check boxes, do not change the state of any other check boxes within the group.

Optional

Setting one check box in a group can override remaining check boxes; when the first check box is set, make the others unavailable. For example, a group of check boxes could have the labels Plain, Bold, Italic, and Underline. Setting Plain unsets the other check boxes.

Required

When a check box is initially displayed in a set or unset state, toggle only between those two values.

Required

When a check box is displayed in an indeterminate state, toggle from indeterminate to set to unset.

Selection Properties

Required

When a check box represents a property shared by more than one selected element:

Show it as set if all of the selected elements have that property turned on. For example, if all of the text in a text selection is bold, and a check box is used to indicate whether the selected text is bold, show the check box as set.

Show it as unset if none of the selected elements have that property turned on. For example, if none of the text in a text selection is bold, show the check box as unset.

Show it as indeterminate if some, but not all, of the selected elements have that property turned on. For example, if only some of the text in a text selection is bold, show the check box as indeterminate.

Required

When a check box represents a property shared by more than one selected element, toggling it to a particular state has the following effect:

Toggling it to set ensures that all of the selected elements have that property. For example, toggling the Bold check box to set makes all the selected text bold.

Toggling it to unset ensures that all of the selected elements do not have that property. For example, toggling the Bold check box to unset unbolds all the selected text.

Toggling it to indeterminate gives each selected element the value for the property it had when the check box was displayed as indeterminate.

Visuals

Required

A check box consists of a check graphic and a label. The check graphic is displayed to the left of the label. For information on bidirectional and vertical language support, see Chapter 11.

Recommended

When set, the check graphic is one of the following:

Checkmark
Cross
Box
Checkmark in a box
Cross in a box

When unset, the check graphic is empty or an empty box.

Recommended

The indeterminate state of the checkbox is represented by shading or otherwise diminishing the contrast between the check graphic in the on state and the background.

Recommended

If the box is to be filled, fill the box with an appropriate color different from the foreground and background color on displays that support multiple colors.

Recommended

If a mark is displayed in the check box, it is displayed in the foreground color on displays that support multiple colors.

Essential Related Topics

For more information, see the **Control** reference page.

Supplemental Related Topics

For more information, see the Choice and Radio Button (Control) reference pages.

Choice

NAME

Choice -Reference

Description

A choice is an alternative, displayed as a label (text or graphics) on the screen that a user can choose. Choices are available via controls through which the user chooses values or invokes actions. Choices can be represented by push buttons, by menu items, or by check boxes. There are four types of choices:

- Action choices
- Dialog choices
- Cascading choices
- Value choices

Action, dialog, and cascading choices are activated. Value choices are toggled.

When to Use

Required

Provide a choice for every action that the user can invoke or for every value that the user can set or unset.

Guidelines

Labels and Naming

Required

Use the predefined label for each predefined choice, except when Yes, No, Apply, and OK are replaced by more descriptive labels.

Recommended

If a user changes the text label of a choice, reflect this change in the Help system.

Recommended

Use verbs to describe what occurs when the user activates an action or dialog choice.

Recommended

Identify a choice with a graphic, text, or both, depending on which best identifies the choice. For example, for a drawing application, use graphic patterns rather than text to represent the choices.

Recommended

Dynamically add text to a choice to make the meaning of that choice clearer in a given context. For example, changing the name of the Undo choice to Undo Typing clarifies the meaning and differentiates the choice from Undo Delete or Undo Apply.

Recommended

Do not change the function or label of a choice because the context changes and the functions are significantly different or opposite from one another, except for the labels of the OK or Apply button. Instead, provide different choices that may sometimes be unavailable as the state changes. This allows the user to see all of the actions that are available.

For example, do not provide a push button labeled Pause/Resume (or a push button whose label changes from Pause to Resume) that would have the function of either pausing or resuming a process, depending on the current state. Instead, provide one push button labeled

Pause and another push button labeled Resume and display unavailable emphasis on the one that is not available in the current context.

Optional

You may change the label and function of a choice as the state changes if the functions are close variants of one another. For example, on the Edit menu, switch between Repeat and Redo as the state changes.

Layout

Recommended

Arrange related choices of the same type next to one another in a group; for example, arrange related check boxes in rows or columns, and use spacing, a separator, or a group box to separate the group from other controls.

Behavior

Required

Pressing the SELECT button on a choice and then releasing the SELECT button on the same choice activates or toggles it.

Required

When focus is on an element that represents a choice, pressing `Select` or `Space` activates or toggles it.

Essential Related Topics

For more information, see the **Action (Choice Type)** , Cascading (Choice Type) , Control , Dialog (Choice Type) , Menu (Control) , and Value (Choice Type) reference pages.

Supplemental Related Topics

For more information, see the Information and Message Areas reference page.

Clear (Action Choice)

NAME

Clear (Action Choice) -Reference

Description

The Clear choice includes the following:

Clear

An action choice that removes selected elements without compressing the visible space they occupied.

Clear to Trash

An action choice that removes selected elements (generally objects) without compressing the visible space they occupied and puts them into the trash.

When to Use

Recommended

Provide a Clear choice to allow a user to remove an element or group of elements from a window without compressing the visible space the elements occupied. For example, provide a Clear choice to allow a user to remove information from selected rows and columns of a spreadsheet without removing the rows and columns that contained the information.

Guidelines

Required

Include the Clear choice in a pop−up menu on an unselected element only if it can be removed without compressing the visible state it occupied.

Required

Include the Clear choice in a pop−up menu on a selected element only if the selected elements can be removed without compressing the visible space it occupied.

Required

If only objects are selected, the Clear choice must be unavailable in the Edit menu; use Clear to Trash on the Selected menu instead. If the selection scope includes only objects, then Clear must not be in the Edit menu at all.

Essential Related Topics

For more information, see the Delete (Action Choice) , Edit (Menu) , and Selected Menu reference pages.

Supplemental Related Topics

For more information, see the Menu (Control) reference page.

Client Area

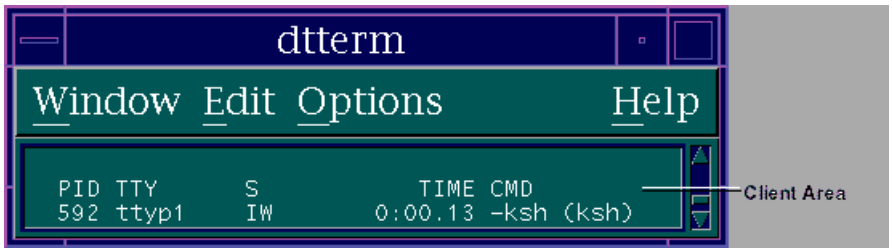
NAME

Client Area –Reference

Description

The client area is the application content area within a window.

Figure 1 Client Area



When to Use

Required

Include a client area when designing primary and secondary windows, dialog boxes, and pop–up menus.

Guidelines

Required

Place the client area directly beneath the window menu. For information on designing applications for vertical languages, see Chapter 11.

Required

Include scroll bars when the client area is too large to be completely displayed in the window.

Recommended

Include a command area, message area, window menu, and scroll bars when building client areas.

Recommended

Provide split bars in a client area that is too large to be completely displayed in the window.

Recommended

Scale the client area's contents with the window size borders.

Essential Related Topics

For more information, see Chapter 10 and the **Primary Window** , Secondary Window , View , and Viewing Area (Control) reference pages.

Supplemental Related Topics

For more information, see the Palette Area (Area) , Paned Box (Control) , Window Frame , and Window Menu reference pages.

Clipboard

NAME

Clipboard -Reference

Description

The clipboard allows the user to transfer a selection from a source to the clipboard and then subsequently from the clipboard to a destination. Clipboard transfer includes the following choices:

Copy

An action choice that copies selected elements to the clipboard.

Copy Link

An action choice that places a link to selected elements on the clipboard.

Copy Special

A dialog choice that displays a dialog through which a user can choose the formats in which to copy selected elements to the clipboard.

Cut

An action choice that removes selected elements and places them onto the clipboard. The space they occupied is usually filled by the remaining elements in the scope of selection.

Paste

An action choice that places the contents of the clipboard into a specified location or into an object or objects displayed within the window.

Paste Link

An action choice that pastes a link to elements previously placed on the clipboard into a specified location or into an object or objects displayed within the window.

Paste Special

A dialog choice that displays a dialog through which a user can choose the format in which elements previously placed on the clipboard are pasted into a specified location or into an object or objects displayed within the window.

When to Use

Required

All selection scopes support transfer to the clipboard.

Required

Support clipboard transfer in any selection scope in which the current selection can be deleted or cut.

Required

If a control supports text entry or any form of data transfer into the control, it supports Paste.

Guidelines

Required

The source of a Copy, Copy Link, Copy Special, or Cut operation are the selected elements of the control for which the operation was invoked.

Required

The target of a Copy, Copy Link, Copy Special, or Cut operation is the clipboard.

Required

Cut is a move operation, Copy is a copy operation, Copy Link is a link operation. Copy Special results in either a copy or link operation, depending on the format chosen in the dialog.

Required

If Paste or Paste Special transfers a data format, it is a copy operation whose source is the clipboard.

Required

If Paste, Paste Special, or Paste Link transfers a link format, it is a link operation whose source is that of the operation that placed the link on the clipboard.

Required

The target of Paste, Paste Link, or Paste Special is the control for which the action was invoked.

Recommended

When using Copy or Cut, place as many formats of the transferred data as is practical on the clipboard.

Availability

Required

Do not make any Paste choices available when no data is in the clipboard or when the data cannot be pasted.

Required

If you provide Paste Special, do not provide Paste Link (the equivalent functionality is available via the Paste Special Dialog).

Required

If you provide Copy Special, do not provide Copy Link (the equivalent functionality is available via the Copy Special Dialog).

Essential Related Topics

For more information, see the **Edit (Menu)** reference page.

Close (Choice)

NAME

Close (Choice) -Reference

Description

Close is an action choice that removes a window and all of the windows dependent on it from a screen.

When to Use

Required

Provide the Close choice in the window menu of every window.

Recommended

In a secondary window, use the Close choice as a push button instead of Cancel if Close would be more appropriate or less ambiguous.

Recommended

Provide a Close choice in the File menu if the window can be closed independently of other primary windows.

Guidelines

Recommended

Provide the Close choice as a push button in a secondary window if the window is used only to present status information to the user.

Recommended

If the effect of activating a choice or choices in a secondary window is irreversible, then the Close choice can replace the Cancel choice after the first irreversible action.

Recommended

When closing a window, save its state (including its position, size, dependent secondary windows, and associated messages that are currently displayed) so that they can be restored if the window is opened at a later time.

Behavior

Required

When a user activates the Close choice for the last primary window displayed for an application, remove all dependent secondary windows and exit from the application.

Required

When a user activates the Close choice for a window, remove the window and any dependent secondary windows.

Required

Use a Close push button to close an in−progress message window without affecting the task in progress. Do not terminate the task when a user presses the Close push button in an in−progress message window.

Confirmation on Close

Required

If information in a primary window could be lost when the user activates the Close choice, display a warning message.

Recommended

If information in a secondary window could be lost when the user activates the Close choice, do not display a warning message.

Essential Related Topics

For more information, see the **Exit (Choice)** , File Menu , Push Button (Predefined) , and Window Menu reference pages.

Supplemental Related Topics

For more information see the Menu Guidelines reference page.

Combination Box (Control)

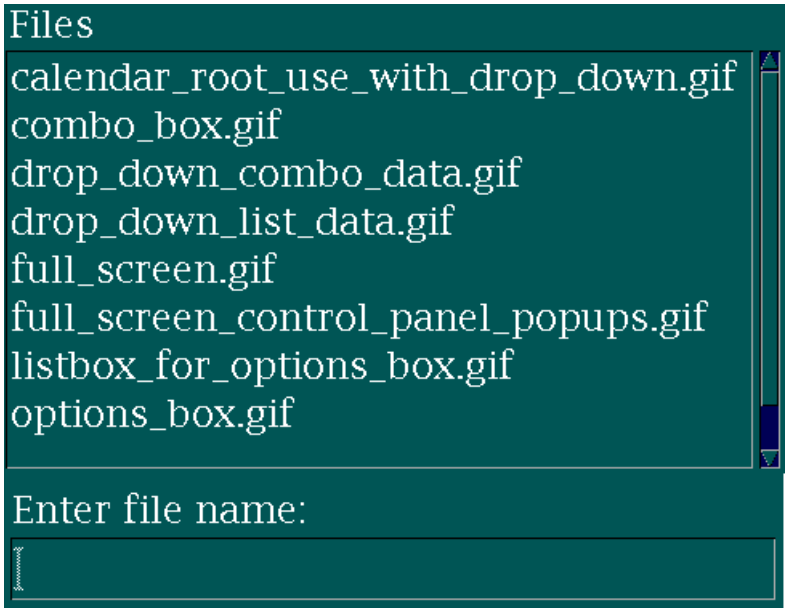
NAME

Combination Box (Control) –Reference

Description

A combination box is a combination text–list control in which both the text field and the list box are displayed permanently and in which the text field can be used for text entry. Combination Box is an example of a combination box.

Figure 1 Combination Box



When to Use

Recommended

Provide a combination box when the user can either choose from a set of commonly used predefined items in the list box for replacing the contents of the text field or enter values into a text–entry field that are not in the list box.

Recommended

Do not use a combination box if the window in which the combination box is displayed is sized so that two items cannot be displayed in the list box. Instead, use a drop–down combination box.

Guidelines

Navigation

Required

Do not allow the list box in a combination box to take focus.

Essential Related Topics

For more information, see the Combination Text–List Control (Control Type), Control, Drop–Down Combination Box (Control), Drop–Down List (Control), List Box (Control), Option Menu (Menu Type), Scroll Bar (Control), and Text–Entry Field (Control) reference pages.

Combination Text−List Control (Control Type)

NAME

Combination Text−List Control (Control Type) -Reference

Description

A combination text−list control is a control that combines the functions of a single−line text field (either a text−display field or a text−entry field) and a list box. A combination text−list control may also contain a list cascade button that is used to display the list box if it is not displayed. This guide defines the following combination text−list controls: command box, combination box, drop−down combination box, drop−down list, and selection box.

When to Use

Recommended

Provide a combination text−list control when the user can either choose a predefined item from the items in the list box for placement into a text field or enter values into a text−entry field that are not in the list box.

Recommended

Provide a combination text−list control when the user might want to type values into the text−entry field and the list of items in the list box can provide contextual support or otherwise assist the user in completing the text−entry field.

Recommended

Provide a combination text−list control when the user wants to easily see the last value chosen from a list.

Guidelines

Required

Order the items in the list box in some application−defined way that is meaningful to the user, such as alphabetically, numerically, chronologically, or other such order.

Recommended

Initially display one of the text items of the list box in the text field if a meaningful, application−specific one can be selected. Otherwise display nothing.

Selection

Required

Use either a single or a browse selection model in the list box.

Required

Use the browse selection model in the list box if a text−entry field is used and the list box cannot take focus.

Required

Use the browse selection model in the list box if prefix completion is in use.

Required

Move input focus to the text field if the user presses the SELECT button on an item in the list box and the list box cannot take focus. The item indicated is selected as usual, however.

Navigation

Required

If the user presses [darr] when focus is in the text field and the list box is not displayed, do one of the following:

Display the list.

Navigate down the list, select the cursored list item, and display it in the text field without displaying the list.

Required

If the user presses keys used for vertical navigation or paging when focus is in the text field and the list is not displayed, do one of the following:

If [darr] would display the list, either display the list or do nothing.

Navigate in the list as indicated, select the cursored list item, and display it in the text field without displaying the list.

Required

If the user presses a key used for vertical navigation or paging when focus is in the text field and the list box is displayed, navigate in the list as indicated, select the cursored list item, and display it in the text field.

Required

A combination text–list control whose list box cannot take focus is a tab group.

Required

When focus is in a text–list control whose list box is temporarily displayed and keyboard navigation (for example, tab group or window navigation) is used to move the focus elsewhere, the list box is removed if currently displayed.

Recommended

Display the list box when the user presses Alt [darr] if the text field has focus and the list box is not currently displayed.

Character Entry

Required

When a text–entry field in a combination text–list control has focus, input of printing characters enters them into the text field, with or without prefix navigation.

Recommended

When a text–display field in a combination text–list control has focus, use entered printing characters either for prefix completion or for first–letter cursor navigation in the associated list.

Behavior

Required

When a list item is selected, other than by prefix navigation, place the text of the newly selected list item in the text field.

Required

When a newly selected list item is placed in the text field and is selected in the text field, place the cursor at the end of the text field.

Recommended

When a newly selected list item is placed in the text field, other than by prefix completion, select it in the text field as a persistent selection.

Recommended

When scrolling the text field horizontally, scroll the associated list horizontally in conjunction with it.

Visuals

Required

Provide vertical scroll bars if some of the items in the list box are not displayed when the list box is displayed.

Required

Make the text field at least as wide as the list box would be if it were displayed.

Recommended

If the window in which the combination text–list control is displayed is sized so that even two items cannot be displayed in the list box, do not display the list box permanently.

Recommended

If the items in the list box are wider than the list box, support horizontal scrolling in the list box.

Visuals When the List Box is Not Permanently Displayed

Required

If the list box is not always displayed, include a list cascade element from which the list box can be displayed.

Recommended

If enough space is not available to display the text field and the list cascade button, clip the text field before clipping the list cascade button horizontally.

Recommended

Use a down arrow graphic placed to the right of the text field, and visually distinct from it, as the list cascade button. For bidirectional and vertical language guidelines, see Chapter 11.

Visuals When the List Box is Permanently Displayed

Recommended

Make the text field the same width as the list box if the list box is permanently displayed.

Recommended

When a user increases the size of the window in which the combination text–list control is displayed, increase the number of items displayed in the list box.

Recommended

When a user decreases the size of the window in which the combination text–list control is displayed, decrease the number of items displayed in the list box.

Recommended

Make the list box large enough to display at least two items.

Essential Related Topics

For more information, see the **Control** , Drop–Down Combination Box (Control) , Drop–Down List (Control) , List Box (Control) , Option Menu (Menu Type) , Scroll Bar (Control) , Selection Box (Control) , and Text–Entry Field (Control) reference pages.

Command Area (Area)

NAME

Command Area (Area) –Reference

Description

A command area is an area in a window that provides a place for users to enter commands. Command Area illustrates a command area.

Figure 1 Command Area

When to Use

Recommended

Provide a command area for entering commands if a task analysis of the user's tasks shows that the user will need to enter commands.

Recommended

Use a command area with a text–entry field if commands are simple textual strings or are unique and not usually repeated.

Recommended

Use a command area with a command box if the user could use a history list of past commands for repetition or for composing a new command.

Recommended

Use a command area consisting of push buttons if the user will issue commands from only a small set (from two to ten) of fixed commands.

Recommended

Do not use a command area consisting of push buttons if the user will issue commands from a medium–sized set (from ten to twenty) of commands. Instead, place a critical subset in the command area as push buttons and place the rest in a palette area or in menus.

Guidelines

Required

If you include a command area, provide a text–entry field or a command box or a group of push buttons or both for common actions.

Required

Do not scroll the command area when the viewing areas of the window are scrolled.

Layout

Recommended

If you include a command area, design it to run from border to border across the width of the window.

Recommended

If you include a command area in a window that has an information area, place the command area just above the information area. Otherwise, place the command area at the bottom of the window.

Essential Related Topics

For more information, see the Command Box (Control) and Push Button (Control) reference pages.

Supplemental Related Topics

For more information, see the Information and Message Areas (Area) reference page.

Command Box (Control)

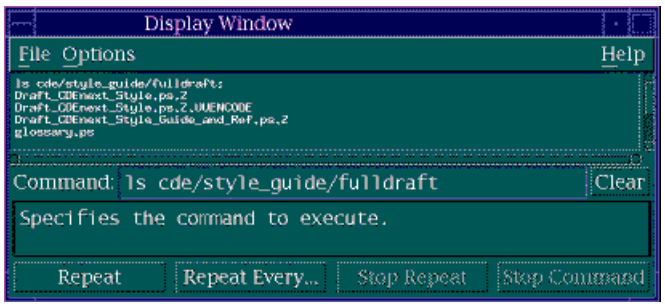
NAME

Command Box (Control) –Reference

Description

A command box is a combination text–list control in which the text field can be used for entry of commands. A command box (and related push buttons) is contained in a secondary window called a command dialog.

Figure 1 Command Box



When to Use

Recommended

Provide a command box to allow users to review previous commands and either to choose a previous command or to compose a new command.

Guidelines

Required

Either display the list box permanently or do not display it at all.

Behavior

Required

When the user invokes the default action of the command box, issue the command in the text field of the command box.

Required

If the command is issued successfully, add the command to the end of the list box.

Required

If the last command in the list box is selected, and a successfully issued command adds the new command to the end of the list, select the new command.

Required

If the command failed, issue a warning signal but do not add the command to the end of the list box, and leave the command unchanged in the text field.

Push Buttons in Command Dialogs

Recommended

In a command dialog, provide an OK push button that invokes the default action of the command box and then, if the command succeeds, remove the command box.

Recommended

In a command dialog, provide a Cancel push button that removes the command box but does not issue any command.

Recommended

In a command dialog, provide an Apply push button that invokes the default action of the command box.

Recommended

In a command dialog, provide a Help push button that displays a secondary window containing context−sensitive help information.

Essential Related Topics

For more information, see the **Command Area (Control)** and Push Button (Control) reference pages.

Supplemental Related Topics

For more information, see the Information and Message Areas reference page.

Component Activation (CDE)

NAME

Component Activation (CDE) -Reference

Description

Component activation refers to using controls to perform actions in CDE, such as initiating an action choice.

Guidelines

Basic Activation

Required

Pressing the SELECT button activates a push button.

Required

When a push button has focus, pressing `Select` or `Spacebar` activates it.

Required

When an activatable menu entry has the focus, pressing `Select`, `Spacebar`, `Enter`, or `Return` activates the entry.

Required

When the SELECT button is pressed over a push button, change the appearance of the push button to indicate that releasing the SELECT button will activate the push button. If, while the SELECT button is pressed, the pointer is moved outside of the push button, restore the visual state. If, while the SELECT button is still pressed, move the pointer back inside of the push button and change the visual state again to indicate the pending activation. If the SELECT button is pressed and released within a push button, activate the button, regardless of whether the pointer has moved out of the push button while it was pressed.

Required

If a selectable element of a collection is activatable, clicking the SELECT button, pressing `Select`, or pressing `Spacebar` (except in text) selects it. Double−clicking the SELECT button selects and activates it.

Required

The time allowed to detect a double−click *(**doubleClickTime: 500)* should be no less than 500 milliseconds.

Accelerators (Shortcut Keys)

Required

If your application uses accelerators (shortcut keys), the component with the accelerator should display the accelerator key or key combination following the label of the component.

Required

If a button with an accelerator is within a primary or secondary window, or within a pull−down menu from its menu bar, make it activatable whenever the input focus is in the window or the menu−bar system. If a button with an accelerator is within a pop−up menu, make it activatable whenever the focus is in the pop−up menu or the component with the pop−up menu.

An accelerator must be activatable from the window or component associated with the accelerator.

Mnemonics

Required

If your application uses mnemonics, the label for the component with the mnemonic should contain the character that is its mnemonic. If the label does not naturally contain the character, place the mnemonic in parentheses following the label.

Required

Mnemonic characters must be chosen for ease−of−location within the text of a label. Wherever possible, use the first character of the label. If that is not possible, use the last character of the label or, if there is more than one word, the first character of the second word. After that, go through the label from the second character on until a unique mnemonic is found.

Required

Make all mnemonics case insensitive for activation.

Required

When the location cursor is within a menu or a menu bar, pressing the mnemonic key of a component within that menu or menu bar moves the location cursor to the component and activates it. If a mnemonic is used for an option button or for a cascading button in a menu bar, pressing `Alt` and the mnemonic anywhere in the window or its menus moves the cursor to the component with that mnemonic and activates it.

Tear−off Activation

Required

When the user activates a tear−off choice, tear off the menu that contains the choice.

Required

When a menu with a tear−off choice is posted, pressing the TRANSFER button in the tear−off choice starts a tear−off action. As long as the TRANSFER button is held, a representation of the menu should follow the movement of the pointer. Releasing the TRANSFER button ends the tear−off action by unposting the menu system, creating a new window at the current pointer location that contains the contents of the menu, and giving focus to the new window in explicit pointer mode.

Help Activation

Required

When the user presses `Help` on a component. invoke any context−sensitive help for the component or its nearest ancestor with context −sensitive help available.

Required

Provide context−sensitive help at all locations.

Never use a "Help not available" message.

Default Activation

Required

If your application uses default push buttons in a window, highlight the current default push button. When the focus is on a push button, make its action the default action and highlight the push button. If the default action in a window varies, some push button must always have default highlighting, except when there is no current default action.

Required

When focus is in a window with a default action and an activatable menu does not have the focus, pressing `Enter` or `Ctrl Return` invokes the default action. If focus is in a component other than multiline text or an activated menu, `Return` also invokes the default action. These actions should have no other effect on the component with the focus, unless the default action has some effect on that component.

Required

Except in the middle of a button motion operation, make pressing `Cancel` anywhere in a dialog box equivalent to activating the Cancel push button in the dialog box.

Expert Activation

Required

If your application supports expert activation, expert actions should exist only as shortcuts to application features that are available through another mechanism.

Expert activation, using mouse double–clicking on buttons, provides a convenient way for experienced users to perform certain tasks quickly. However, new users and keyboard–only users need to be able to perform the same tasks.

Required

When the focus is on a button used for expert activation, no default action should be available, unless the default and expert actions are the same.

Required

If a component with an expert action is selectable, activating the expert action first selects the component and then performs the expert action.

Previewing and Autorepeat

Required

If your application supports activation preview when the user presses the SELECT button, remove the previewing information when the user releases the SELECT button. Activation preview presents the user with additional information that describes the effect of activating a button. This information cannot interfere with the normal operation of the application.

Cancel Activation

Required

Pressing the `Cancel` key stops current interaction in the following contexts:

During a mouse–based selection or drag operation, it cancels the operation.

During a mouse–based scrolling operation, it cancels the scrolling action and returns the system to its state prior to the start of the scrolling operation.

Anywhere in a dialog box that has a Cancel push button, it activates that push button, except during a mouse–based selection or drag operation.

In a pull–down menu, it either dismisses the menu and moves the location cursor to the cascading button used to pull it down, or it unposts the entire menu system. In a pop–up menu, option menu, tear–off menu, or menu bar, it unposts the menu system.

When the focus is in a torn off menu window, it closes the torn–off menu window.

Essential Related Topics

For more information, see the **Cancel (Action and Choice)** , Mnemonic , Shortcut Key , Undo, Redo, Repeat (Action Choices) , and Window Navigation reference pages.

Supplemental Related Topics

For more information, see the Selection reference page.

Controls, Groups, and Models (CDE)

NAME

Controls, Groups, and Models (CDE) –Reference

Description

Controls and groups are components that the user interacts with in CDE. Models are guidelines for the layout and functionality of controls.

Guidelines

Check Button

Required

Check buttons should only select settings that are not mutually exclusive. A check button graphically indicates its state with the presence or absence of a check mark.

Required

When the user presses the SELECT button in a check button, display the check button with armed emphasis. If the check button was previously unset, show it in the set state. If the check button was previously set, show it in the unset state.

Pressing the SELECT button arms a check button; releasing the SELECT button shows the result of activating it.

Required

When the user releases the SELECT button in the same check button in which the press occurred:
If the check button was previously unset, set it.
If the check button was previously set, unset it.

In all cases, disarm the check button and, if the check button is in a menu, unpost the menu. When the user releases the SELECT button, activate the check button.

Required

When the user presses `Enter` or `Return` in a check button, if the check button is in a window with a default action, activate the default action. If the check button is in a menu:
If the check button was previously unset, set it.
If the check button was previously set, unset it.
In both cases, disarm the check button and unpost the menu.

Required

When the user presses `Select` or `Spacebar` in a check button, if the check button was previously unset, set it. If the check button was previously set, unset it. In both cases, disarm the check button and, if the check button is in a menu, unpost the menu.

Combination Box

Required

In a list that can be scrolled, such as a scrollable list box, do not allow the cursor to wrap.

Required

Provide vertical scroll bars when some of the data is not visible in the combination box.

Recommended

Provide horizontal scroll bars when elements are wider than the list box.

Recommended

Display the elements in an order that is meaningful to the user.

Recommended

Display an initial value from the list in the text–entry field. Display selected emphasis on the initial value so that typed text will replace the value.

Recommended

Make the combination box large enough to display a minimum of six list items at a time.

Recommended

When a user increases the size of the window in which the combination box is displayed, increase the number of items displayed in the combination box.

Recommended

When a user decreases the size of the window in which the combination box is displayed, decrease the number of items displayed in the combination box. As a minimum, reduce the combination box to the text–entry field and a list box with one entry displayed. If the window is sized so that two list items cannot be displayed, clip the combination box.

Command Box

Required

If your application uses a command box, include a text component with a command–line prompt for text input and a list component for a command history area. Use either the single selection or browse selection model.

Required

When an element of a command box list is selected, place its contents in the text area.

Required

Make the list navigation actions `[uarr ]`, `[darr ]`, `Ctrl Begin`, and `Ctrl End` available from the text component so the user can move the cursored element within the list and thus change the contents of the text.

Required

The default action of the command box should pass the command in the text area to the application for execution and add the command to the end of the list.

File Selection Box

Required

If your application uses a file selection dialog box, it should contain the following components:

A directory text component that shows the current directory path. The user can edit the directory text component and press `Return` or `Enter` to change the current directory.

For applications that allow saving to different formats, an option button that allows users to specify the format when saving a file.

A file name text component for displaying and editing a file name. This component is optional when the file selection box is used to choose an existing file or directory.

A group of push buttons, including a command button, and Update, Cancel, and Help buttons. The command button is typically labeled Open or Save, but if there is another label that better describes the resulting action (such as Include), use that label. Activating the command button carries out the corresponding action and dismisses the file selection box.

Recommended

When the file selection box is used to specify an existing file (for example, to open a document), the command button is normally labeled

Open and it should be the default action.

Recommended

If the Update button is activated while a directory is selected in the contents list, open the directory, display its contents in the contents list, and update the directory text.

Required

If the Open button is activated while the appropriate file is selected in the contents list, the file should be utilized by the application and the file selection box dismissed.

Recommended

When the file selection dialog box is used to choose an existing directory (for example, to install a set of files into the chosen directory) or to specify a new directory, the command button should be given an appropriate label, such as Install, Choose, Create, or OK. If this button is activated while the appropriate directory is selected in the contents list, the directory should be utilized by the application and the file selection box dismissed.

Required

When the file selection dialog box is used to choose an existing directory, there must also be an additional button, labeled Update, that is enabled whenever a directory is selected in the contents list and that opens the directory. Make this Update button the default action.

Recommended

When the file selection dialog box is used to specify a new file name (for example, a Save As dialog box), label the command button Save and make it the default action.

Optional

When the file selection dialog box is used to choose an existing file, show files in the contents list but disable them. Double−clicking the SELECT button on a disabled file name should have no effect.

Required

Make the normal text navigation and editing functions available in the text components for moving the cursor within each text component and changing the contents of the text. These actions provide a convenient way to choose a directory or file name from the corresponding list while focus remains in the text component.

Required

When the user double−clicks the SELECT button on an item in the contents list, select that item and activate the default action. In all cases, double−clicking the SELECT button on a directory in the contents list should open that directory and display its contents in the contents list (the default action is Open).

When the file selection box is used to choose an existing file and the user double−clicks the SELECT button on an appropriate file in the contents list, choose that file and dismiss the file selection box (the default action is Open).

When the file selection box is used to choose an existing directory or to specify a new directory or file, the files list should not appear.

Required

Make the normal text navigation and editing functions available in the text components for moving the cursor within each text component and for changing the contents of the text.

Optional

Allow the user to select a file by scrolling through the list of file names and selecting the desired file or by entering the file name directly into the file selection text component. Selecting a file from the list should cause that file name to appear in the file selection text area.

This method of selecting a file needs to be consistent across applications.

Required

Make use of a selection when one of the following occurs:

The user activates the command push button while an appropriate item is selected in the contents list

The user double−clicks the SELECT button on an appropriate file in the contents list

The user presses `Return` or `Enter` while the file name text component has keyboard focus and contains an appropriate item

Required

Make the file selection box display the contents of a directory in the contents list when the file selection box is initialized, when the user presses `Enter` or `Return` in the directory text component, and when the user opens a directory in the contents list. Update the contents list each time the contents of the directory changes.

Recommended

If the user has opened the application without supplying a file name argument, the Open dialog box should use the user's home directory as the default directory.

An exception to this rule might be made if a clearly more useful directory can be identified; for example, the icon editor might default to *HomeDirectory/.dt/icons*. For applications that allow editing, never default to a directory in which the user does not have read and write permission, such as */usr/dt/bin*.

Required

If the user has opened the application with a file name argument, the Open dialog box should default to the directory in which that file resides.

Optional

When using the file selection dialog box in Save As capacity, provide a default name of Untitled, place the location cursor in the file name field and highlight the file name text to create a "delete pending type−in" mode. If the current directory already has a file of that name, create a name Untitled2, and so forth.

Optional

When using the file selection dialog box in a Save As capacity, add a file name extension if the application supports file typing by extension, and make this extension visible in the file name field. Do not highlight the extension to create a "delete pending type−in" mode, but allow the user to modify the extension or to delete it explicitly.

Optional

The file selection dialog box should come up in a directory that makes sense for the task. For example, when saving a new file from an editor, the file selection box should come up in the user's home directory. If the user navigates to some other directory within the file selection box, the application should remember that directory the next time it is brought up.

Optional

Never allow the user to overwrite an existing file through the file selection box without providing a warning dialog box prompt.

Optional

Place keyboard focus in the file name field each time the user brings up a file selection dialog box.

Optional

Present directory and file name lists alphabetically, case insensitive. The first item on the directory list should be the parent directory and it should be labeled "..".

Optional

Create clear labels. In the English language, use the labels in **File Selection Dialog Box Labels** for the file selection dialog box fields and lists.

Table 1 File Selection Dialog Box Labels

Component	Label
Directory text field	Enter Path or Folder Name:

Filter text field	Filter:
Directory list	Folders:
Contents list	Files:
File text field	Enter File Name:*

Application developers can make this label more instructive and specific, such as Enter File to Open for Open dialog boxes.

These labels should be the default labels. If they are not set by default, you need to set them through resources in your application's *app−defaults* file.

List

Required

Within a list component, pressing `[uarr ]` moves the location cursor to the previous item in the list, and pressing `[darr ]` moves the location cursor to the next item in the list. In a scrollable list, pressing `[larr ]` scrolls the list one character to the left, and pressing `[rarr ]` scrolls the list one character to the right.

Required

Within a list component, pressing `Ctrl Begin` moves the location cursor to the first item in the list, and pressing `Ctrl End` moves the location cursor to the last item in the list. In a scrollable list, pressing `Begin` moves the horizontal scroll region so that the leftmost edge of the list is visible, and pressing `End` moves the horizontal scroll region so that the rightmost edge of the list is visible.

Required

Within a scrollable list, pressing `Page Up` moves the location cursor to the item one page up in the list, and pressing `Page Down` moves the location cursor to the item one page down in the list. In a scrollable list, pressing `Page Left` (or `Ctrl Page Up`) scrolls the list one page to the left, and pressing `Page Right` (or `Ctrl Page Down`) scrolls the list one page to the right.

Required

Within a list component, clicking the SELECT button selects the item that was double−clicked and then initiates any default action for the window.

Option Button

Required

If your application uses option buttons, make the label for the button the last selection made from the option button.

Required

When the user presses the SELECT button or the MENU button on an option button, post the associated option menu.

Required

When the user releases the SELECT button or the MENU button within the same option button that the press occurred in, post the associated option menu if it was not posted at the time of the press. When the user releases the SELECT button or the MENU button outside of the option button, unpost the associated option menu.

Required

When the user presses `Select` or `Spacebar` in an option button, post the associated option menu.

Paned Window

Required

If your application uses paned windows, they should be composed of any number of groups of components, called panes, each separated by a sash and a separator. Group the panes, sashes, and separators linearly, either horizontally or vertically. A sash is the handle on a separator between two panes that is used to adjust the position of the separator.

Panel

Required

The [uarr], [darr], [rarr], and [larr] directional keys navigate among components in a panel.

A panel group organizes a collection of basic controls in a horizontal, vertical, or two−dimensional layout. The directional keys are used to navigate among the controls.

Push Button

Required

When the user presses the SELECT button in a push button, arm the push button. When the user releases the SELECT button in the same push button where the press occurred, disarm and activate the push button. When the user releases the SELECT button outside the push button, disarm but do not deactivate the push button.

Required

When the user presses Enter or Return in a push button that is in a window with a default action, activate the push button. When the user presses Enter or Return in a push button in a menu, activate the push button and unpost the menu.

Required

When the user presses Select or Spacebar in a push button, activate the push button. If the push button is in a menu, unpost the menu.

Radio Button

Required

If your application uses radio buttons, graphically indicate each button's state.

Required

When the user presses the SELECT button in a radio button, the radio button is armed. If the radio button was previously unset, show it in the set state.

Required

When the user releases the SELECT button in the same radio button where the press occurred, and the radio button was previously unset, set it and unset any other radio button in the same panel that was previously set. Disarm the radio button and, if the radio button is in a menu, unpost the menu.

Required

When the user presses Enter or Return in a radio button, if the radio button is in a window with a default action, activate the default action. If the radio button is in a menu:

If the radio button was previously unset, set it and unset any other radio button in the same panel that was previously set.
Disarm the radio button and unpost the menu.

Required

When the user presses Select or Spacebar in a radio button, if the radio button was previously unset, set it and unset any other radio button in the same panel that was previously set. Disarm the radio button and, if the radio button is in a menu, unpost the menu.

Sash

Required

Within a paned window, a sash adjusts the position of a separator, which adjusts the sizes of the panes next to it. As a sash is moved, make the pane in the direction of the sash movement get smaller and the opposite pane get larger by an equal amount.

Required

Within a sash, pressing the SELECT button or the TRANSFER button and moving the pointer causes the sash to track the movement of the pointer. In a vertically oriented paned window, the sash tracks the vertical position of the pointer. In a horizontally oriented paned

window, the pane tracks the horizontal position of the pointer.

Required

Pressing [uarr] or [darr] (for a sash that can move vertically) and [larr] or [rarr] (for a sash that can move horizontally) moves the sash one increment in the specified direction.

Required

Pressing Ctrl [uarr] or Ctrl [darr] (for a sash that can move vertically) and Ctrl [larr] or Ctrl [rarr] (for a sash that can move horizontally) moves the sash one large increment in the specified direction.

Scale

Required

If a scale has arrow buttons, pressing the SELECT button on an arrow button moves the slider one increment in the direction of the side of the slider on which the button was pressed and autorepeats until the button is released.

Required

In a scale trough, if the scale has tick marks, pressing the SELECT button moves the slider one major tick mark in the direction of the side of the slider on which the trough was pressed and autorepeats until the button is released. If the scale does not have tick marks, pressing the SELECT button in the trough moves the slider one large increment in the direction of the side of the slider on which the trough was pressed and autorepeats until the button is released.

Required

Within a scale slider, pressing the SELECT button and moving the pointer causes the slider to track the position of the pointer. In a vertical scale, the slider tracks the vertical position of the pointer. In a horizontal scale, the slider tracks the horizontal position of the pointer.

Required

Within a scale slider or trough, pressing the TRANSFER button and moving the pointer positions the slider to the point of the button press and then causes the slider to track the position of the pointer. In a vertical scale, the slider tracks the vertical position of the pointer. In a horizontal scale, the slider tracks the horizontal position of the pointer.

Required

If a mouse–based sliding action is in progress, pressing Cancel cancels the sliding action and returns the slider to its position prior to the start of the sliding operation.

Required

In a vertical scale, pressing [uarr] or [darr] moves the slider one increment in the specified direction. In a horizontal scale, pressing [rarr] or [larr] moves the slider one increment in the specified direction.

Required

In a vertical scale, pressing Ctrl [uarr] or Ctrl [darr] moves the slider one large increment in the specified direction. In a horizontal scale, pressing Ctrl [larr] or Ctrl [rarr] moves the slider one large increment in the specified direction.

Required

Pressing Begin or Ctrl Begin moves the slider to its minimum value. Pressing End or Ctrl End moves the slider to its maximum value.

Scroll Bar

Required

Within a scroll bar, pressing the SELECT button in an arrow button moves the slider one increment in the direction of the side of the slider on which the button was pressed and autorepeats until the button is released.

Required

In the trough of a scroll bar, pressing the SELECT button moves the slider one page in the direction of the side of the slider on which the trough was pressed and autorepeats until the button is released.

Required

Within a scroll bar slider, pressing the SELECT button and moving the pointer causes the slider to track the position of the pointer. In a vertical scroll bar, the slider tracks the vertical position of the pointer. In a horizontal scroll bar, the slider tracks the horizontal position of the pointer.

Required

Within a scroll bar slider or trough, pressing the TRANSFER button and moving the pointer positions the slider to the point of the button press and then causes the slider to track the position of the pointer. In a vertical scroll bar, the slider tracks the vertical position of the pointer. In a horizontal scroll bar, the slider tracks the horizontal position of the pointer.

Required

If a mouse–based scrolling action is in progress, pressing the `Cancel` key cancels the scrolling action and returns the slider to its position prior to the start of the scrolling operation.

Required

In a vertical scroll bar, pressing `[uarr ]` or `[darr ]` moves the slider one increment in the specified direction. In a horizontal scroll bar, pressing `[larr ]` or `[rarr ]` moves the slider one increment in the specified direction.

Required

In a vertical scroll bar, pressing `Ctrl [uarr ]` or `Ctrl [darr ]` moves the slider one large increment in the specified direction. Pressing `Ctrl [larr ]` or `Ctrl [rarr ]` moves the slider one large increment in the specified direction.

Required

Pressing `Page Up` or `Page Down` moves the slider in a vertical scroll bar one page in the specified direction. Pressing `Page Left` (or `Ctrl Page Up`) or `Page Right` (or `Ctrl Page Down`) moves the slider in a horizontal scroll bar one page in the specified direction.

Required

Pressing `Begin` or `Ctrl Begin` moves the slider to the minimum value. Pressing `End` or `Ctrl End` moves the slider to the maximum value.

Selection Box

Required

If your application uses a selection box, include at least a text component for the selected alternative and a list component above the text component for presenting alternatives. Use either the single selection or browse selection model. Selecting an element from the list should place the selected element in the text component.

Required

Make the list navigation actions `[uarr ]`, `[darr ]` `Ctrl Begin`, and `Ctrl End` available from the text component so the user can move the cursored element within the list and thus change the contents of the text.

Spin Box

Required

Present the items as a ring of items that wrap. For example, if a user is at the largest number and presses the up arrow, display the smallest number and vice versa so that the user can spin through all the items by pressing the same arrow.

Required

Move through the items in a spin box as shown in Navigation in a Spin Box .

Table 2 Navigation in a Spin Box

Movement	Keys	Example
Toward the beginning of the list	[larr], [darr]	Chronology: If Tuesday is displayed, move to Monday when the user presses [larr] or [darr] . Magnitude: If 15 is displayed, move to 14 when the user presses [larr] or [darr] .
Toward the end of the list	[rarr], [uarr]	Chronology: If Tuesday is displayed, move to Wednesday when the user presses [rarr] or [uarr] . Magnitude: If 15 is displayed, move to 16 when the user presses [rarr] or [uarr] .

Recommended

Allow the user to set values with the arrow buttons or through keyboard input. Values should be evaluated immediately upon entry. If a value is entered that is already in the list, scroll to the position of that entry in the list.

Recommended

If entry of nonlisted items is permitted, use the following behavior: when a new value is entered, scroll the list to the position appropriate for the new entry; if the user scrolls off the new entry, scroll to the next appropriate value in the list and dismiss the keyboard–entered value.

Recommended

On entry of an invalid value, provide an auditory warning and error message.

Text

Required

In a multiline text component, pressing [uarr] moves the location cursor up one line and pressing [darr] moves the location cursor down one line. In a single–line text component, pressing [uarr] navigates upward to the previous component and pressing [darr] navigates downward to the next component, if the text component is designed to act like a basic control.

Required

Pressing [larr] moves the location cursor left one character and pressing [rarr] moves the location cursor right one character.

Required

In a text component generally used to hold multiple words, pressing Ctrl [rarr] moves the location cursor to the right by a word and pressing Ctrl [larr] moves the location cursor to the left by a word.

Required

In a text component generally used to hold multiple words, pressing Begin moves the location cursor to the beginning of the line and pressing End moves the location cursor to the end of the line.

Required

In a multiline text component, pressing Ctrl Begin moves the location cursor to the beginning of the file and pressing Ctrl End moves the location cursor to the end of the file.

Required

Pressing Spacebar or Shift Spacebar inserts a space in a text component. Modifying these keys with Ctrl invokes the normal selection function.

Required

Pressing `Return` in a multiline text component inserts a carriage return. Pressing `Enter` or `Ctrl Return` invokes the default action.

Required

In a multiline text component, `Tab` is used for tabbing. In a single−line text component, `Tab` is used either for tabbing or to move to the next field.

Required

If a text component supports replace mode, insert toggles between insert mode and replace mode.

By default, the component should start in insert mode, where the location cursor is between two characters. In insert mode, typing a character inserts the character at the position of the location cursor.

In replace mode, place the location cursor on a character. Typing a character replaces the current character with the newly entered character and moves the location cursor to the next character, selecting it.

Required

Double−clicking the SELECT button selects text a word at a time.

Gauge

Required

A gauge is a display−only device and should have no user interactions. The appearance of a gauge is similar to a scale, but the gauge lacks a scale slider.

Optional

Despite being a display−only device, a gauge should get keyboard focus so that the user can access Help or settings for that control.

Essential Related Topics

For more information, see the Control and Selection Models reference pages.

Supplemental Related Topics

For more information, see the Data Transfer and Selection reference pages.

Container (Control)

NAME

Container (Control) –Reference

Description

A container is a control whose specific purpose is to display objects as icons and allow them to be selected and operated upon.

Figure 1 Container

The objects in a container may be viewed in two ways:

Spatial view

Icons are laid out in two dimensions, possibly with additional layout constraints imposed by the application, and may be positioned (within those constraints) by the user.

Linear view

The icons are laid out linearly from top to bottom, one row per icon. A linear view may have one or both of the following characteristics:

Detailed

A detailed linear view provides additional columns in which various properties or additional data associated with each object may be shown. Each row then also contains the detail elements specific to the object whose icon is shown.

Hierarchical

A hierarchical linear view allows objects that contain other objects (such as directories, folders, or any object that can itself be viewed as a container) to have its subobjects displayed in–line. Each row may also contain a display control button (which cannot take focus) that indicates whether the object's subobjects are displayed or not.

Guidelines

Recommended

When a container is the main control in a view, allow a user to choose a spatial or a linear view through the View menu.

Recommended

When a container is the main control in a view, allow a user to determine which details are displayed in a linear view and the order in

which they appear.

Recommended

When a container is the main control in a view, and it contains objects of the same type as the object being viewed (using the container), allow the view to be hierarchical.

Recommended

Provide an Include choice for a view based on a container.

Required

Based on a task analysis, allow a container to either support a single, browse, multiple, or extended selection model.

Focus on the Container as a Whole

Required

If you design a container so that the container as a whole can take focus, then the following should occur:

If the container has not had focus previously, initially moving focus to the container places the active cursor only on the container as a whole (not on an element within the container).

If the container previously had focus, moving the focus to the container places the active cursor on the element that was last cursored when the container had focus −− either the container as a whole or an element within the container if it still can be cursored.

When focus is on an element within the container and the user presses `Ctrl Home`, move the active cursor to the container as a whole. If normal mode is being used, this should deselect all selected objects in the container.

When the user presses `[darr ]` or `[rarr ]` and the active cursor is on the container as a whole, move focus to the top–left element within the container that can be cursored. For information on bidirectional and vertical language, see Chapter 11.

Required

When a container as a whole can take focus, the following should occur:

If the user clicks the SELECT button in the background of the container, move the active cursor to the container as a whole. If select mode is used within the container, all objects should be deselected.

If the user clicks `Ctrl` SELECT in the background of the container, move the active cursor to the container as a whole without affecting the selection state of the objects within the container.

Required

When the container as a whole has focus, a transfer operation whose destination depends upon the focus (for example, Paste) transfers into the object represented by the container as a whole. If the elements being transferred are not objects, the application encapsulates them as objects where this is possible.

Required

Design a container so that the container as a whole can take focus if support for transferring objects to the container is required, unless the user can switch to a spatial view by using a graphics cursor.

Spatial Views

Required

In a spatial view, either use an element cursor or a graphics cursor.

Recommended

When displaying a spatial view of a container, provide a choice or choices (for example, Arrange) in the View menu to lay out the objects in the container.

Recommended

When using a spatial view to select multiple elements in a container, support the area and touch selection techniques.

Linear Views

Required

In a linear view, use an element cursor.

Required

In a linear view, when the active cursor is on an object, pressing [darr] (except on the last object) moves it to the object in the row below.

Required

In a linear view, when the active cursor is on an object, pressing [uarr] (except on the first object) moves it to the object in the row above.

Required

In a linear view that is neither detailed nor hierarchical, pressing [rarr] has the same effect as [darr], and [larr] has the same effect as [uarr]. For bidirectional and vertical language support, see Chapter 11.

Required

In a linear view, to represent an object use a small icon with the text label adjacent to the image if both are present.

Required

In a linear view, do not allow the icon's text label to be cursored separately from the icon image. If you want to allow the user to change the label text, support direct editing techniques, property−editing dialogs, or both.

Required

In a linear view, only the text label or the image of an icon needs to be shown. However, whichever one is present should uniquely identify the object.

Recommended

Provide a Sort choice when displaying a linear view of a container.

Required

Support the range technique when multiple elements can be selected in a container that uses a linear view.

Linear Views with Detail Elements That Take Focus

Required

In a linear detailed view, either all detail elements take focus or they all do not take focus. If they do not take focus, and you want to allow the user to change the details, support direct editing techniques, property−editing dialogs, or both. Detail elements are not selectable, even when they can take focus.

Required

When multiple elements in a row of a nonhierarchical linear view can be cursored (because detail elements can take focus in addition to the icon in each row), follow the standard rules for directional navigation; that is, [larr] and [rarr] traverse through rows, and [uarr] and [darr] traverse through columns, with the objects treated as a single column.

Required

In normal mode, if keyboard navigation is used to move focus to a detail element, deselect all objects in the container.

Required

If the user clicks Ctrl SELECT on a detail element in the container that can take focus, move the focus to that element without affecting the selection state of the objects within the container or having any other effect.

Required

If the user clicks SELECT on a detail element in the container that can take focus, move the focus to that element. If select mode is used within the container, all objects are deselected as well.

Linear Hierarchical Views

Required

In a linear hierarchical view, each object that contains subobjects should be displayed with a display control button to its left. The button should contain a value choice that may be toggled between an *''expanded''* and a *''contracted''* state and should not take focus.

Required

When an object's display control button is in the contracted state, the subobjects of the object should not be shown; in the contracted state, they should be shown. All of an object's subobjects should be laid out vertically immediately below the object, but indented to the right (only the object column −− not the detail columns). If any subobject itself contains subobjects, it should also be shown with a display control button that indicates whether its subobjects should be displayed. For information on bidirectional and vertical language support, see Chapter 11.

Required

When focus is on an object with a display control button and the user presses `Ctrl [larr ]`, set the button to its expanded state. If the user presses `Ctrl [rarr ]`, set the button to its contracted state. For information on bidirectional and vertical language support, see Chapter 11.

Required

If the user clicks the SELECT button on a display control button, toggle its state and move focus to its corresponding object icon.

Required

Toggling a display control button (either with a standard keyboard or mouse) has the following effect:

If the display control button toggles from the contracted to expanded state, the selection state of the objects in the container is unchanged.

If the display control button toggles from the expanded to the contracted state, objects that are no longer displayed are deselected, but the selection state of other objects is unchanged. If the active cursor is on an element that is no longer displayed, move the cursor to the object icon that corresponds to the display control button.

Recommended

In a linear hierarchical view, when the active cursor is on an object, pressing `Ctrl [uarr ]` moves the cursor upward to the nearest sibling object. If there is no sibling, it moves the cursor to its parent object.

Recommended

In a linear hierarchical view, when the active cursor is on an object, pressing `Ctrl [darr ]` moves the cursor downward to the nearest sibling object. If there is no sibling, it acts as if `Ctrl [darr ]` were pressed with the cursor on its parent object.

Recommended

When a display control button is in its expanded state, it should be shown as a right−pointing arrow; in its contracted state, it should be shown as a down−pointing arrow. For information on bidirectional and vertical language support, see Chapter 11.

Optional

In a linear hierarchical view in which detail elements do not take focus (or without detail elements), when focus is on a subobject, pressing `[larr ]` moves focus to its parent object. For information on bidirectional and vertical language support, see Chapter 11.

Essential Related Topics

For more information, see the **Control** and Selection reference pages.

Supplemental Related Topics

For more information, see Chapter 4 and the Data Transfer, Icon, Include (Choice and Dialog), Object, Primary Window, Sort (Choice and Dialog), Status Area, and View reference pages.

Context–Sensitive Help

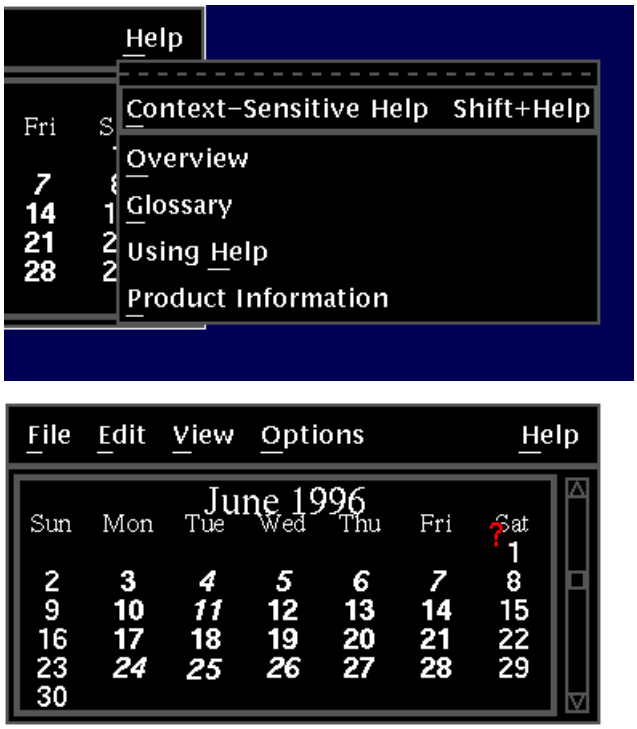
NAME

Context–Sensitive Help -Reference

Description

Context–sensitive help is help information that is adapted to the current context of an element or group of elements.

Figure 1 Context–Sensitive Help



When to Use

Recommended

Provide context–sensitive help for each window, control, and, where useful, for each element with which the user can interact.

Guidelines

Required

When the user presses `F1` or `Help`, display context–sensitive help related to the element that has focus.

Required

When the user is in help mode (as a result of pressing `Shift F1`, `Shift Help`, or selecting the On Item entry in the Help menu), display context−sensitive help on the element on which the user clicks the SELECT button, if it is in the same window in which help mode was initiated.

Required

If help is not available for the element for which it was requested, provide help for the control that contains the element, or for some higher −level grouping of controls, or for the window as a whole.

Recommended

Display context−sensitive help in a secondary window. However, if all the help information provided within a window is brief, it may be displayed in the information or message area.

Recommended

Allow a user to enter and exit from instant help mode. This has no effect on user interaction except that it always displays instant help for the element under the pointer.

Recommended

Display instant help in the information area of the window that contains the pointer or in a separate window popped up at the pointer location.

Recommended

Provide context−sensitive help for choices that are unavailable when using help mode.

Recommended

Provide context−sensitive help for choices that are unavailable.

Recommended

If a choice is displayed with unavailable emphasis and context−sensitive help is provided, describe in a Help dialog why the choice is unavailable and how the user might make the choice available. For example, if the user requests help on the Paste choice currently displayed with unavailable emphasis, explain in the Help dialog that Paste is not available because the clipboard is empty.

Supplemental Related Topics

For more information, see the **Help (Menu/Action Choice)** and Keyboard (Device) reference pages.

Control

NAME

Control –Reference

Description

A control is a visually recognizable element or group of elements that the user interacts with. Controls are usually identified by text (headings, push button labels, and window titles for example).

When to Use

Provide controls to allow the user to interact with your application.

Guidelines

Recommended

Identify each control or group of controls with a label, a column heading, a group heading, or a window title, whichever is most appropriate.

Recommended

In documentation, avoid using the term *''control.''* Instead, describe the specific type of control.

Recommended

Use a label that clearly indicates the control's function.

Recommended

If a standard or common control meets your needs, use it rather than designing a new control unless an analysis indicates a clear advantage for the new control.

Recommended

When a window is sized, adjust the size of the controls. For example, make the text–entry fields longer or shorter as the window is sized larger or smaller. At some minimal useful size, clip the controls instead of adjusting the size; that is, remove characters from the right side of the displayed text in the controls to allow them to fit into the available space.

Supplemental Related Topics

For more information, see the **Check Box (Control)** , Choice , Combination Box (Control) , Command Box (Control) , Container (Control) , Default Action , Drop–Down Combination Box (Control) , Drop–Down List (Control) , Gauge (Control) , Group Box , Label , List Box (Control) , Menu (Control) , Menu Bar (Menu Type) , Notebook (Control) , Option Menu (Menu Type) , Push Button (Control) , Radio Button (Control) , Selection Box (Control) , Slider (Control) , Spin Box (Control) , Text–Entry Field (Control) , Text Field (Abstract Control) , Value Set (Control) , Window Frame , Window Navigation , and Window Title reference pages.

Control Navigation

NAME

Control Navigation –Reference

Description

Control navigation has two types: horizontal and vertical.

Horizontal navigation refers to navigating with the [larr] and [rarr] keys, possibly augmented by the Ctrl modifier.

Vertical navigation refers to navigating with the [uarr] and [darr] keys, possibly augmented by the Ctrl modifier.

When to Use

Required

Allow control navigation to move the active cursor among controls in a tab group.

Guidelines

Horizontal Navigation

Required

When unaugmented horizontal navigation is used to navigate among controls in a tab group, the [rarr] and [larr] keys work as follows:

When the user presses [rarr], move the active cursor through all the controls that can take focus in a tab group, starting from the upper left and ending at the lower right.

If the user presses [rarr] when controls are laid out in rows, traverse through one row, then traverse the row below it, and so forth.

When the user presses [rarr] at the lower right control, wrap back to the control at the upper left if and only if the tab group is not horizontally scrollable; otherwise, it should have no effect.

When the user presses [larr], move the active cursor through all the controls that can take focus in a tab group in the exact opposite order as pressing [rarr].

Required

When augmented horizontal navigation is used to navigate among controls in a tab group, pressing Ctrl [larr] and Ctrl [rarr] move the active cursor through all the controls that can take focus in a tab group, following the same set of rules as for unaugmented horizontal navigation described previously. For information on bidirectional and vertical language support, see Chapter 11.

Required

Support unaugmented horizontal navigation for control navigation within a tab group, unless controls within the tab group use horizontal navigation internally.

Required

If controls within the tab group use unaugmented horizontal navigation internally, support augmented horizontal navigation for control navigation, unless controls within the tab group also use augmented horizontal navigation internally.

Recommended

If unaugmented and augmented horizontal navigation are both supported for control navigation, they navigate to the same control.

Vertical Navigation

Required

When unaugmented vertical navigation is used to navigate among controls in a tab group, the [darr] and [uarr] keys work as follows:

When the user presses [darr], move the active cursor through the controls that can take focus in a tab group, starting from the upper left and ending at the lower right.

When the user presses [darr] and the controls are laid out in columns, traverse through one column, then traverse the column to its right, and so forth.

When the user presses [darr] at the lower right control, wrap back to the control at the upper left if and only if the tab group is not vertically scrollable; otherwise, it should have no effect.

When the user presses [uarr], move the active cursor through all the controls that can take focus in a tab group in the exact opposite order as pressing [darr].

Required

When augmented vertical navigation is used to navigate among controls in a tab group, pressing Ctrl [darr] and Ctrl [uarr] move the active cursor through all the controls that can take focus in a tab group, following the same set of rules as for unaugmented vertical navigation described previously.

Required

Support unaugmented vertical navigation for control navigation within a tab group, unless controls within the tab group use vertical navigation internally.

Required

If controls within the tab group use unaugmented vertical navigation internally, support augmented vertical navigation for control navigation, unless controls within the tab group also use augmented vertical navigation internally.

Recommended

If unaugmented and augmented vertical navigation are both supported for control navigation, they should navigate to the same control.

Navigation

Required

If a tab group contains more than one control that can take focus, then either horizontal or vertical navigation should be supported for control navigation within the tab group.

Recommended

If a tab group contains controls laid out in both dimensions (for example, not just in a single row or column), support both horizontal and vertical navigation within the tab group.

Optional

If the controls in a tab group use horizontal navigation internally, allow Tab and Shift Tab to be used (instead of [larr] and [rarr]) for horizontal navigation (requiring Ctrl Tab and Ctrl Shift Tab to be used to move to the next or previous tab group).

Tab Groups That Contain Text Fields

Optional

Allow a tab group of vertically organized multiple−line text fields to use horizontal and unaugmented vertical navigation internally, while using augmented vertical navigation to move among the fields,

Optional

Allow a tab group of vertically organized single−line text fields to use horizontal navigation internally, while using vertical navigation to move among the fields.

Optional

Allow a table of single−word text fields to be organized as a tab group in which unaugmented horizontal navigation is used to navigate internally within the text fields, and in which augmented horizontal navigation is used to move among the fields.

Optional

Allow a table of text fields to be organized as a tab group in which horizontal navigation is used to navigate internally within the text fields, while `Tab` and `Shift Tab` are used for horizontal control navigation (requiring `Ctrl Tab` and `Ctrl Shift Tab` to be used to move to the next or previous tab group).

Initial and Final Control

Required

The initial control of a tab group (the control that gets focus when the user presses `Tab` or `Ctrl Tab` to navigate to the tab group) should be either the top−left control that can take focus or the control with which the user is most likely to want to interact.

Required

In a tab group, if you allow `Tab` to navigate internally among all of its controls, the final control in the tab group should be the final control visited (without repeating) when starting at the initial control.

Required

If you do not allow `Tab` to navigate internally among all the controls in the tab group with wrapping, then allow `Shift Tab` or `Ctrl Shift Tab` to place focus on the initial control of the tab group.

Required

In a tab group, if you allow `Tab` to navigate internally among all the controls in the tab group, but do not allow warping from the final to the initial control, then allow `Shift Tab` or `Ctrl Shift Tab` to place focus on the final control of the tab group.

Required

If a tab group of push buttons contains the push button that has default emphasis when focus initially enters a window, then make that push button the initial control of its tab group.

Tab Groups of Push Buttons

When focus is within a tab group of push buttons, support `Tab` in one of the following ways:

When focus is on the final push button in the tab group, move focus to the next tab group.
If the tab group is laid out in columns, but not in rows, navigate vertically to the next push button.
Navigate horizontally to the next push button.

Continue to allow the directional keys to navigate within the tab group as well.

Required

When focus is within a tab group of push buttons, support `Shift Tab` in one of the following ways:

When focus is on the initial push button of the tab group, navigate to the previous tab group.
Navigate in the reverse direction of `Tab`.

Home and End Keys

Recommended

When a tab group contains controls laid out in multiple rows and columns, allow `Home` (or `Begin`) to move to the first control in a row and `End` to move to the last control in a row.

Recommended

When the controls in a tab group are not laid out in multiple rows and columns, and when horizontal navigation is used to move among the controls, `Home` (or `Begin`) should move the active cursor to the initial control in the tab group, and `End` should move the active cursor to

the final control in the tab group visited (without repeating) when starting at the initial control and navigating horizontally.

Recommended

When vertical navigation is used to move among the controls in a tab group, allow `Ctrl Home` (or `Ctrl Begin`) to move the active cursor to the first control in the tab group and `Ctrl End` to move the active cursor to the final control in the tab group visited (without repeating) when starting at the initial control and navigating vertically.

Essential Related Topics

For more information, see the **Browse Technique** , Cursor , Internal Navigation , Mnemonic , Tab–Group , and Window Navigation reference pages.Chapter 11.

Supplemental Related Topics

For more information, see the Control , Keyboard (Device) , and Push Button (Predefined) reference pages.

Copy To (Dialog Choice)

NAME

Copy To (Dialog Choice) –Reference

Description

The Copy To choice includes the following:

Copy To

A dialog choice that displays a file selection dialog in which the user can specify where copies of files or objects should be placed.

Copy To

Used from the File menu, allows the user to make a copy of the data state of the current window and store it in a file or object. The user can continue to edit the data being viewed. The original source is not updated.

Copy To

Used from the Selected menu, allows the user to copy the selected objects to a specified place.

When to Use

Provide a Copy To choice for all objects for which the user can create a duplicate and for data that the user can duplicate or save to a file.

Guidelines

Required

When appropriate, the File Selection dialog should allow the user to differentiate between copying a file or object versus copying or creating a link or reference to it.

Recommended

The File Selection dialog resulting from choosing Copy To in the File menu should allow the user to copy the file or object (its contents or a reference to it) to the clipboard.

Recommended

When multiple files or objects are being copied, the File Selection dialog must allow the user to name only the place to which the files or objects are to be copied, but not to provide new names for each of them. If any new names are needed, they should be generated from their current names in a way that does not conflict with any names at their destination.

Essential Related Topics

For more information, see the **File Menu** and Selected Menu reference pages.

Supplemental Related Topics

For more information, see the Menu Structure reference page.

Cursor

NAME

Cursor –Reference

Description

A cursor is a visible cue that indicates where the user's interaction with the keyboard will be performed. There are three types of cursors: element cursors, text cursors, and graphics cursors.

Element Cursors

An element cursor indicates that a control can be activated or manipulated, but the control cannot be edited. An element cursor can be any of the following:

Box

A box drawn around an element. This is the default element cursor. It usually has solid borders, but should have dashed borders when used within a selection scope in add mode.

Shadow

An outline with a shadow effect around a control. Provide the shadow shape on a control whose outline is not normally shown, for example, the menu system.

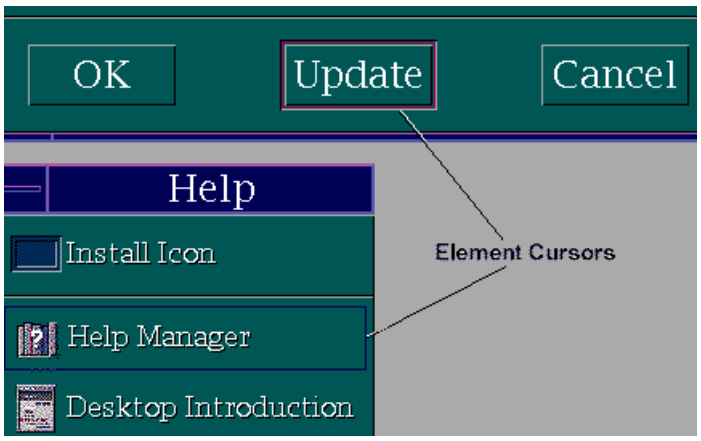
Handles

Solid or hollow square boxes that surround an object (sometimes called resize borders). You typically use handles in applications when the size, position, or shape of an element can be resized.

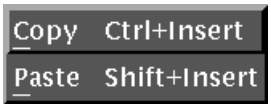
Element Cursors shows each type of element cursor.

Figure 1 Element Cursors

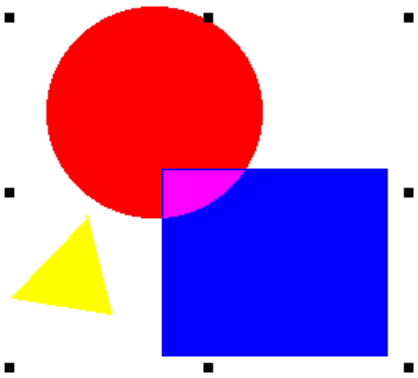
Box Cursor:



Shadow Cursor:



Handles:



Text Cursors

A text cursor indicates the location in text at which characters are inserted or pasted. The text cursor is usually a vertical bar or I–beam between characters when in insert mode and a box over a character when in replace mode. A text cursor should blink when the text has the input focus. A text cursor can be any of the following:

Vertical bar or I–beam

A vertical bar or I–beam in text to indicate where text may be inserted (insert mode). The bar should blink when the text has input focus.

Block or underline

A shaded block behind the character (or sometimes a line under a character). Use it to indicate that it will be replaced (replace mode) by the next character typed. The box or underline may blink when the text has the input focus.

Text Cursors shows each type of text cursor.

Figure 2 Text Cursors

I–Beam Cursor:

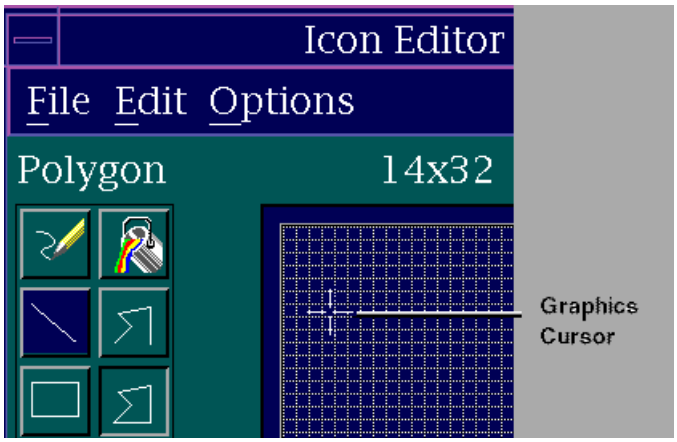
Block Cursor:

Graphics Cursors

A graphics cursor identifies an (x,y) location within a selection scope that indicates the current edit point within a graphics area. You typically use a cross–hair, circle–dot, or circle–X as a graphics cursor to distinguish it from any sighting pointer in the scope.

Graphics Cursor shows a graphics cursor.

Figure 3 Graphics Cursor



When to Use

Required

When using an explicit focus policy, use an element cursor on a control to indicate focus emphasis, unless a cursor is visible on some element within the control.

Required

Use a text cursor at the appropriate point when a control is in a state where characters typed by the user may be inserted in the control or in which characters displayed within the control may be selected.

Required

Use a graphics cursor in a control when keyboard users need to be able to place elements at specific (x,y) positions or to select elements within an arbitrary rectangle.

Required

Use an element cursor on a control to indicate focus emphasis if a cursor is displayed within that control but is not visible because it is clipped or scrolled out of view.

Recommended

Always use an element cursor on a control to indicate focus emphasis.

Recommended

When the purpose of a control is to allow a positional indicator to be set to a value (for example an arm in a slider), indicate focus emphasis with an element cursor on the positional indicator.

Recommended

Within a selection scope in which the user can move elements to arbitrary positions, use an element cursor within the scope instead of a graphics cursor if and only if the following are true:

- Automatic placement is acceptable for newly added elements to the scope
- Elements rarely need to be moved by the user
- Support is provided for keyboard–based drag and drop
- Area selection of more than a few elements is rare

Guidelines

Required

A control that uses a text or graphics cursor must keep track of the cursor position even when the control does not have focus.

Recommended

When designing new controls, use one of the existing cursor styles.

Mouse−Based Cursor Placement

Required

When the user clicks the SELECT button at a legitimate cursor position within a control that uses keyboard navigation to move the cursor, move the cursor to the location at which SELECT was clicked.

Required

When the user presses `Ctrl` SELECT at a legitimate cursor position within a control that uses keyboard navigation to move the cursor, move the cursor to the location at which SELECT was pressed, but do not perform any other action, unless pressing `Ctrl` SELECT already has a predefined meaning in that control.

Pointer Tracking

Required

Text and graphics cursors should not track the pointer; they should only move via keyboard navigation or when a mouse button is pressed, even when in a window that uses an implicit focus policy.

Required

An element cursor used within a control (other than a spring−sensitive control) should not track the pointer when in a window that uses an explicit focus policy.

Recommended

An element cursor used within a control should track the pointer when in a window that uses an implicit focus policy.

Traversal to Obscured Elements

Required

When using keyboard navigation, do not skip over an element solely because it is obscured by another window or by an icon in the workspace.

Recommended

If an element is obscured by other elements in the same window, allow keyboard navigation to it only if it can be made visible within the window.

Visual Guidelines

Required

If a control keeps track of a cursor position within the control even when the control does not have focus, do one of the following:

Make the cursor more visible when the control has focus.

Place an element cursor on the control as a whole when it has focus.

Required

If a control hides a cursor but keeps track of its position within the control when the control does not have focus, then when the control gains focus, show the cursor at the position it was in (relative to the underlying data) when the control last had focus, unless the application has changed the data or cursor position in the meantime.

Required

Within a selection scope that supports add mode and normal mode, change the appearance of the cursor, depending upon the mode. In particular, when an element cursor is used, draw a solid box around the element in normal mode and a dashed box around the element in add mode.

Required

Make sure that a text or graphics cursor used in a control is clearly distinguishable from the shape of the pointer when it is in that control.

Required

To make a cursor more visible when the control containing it has focus, use one of the following methods:

Darken the cursor when the control has focus and dim the cursor when the control does not have focus.
Use a blinking cursor when the control has focus and use a static cursor when the control does not have focus.
Show the cursor when the control has focus and hide the cursor when the control does not have focus (except when the cursor is being used to show interacted emphasis).

Recommended

If a control keeps track of a cursor position within the control even when the control does not have focus, make the cursor more visible when the control does have focus.

Recommended

Indicate when a graphics cursor is on an element, for example, by also showing an element cursor on the element.

Text Cursor Visual Guidelines

Required

When insert mode is being used, display a text cursor as a vertical bar or I–beam at the cursor position.

Required

The size of the I–beam cursor should be proportional to the current font.

Required

When replace mode is being used, use a block or underline to make the text cursor appear to be displayed on a character, though it actually is positioned immediately before that character.

Recommended

Make a cursor distinct from any other form of emphasis.

Supplemental Related Topics

For more information, see the Choice , Pointer , Scrolling , and Text–Entry Field (Control) reference pages.

Cut, Copy, Paste (Action Choice)

NAME

Cut, Copy, Paste (Action Choice) –Reference

Description

Cut, Copy, and Paste are action choices that are used in conjunction with the clipboard. The following describes each choice:

Cut

An action choice that removes selected elements and places them onto the clipboard. The space they occupied is usually filled by the remaining elements in the scope of selection.

Copy

An action choice that copies selected elements and places them onto the clipboard. The selected elements are unaffected.

Paste

An action choice that places the contents of the clipboard into a specified location or into an object or objects displayed within the window.

When to Use

Required

Provide a Cut choice for all elements and data.

Required

Provide a Copy choice for all elements for which the user can create a duplicate, and for data that the user can duplicate.

Required

Provide a Paste choice for all objects or elements that the user can place or insert into other objects or elements.

Guidelines

Cut

Required

When the user chooses Cut, remove the objects or data to which the choice applies from their current location and place them on the clipboard.

Required

Display the Cut choice in a pull–down menu with unavailable emphasis when the element to which the Cut choice applies cannot be cut.

Recommended

If you provide the Cut choice, provide the user with a warning message to confirm the requested action. For example, when the user cuts data from a spreadsheet, display a warning message that cutting the data will affect the remaining data.

Required

If you provide the Cut choice on an element, place Cut in the element's pop–up menu when it can be cut.

Required

Assign `CtrlX` or `Shift Delete` as the shortcut key combination for a Cut choice.

Copy

Required

When the user chooses Copy, copy the objects or data to which the choice applies from their current location to the clipboard.

Required

Display the Copy choice on a pull–down menu with unavailable emphasis when the element to which the Copy choice applies cannot be copied to the clipboard.

Required

If you provide the Copy choice for an element, place Copy in the element's pop–up menu when it can be copied.

Required

When the user copies an element to the clipboard, do not change the state of the source element. For example, do not change the selection state of the element as a result of copying it.

Required

When the user copies an element to the clipboard, do not change the state of the copied object on the clipboard.

Required

Assign `CtrlC` or `Ctrl Insert` as the shortcut key combination for a Cut choice.

Paste

Required

When the user chooses Paste, insert the objects or data to which the choice applies from the clipboard to the current cursor location.

Required

Display the Paste choice on a pull–down menu with unavailable emphasis when no data is in the clipboard or when the clipboard data cannot be pasted at the cursor's position.

Recommended

If you provide the Paste choice, provide the user with a warning message to confirm the requested action. For example, when the user pastes data into a spreadsheet, display a warning message that pasting the data will affect the remaining data.

Required

If you provide a Paste menu item for an element, place Paste in the element's pop–up menu when the element or data currently in the buffer can be passed into the element at the cursor's location.

Required

Assign `CtrlV` or `Shift Insert` as the shortcut key combination for a Paste choice.

Essential Related Topics

For more information, see the **Clipboard** and Edit (Menu) reference pages.

Supplemental Related Topics

For more information, see the Copy To (Dialog Choice) , Delete (Action Choice) , and Undo (Choice) reference pages.

Data Transfer

NAME

Data Transfer -Reference

Description

Data transfer operations transfer objects or data from a source to a target. The target may simply store the objects or data or may process them in some way as a result of the transfer.

There are five classes of data transfer operations:

Drag and drop

Drags source elements and drops them on the target to which they are transferred.

Primary

Transfers the primary selection to an indicated target.

Quick

Designates elements to be transferred to the control in which the user was most recently working.

Clipboard

Transfers elements to the clipboard and pastes them from the clipboard.

File

Transfers elements to a file (or object) and inserts them from a file (or object).

Each transfer operation is one of the following:

Copy operation

Transfers the data to the target, without affecting the source.

Move operation

Transfers the data to the target, removing it from the source.

Link operation

Transfers the data to the target (or simply places a link at the target) so that it remains linked to the source.

When to Use

Required

Allow any selectable element to be transferred.

Recommended

Provide support for users to transfer large labels that they might want to copy by using keyboard or mouse−based transfer. For example, a user might want to copy a complex icon or logo.

Guidelines

Required

A copy operation should transfer the source elements to the target without affecting the original source element in any way.

Required

A move operation should transfer the source elements to the target and then remove them from the source.

Required

A link transfer operation should do one of the following:

Transfer the source element to the target, keeping it linked (in an application−specific manner) to the original source element.

Place a link to the source element (made and displayed in an application−specific manner) in the target without copying the source element.

Insertion

Required

The insertion point for an operation that transfers data into a selection scope is determined as follows:

For a mouse−based TRANSFER button primary transfer or drag−and−drop transfer, the insertion point is the pointer position when the TRANSFER button is released.

For a transfer invoked via a menu (popped up via the MENU button) on a selection or object, the insertion point is the pointer position when the MENU button is pressed.

If neither of the previous applies, the pointer position is the cursor position.

If the insertion position is not on an object, then do one of the following:

If insertions can happen only at a fixed place in the selection scope, the insertion point should be placed at that point (for example, append−only text).

If the elements in the selection scope are ordered in some way, the insertion point should be moved (for each element inserted) so that the element may be inserted at the proper position.

Required

If the insertion point is on a selection, but not on an object, a transfer operation into the scope should have one of the following effects:

If a link operation, link the selection to the source elements, if such links are supported by your application.

If pending delete behavior is enabled, replace the contents of the selection by the transferred data or link (except possibly for a primary copy at the edge of the primary selection).

Insert the transferred data or link at the insertion point.

Required

If the insertion point is neither on a selection nor on an object, a transfer operation should insert the transferred data or link at the insertion point.

Required

If the elements are laid out in a linear manner, insert elements immediately prior to the insertion point.

Recommended

If the insertion point is on an unselected object, transfer data into that object (in an object−specific way; for example, if the object is text −based, append text to the end of the object).

Recommended

If the insertion point is on a selected object, transfer data into all the objects selected in the scope. However, issue a warning message instead if all selected elements in the scope are not objects.

Exclusions

Required

If the target insertion point is within the region of the source elements, do not perform the transfer; instead, issue a warning signal.

Required

If an operation that transfers data into a scope is selected from a menu popped up (via the MENU button) on neither an object nor a selection, issue a warning message if the insertion point is on an object.

Recommended

If a scope has elements selected, and a transfer operation uses an insertion point based on the position of the cursor (rather than the pointer) that is not within the selection, issue a warning message.

Rules

Required

When an object is moved or copied, transfer the contents of the object along with the object itself.

Required

When a move operation is performed, remove the source elements only if the data transfer has completed successfully. If the data transfer operation has been canceled or it has failed, do not remove the source elements.

Required

Do not select data transferred to a target as a result of a link or copy operation.

Required

If the source elements used in a data transfer operation are selected, do not deselect them as a result of the operation.

Required

Do not change the source elements used as the source of a copy or link operation.

Required

When transferred data is inserted into a selection scope (not into objects within the scope) by using normal mode at a position disjoint from a selection, deselect the selected elements.

Required

Inserting or pasting elements at a location disjoint from the selection should have no effect on the selection in add mode.

Required

If a transfer operation uses an insertion point based on the position of the pointer (rather than the cursor), place the cursor at the pointer position used.

Recommended

When a data transfer operation inserts multiple elements into a selection scope (except when a selection is being copied by primary or drag–and–drop transfer within the same scope), place the selection anchor at the opposite end of the area or range so that `Shift Select Shift Space` or `Shift Ctrl Space` in text can be used to select it when following is true:

Transferred elements are not selected

Data is inserted as an area or range, which does not overlap existing elements, one end of which corresponds to the position of the cursor when the operation is completed

Recommended

If a selection scope contains only objects, then transferring nonobject data to the scope should encapsulate the data into an object and include the object in the scope.

Recommended

When a list displays names of objects or files, provide an icon that represents the objects or files selected (or their contents) adjacent to the list, allowing the selected list elements to be treated as strings.

Recommended

Display a message asking the user to choose the desired result if the source elements of a data transfer represent files or objects and the user could be confused that more than one of the following could be inserted:

- The images of the icons representing the files or objects
- The names of the files or objects
- Links to the files or objects
- The data contained in the files or objects

Data Format

Required

During a transfer operation, allow the data transferred to change its appearance or representation to suit the requirements of the target.

Recommended

If a source cannot transfer data in the format preferred by the target, provide a warning message and allow the user to cancel the operation.

Recommended

On a move transfer operation, notify the user of possible loss of data if the target element indicates that data may be lost due to conversion of the format of the transferred data. Ask the user to confirm the action before proceeding. For example, if the user moves an ASCII file and the target element will not accept ASCII files without losing data, display a message to request confirmation.

Optional

If you provide a choice of the exact representation of data to be transferred, notify the user which representation the application will use. Alternatively, display a secondary window that allows the user to choose the representation.

Transfer Wells

A transfer well is a special field similar to a clipboard with option buttons that represent transferred elements.

Recommended

Provide a transfer well adjacent to the control that can take focus if a selection scope containing objects does either of the following:

- Uses an element cursor and the control containing the scope does not take focus as a whole
- Does not have a background area (for example, it is represented as a list)

Transferring objects to the target well includes them in the scope.

Optional

Use data transferred to transfer wells associated with selection scopes that consist primarily of text or graphics to replace the contents of the scope, or as input to operations that manipulate the contents of the scope.

Recommended

If your application supports transfer to a transfer well as a means of loading a file into the application, design the application to respond to this operation as if the operation had occurred through more conventional means, such as when the user chooses Open from the File menu. That is, provide the same kind of feedback and behavior for data transfer loading of files as when the Open choice is used. For example, if changes to a currently loaded file have not been saved, display a message asking whether the changes should be saved before loading the new file.

Transferring Groups of Elements

Recommended

If a group of elements cannot all be transferred successfully, and the destination can determine this before any transfers are performed, allow the user to set a property that determines which of the following actions to take:

- Do not transfer any of the elements (the default).

Transfer all the elements that can be transferred.
Display a message asking the user what to do. Include buttons with the following choices:
Resume (transfer all possible elements)
Cancel (do no transfers)
Help

Recommended

If a group of elements cannot all be transferred successfully, but the destination cannot determine this before at least one transfer has been performed, allow the user to set a property that determines which of the following actions to take:
Stop the transfer and undo any transfers that have already occurred (if possible).
Stop the transfer, but allow already completed transfers to remain (the default).
Transfer only the elements that can be transferred.
Display a message asking the user what to do.
Include buttons with the following choices:
Resume (transfer all possible elements)
Stop (stop transferring, but do not undo transfers)
Cancel (undo all transfers)
Help

Transfer Option

Required

Allow the user to override the default operation of a TRANSFER button–based data transfer operation by using a modifier key.
TRANSFER Button Overrides lists these operations and overrides.

Table 1 TRANSFER Button Overrides

Operation	Modifier Override
Move	`Shift`
Copy	`Ctrl`
Link	`Ctrl Shift`

Essential Related Topics

For more information, see the Clipboard, Direct Manipulation, and Primary Transfer reference pages.

Supplemental Related Topics

For more information, see the Copy To (Dialog Choice) and Pop–Up Menu (Menu Type) reference pages.

Default Action

NAME

Default Action –Reference

Description

The default action is the action associated with a window that the user would most likely want to invoke in a given situation when focus is in that window. The default action may change as the focus and state of the window changes. The default action is generally activated when a user double–clicks the SELECT button, presses `Enter` (except when either operation is used for other purposes), or presses `Ctrl Enter` or `keypadEnter`.

When to Use

Recommended

If a user will want to perform a specific action at a specific time, make it the default action.

Recommended

If you use a secondary window for some interaction, provide a default action in that window.

Guidelines

Required

To support new users and keyboard–only users, provide default actions as shortcuts only for application features that are available by some other method.

Required

When a window has focus, show default emphasis on the push button in the window, if any, that performs the current default action for that window.

Required

Except when focus is on a tool or object or in a menu, if the default action of a window varies, provide a push button that corresponds to each default action.

Recommended

Except when focus is on a tool or object or in a menu, provide a push button that corresponds to each default action.

Recommended

When an explicit focus policy is in use and the focus is outside the window, use default emphasis to indicate the push button that would perform the default action if focus were moved to the window by using window navigation.

Default Activation

Required

When the focus is on a push button, make its action the default action.

Required

When the focus is on a choice in a menu other than a tear–off menu, make the default action activate or toggle that choice.

Required

When focus is on an object icon, the default action should present data of the corresponding object; for a window icon display the corresponding window.

Default Action Behavior

Required

When the user presses `keypadEnter`, `Ctrl Enter`, or `Enter` (except in text), invoke the default action. Except possibly when on an element that is both activatable and selectable, it should have no other effect.

Required

When the user double−clicks the SELECT button and focus is on a value choice that is part of a mutually exclusive group of value choices (for example, on a radio button, a tool, or in a value set), invoke the default action.

Optional

When the user double−clicks the SELECT button on a cascading choice that leads to a menu (but does not press `Enter` or `keypadEnter`), you may invoke the action of the menu's default choice. In particular, double−clicking the SELECT button on the window menu button can be used to close the window.

Essential Related Topics

For more information, see the **Push Button (Control)** and Push Button (Predefined) reference pages.

Supplemental Related Topics

For more information, see the Emphasis (Cue) and Window Menu reference pages.

Delete (Action Choice)

NAME

Delete (Action Choice) –Reference

Description

The Delete choice includes the following:

Delete

An action choice that removes selected elements. The space they occupied is usually filled by the remaining elements in the scope of selection.

Delete to Trash

An action choice that removes selected elements (generally objects) and puts them into the trash. The space they occupied is usually filled by the remaining elements in the scope of selection.

When to Use

Recommended

Provide a Delete choice if elements can be removed. Depending on the type of data and view presentation, either compress or do not affect the window previously occupied by the deleted element or group of elements.

Guidelines

Required

If only objects are selected, the Delete choice should not be available in the Edit menu. Provide Delete to Trash in the Selected menu instead. If the selection scope includes only objects, then do not provide Delete in the Edit menu.

Recommended

If the elements selected represent links or references to objects and you want the user to be able to destroy the underlying object, do one of the following:

Provide separate choices (for example, Delete or Delete to Trash).
Make Delete to Trash a cascaded entry through which the user can specify whether to delete the links or references or the underlying objects.

Region Deletion

Required

In an editable selection scope, any operation that deletes a region (not containing objects) has the following effect if something is currently selected in the scope:

In normal mode

Deletes the selection.

In add mode

If the cursor is not outside of the selection, deletes the selection.

In add mode

If the cursor is outside of the selection, deletes the specified region, possibly including part of the selection.

Required

In an editable selection scope, pressing `Delete` and `Backspace` have the following effects if the cursor is within text and the selection is not empty and contains no objects:

In normal mode

Deletes the selection.

In add mode

If the cursor is not outside of the selection, deletes the selection.

In add mode

If the cursor is outside of the selection, pressing `Delete` deletes the character following the cursor and pressing `Backspace` deletes the character preceding the cursor.

Required

When nothing is selected in an editable selection scope, and there is a character following the text cursor, pressing `Delete` deletes it.

Required

When nothing is selected in an editable selection scope, and there is a character preceding the text cursor, pressing `Backspace` deletes it.

Essential Related Topics

For more information, see the **Clear (Choice)** and Edit Menu reference pages.

Supplemental Related Topics

For more information, see the Menu Guidelines reference page.

Dialog (Choice Type)

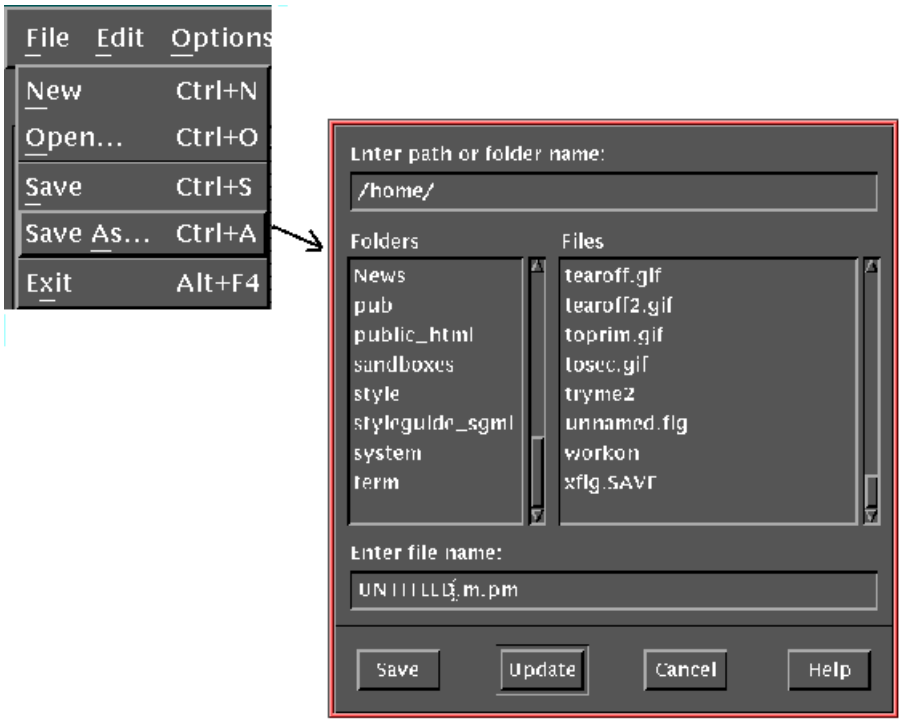
NAME

Dialog (Choice Type) -Reference

Description

Dialog is a type of choice used to display a dialog window to gather parameters for invocation of an action.

Figure 1 Dialog Choice



When to Use

Required

Use a dialog choice to display a secondary window that contains a dialog from which the user can specify parameters and invoke the action corresponding to the choice or cancel the action.

Guidelines

Required

Use ellipses (...) after the textual label of a dialog choice.

Required

Use ellipses after a choice in a menu or on a push button that causes a secondary window to be displayed, if the secondary window gathers additional information required to complete the requested task. For example, provide a Print... choice to allow the user to specify print characteristics before starting the print operation.

Required

Do not use ellipses if a choice results in immediate action. In some cases, the action may be to display a secondary window that supports interaction with the user. For example, do not use an ellipsis on the Help choice, even though the help information is displayed in a secondary window.

Required

Do not use ellipses simply because a confirmation message is displayed.

Essential Related Topics

For more information, see the **Choice** reference page.

Supplemental Related Topics

For more information, see the Action (Choice Type), Help (Menu/Action Choice), Menu (Control), Push Button (Control), and Secondary Window reference pages.

Dialog (Window)

NAME

Dialog (Window) –Reference

Description

A dialog includes the following:

Dialog

An interaction between user and application conducted by using one or more secondary windows, and that allows a user to specify or review parameters for an action to be performed.

Dialog window

A secondary window used for a dialog.

When to Use

Required

Use a dialog to allow the user to specify parameters to an action that is to be performed.

Required

Use a dialog to allow the user to review the parameters to an action before finally approving its invocation.

Recommended

Use a dialog with multiple secondary windows to allow the user to review and approve the results of intermediate steps of a complex action.

Guidelines

Required

Provide one or both of the following push–button choices in a dialog:

OK

Invokes the action and closes the dialog window.

Apply

Invokes the action and leaves the dialog window open.

You can replace the labels of these choices with ones that are descriptive of the action to be performed.

Required

Provide a Cancel choice in a dialog window that closes the dialog without performing the effects of the action.

Optional

If a task analysis shows that the user will fill the fields of a form–filling dialog in order, then when using explicit focus and focus is in any field but the final one, make the default action move focus to the next field. When focus is in the final field, the default action submits the form and closes the window that contains the dialog.

Essential Related Topics

For more information, see the **Secondary Window** reference page.

Essential Related Topics

For more information, see the Choice reference page.

Direct Editing

NAME

Direct Editing -Reference

Description

Direct editing is a mode of interaction in which the contents of a normally noneditable textual label is made available for editing.

When to Use

Recommended

Support a direct editing mode if the user wants to modify a textual label.

Recommended

Support direct editing of any label that names a selectable element.

Guidelines

Recommended

While in direct editing mode, show focus emphasis on the label being edited.

Recommended

Treat the textual label as a text−entry field when operating on it in direct editing mode.

Entering Direct Editing Mode

Required

If direct editing mode is supported for a textual label, enter the mode when the user clicks the SELECT button on the label or when the user double−clicks the SELECT button, if (single) clicking the SELECT button is already used for some other purpose (for example, margin selection).

Required

If direct editing mode is supported for a textual label, provide some means for keyboard users to edit the label as well (for example, by choosing an action from a menu that displays a window in which such labels can be edited via a text−entry field).

Required

Use a persistent selection to select the entire text of the label when entering direct editing mode.

Recommended

If you support direct editing of a label that names a selectable element, associate a pop−up menu with the element, include a Properties choice in the pop−up menu, and include a text−entry field in the properties window through which the label can be changed.

Recommended

When focus is on a static text control, provide a means to enter direct editing mode from the keyboard, for example, by including a Rename choice on a pop−up menu that initiates direct editing.

Exiting from Direct Editing Mode

Required

Exit from direct editing mode when the user presses `Enter` or `keypadEnter`.

Required

Exit from direct editing mode and undo any changes made to the text being edited when the user presses `Cancel`.

Required

When the user exits from direct editing mode, deselect the text in the label.

Required

When the user exits from direct editing mode without explicitly moving focus elsewhere, and the focus remains in the window or is moved back to it, place the active cursor as follows:

If a static text control is being directly edited, place the active cursor on it.
If the label being directly edited is associated with all or part of another element (for example, an icon) that can be cursored with an element cursor, move the active cursor to that element.
If the label being directly edited is a group heading, navigate to the group.
If none of the previous statements apply, the active cursor should return to where it was before entering direct editing mode.

Required

During direct editing, mouse buttons have their usual effect, and all operations on other controls except those that transfer data to the label cause direct editing mode to exit.

Required

During direct editing, keyboard–based window navigation should exit from direct editing mode and move focus to the window indicated.

Required

During direct editing, control navigation should not be supported.

Required

During direct editing, tab–group navigation should exit from direct editing mode and allow navigation elsewhere, but only in the following cases:

When a static text control is being directly edited, tab group navigation should allow navigation in the same way as when the control ordinarily has focus.
If the label being directly edited is associated with all or part of another element (for example, an icon) that can be cursored with an element cursor, tab navigation should tab away from that element.
If the label being directly edited is a group heading, allow tab navigation to tab away from the group.

Essential Related Topics

For more information, see the **Label** and Selection reference pages.

Supplemental Related Topics

For more information, see the Choice , Push Button (Control) , and Text–Entry Field (Control) reference pages.

Direct Manipulation

NAME

Direct Manipulation –Reference

Description

Direct manipulation is the use of the pointer and a pointing device (such as the mouse) as a metaphorical extension of a hand (for example, as a handheld tool) to point to and manipulate elements that represent objects or to change the state of the application.

When to Use

Recommended

> Support direct manipulation for all objects and for all elements that the user can act on or that represent an application state that the user can change.

Required

> Support direct manipulation of the objects or state represented by the element directly under the pointer or, if the pointer is within the area of a selection, the selected elements.

Guidelines

Initiating the Operation

Required

> Initiate a direct manipulation operation when the user presses a pointer button on an element.

During the Operation

Required

> Do not change the active window while a direct manipulation operation is in progress.

Required

> While a direct manipulation operation is in progress, if the action to be performed on termination changes as the pointer is moved or as the state of the keyboard changes, provide appropriate feedback to the user so that the user knows what action will be performed.

Terminating the Operation

Required

> If a direct manipulation operation requires that buttons be pressed while the operation is in progress, terminate the direct manipulation operation. Alternatively, change its mode when the user releases all pointer buttons and does not press any more buttons or keys within a time period specified by the operating environment.

Required

> If a direct manipulation operation does not require that any button be pressed while the operation is in progress, allow the user to terminate the direct manipulation operation. Alternatively, change its mode by pressing shortcut keys or by clicking on some combination of pointer buttons.

Required

> When the user presses `Cancel` or `Esc` while a direct manipulation operation is in progress, cancel the direct manipulation operation.

Recommended

Do not perform any action associated with the operation if the user performs a direct manipulation operation by pressing one or more pointer buttons, moves the pointer but returns it to its starting location, and then releases the buttons, and if there was no change in mode during the motion.

Providing Help During the Operation

Recommended

Display a Help window when the user presses `Help` while a direct manipulation operation is in progress. The Help window should provide help specific to the operation, the application state, and the pointer position. Include information describing what would happen if various inputs were received at that point and how and why the operation could succeed or fail.

Recommended

If you provide a Help window when the user requests Help during a direct manipulation operation, make it an application modal window and do not accept any input until all pointer buttons or keys that are still pressed are released.

Recommended

Support the following choices in a Help window displayed during a direct manipulation operation:

Cancel, to terminate the direct manipulation operation.

Choices that provide more help.

Choices to complete the direct manipulation operation in various ways. For instance, if help is requested while scrolling, the help dialog could provide choices to scroll to the beginning of the viewed object, the end of the viewed object, or to the position just before or after the initially displayed view.

Resume, to resume the direct manipulation operation, but only if the direct manipulation operation does not require that any pointer button be pressed while the operation is in progress.

Optional

Display help information, but not in a secondary window, when the user presses `Help` and continues to press any other pointer buttons or keys that were pressed. Remove the help information when the user releases the `Help` key.

Essential Related Topics

For more information, see the **Cancel (Choice)**, Copy To (Dialog), Drag–and–Drop Transfer, Emphasis (Cue), Help (Menu/Action Choice), Move To (Dialog Choice), Pointer, and Window Menu reference pages.

Supplemental Related Topics

For more information, see the Data Transfer, Keyboard (Device), and Mouse (Device) reference pages.

Drag–and–Drop Transfer

NAME

Drag–and–Drop Transfer –Reference

Description

Drag–and–drop transfer is a user interaction in which a user drags source elements to a target element on which they are dropped.

When to Use

Required

Support dragging any object represented as an icon.

Required

Support dropping onto objects represented as icons.

Required

If a control supports any form of data transfer with its contents used as a source, then those contents can be the source of a drag–and–drop operation.

Required

If a control allows any form of data transfer to the control, it can be used as a target for a drag–and–drop operation.

Guidelines

Required

Make the source of a drag–and–drop transfer the element or selection on which the drag was initiated.

Required

Make the target of a drag–and–drop transfer the control on which the drop was performed; if the control holds a selection scope, then it is the insertion point within that scope.

Required

The default operation for a drag–and–drop transfer must be a move, except in the following cases:

When the source is not editable, make copy the default.

When the destination is an external device (for example, a printer, diskette, or mail out basket), make copy the default.

When the destination is an application that manipulates the data (for example, a compiler), consider making copy the default.

Required

The default operation for a drag–and–drop transfer may be overridden by `Ctrl` and/or `Shift` modifiers when the TRANSFER button is released.

Recommended

Allow users to control whether nonselectable elements such as labels can be dragged or not.

Recommended

Make any basic function available through a drag–and–drop transfer also available through predefined shortcut keys, menus, push buttons, or secondary windows. That is, provide drag–and–drop transfer as a shortcut to functions that are accessible through other interface

controls supported within your application. Do not provide any basic operation that is supported solely through a drag−and−drop transfer.

Separated Transfer

Use the following guidelines when TRANSFER is a separate mouse button from SELECT.

Required

Initiate a drag when the user presses the TRANSFER button and then moves the mouse on selections and elements that can be dragged.

Required

If the user uses the TRANSFER button to drag an element that has not been selected, drag only that element without affecting any existing selection.

Required

Drag all selected elements in the scope when the user presses the SELECT button and then moves the mouse on a selection.

Required

Initiate a drag when the user clicks `Alt` TRANSFER on a selection or an element that can be dragged.

Integrated Selection and Transfer

Required

Whether the TRANSFER button is integrated with SELECT on MB1 or not, when the user clicks SELECT, optionally augmented with `Shift` and/or `Ctrl`, the effect should be the same.

Required

When the user multi−clicks (or multi−presses) the SELECT button, optionally augmented with `Shift Ctrl`, the effect should be the same whether the TRANSFER button is integrated with SELECT on MB1 or not.

Use the following guidelines when the TRANSFER button is the same button as SELECT.

Required

When the user presses MB1 and moves the pointer on a selection, drag all selected elements in the scope.

Required

When the user presses MB1 on an element that can be activated or toggled (other than a tear−off choice), activate or toggle the element; do not allow dragging.

Required

When the user presses MB1 and moves the pointer on an element that cannot be selected (other than a tear−off choice), do not allow dragging.

Required

If the MENU button is bound to a mouse button other than MB1, when the user clicks `Alt` MB1, optionally augmented with `Shift` and/or `Ctrl`, on a selection or an element that can be dragged, initiate a drag.

Integrated Selection and Transfer in the Background

Required

Whether the TRANSFER button is integrated with SELECT on MB1 or not, support the same effect when the user presses SELECT and moves the pointer, optionally augmented with `Shift` and/or `Ctrl`, in the background (or on unselected text) of a scope that supports area or range selection techniques.

Optional

When the TRANSFER button is integrated with SELECT on MB1 in a nontextual scope that supports neither area nor range selection techniques, pressing MB1 and moving the pointer, optionally augmented with `Shift` and/or `Ctrl`, in the background can be used either for dragging the entire selection scope or for panning.

Integrated Selection and Transfer on Unselected Elements

Required

When the TRANSFER button is integrated with SELECT on MB1, when the user presses MB1 and moves the pointer, optionally augmented with `Shift` and/or `Ctrl`, on an unselected element for dragging, the effect should be equivalent to the user clicking the SELECT button (without modifiers) on the element (using whatever selection model is in force) and then dragging the selection.

Required

When the TRANSFER button is integrated with SELECT on MB1 in a nontextual scope, treat pressing MB1 and moving on an unselected element as a selection technique if the following is true:

Area or range selection techniques are supported and the scope is densely populated (in other words, there is not much background space available)

Different selection techniques are used (for example, touch versus area techniques) depending upon whether MB1 is pressed on an element versus in the background

Otherwise, allow the user to press MB1 and move the pointer on an unselected element for dragging.

Recommended

When the TRANSFER button is integrated with SELECT on MB1, then treat pressing `Shift` MB1 and moving the pointer on an unselected element as a selection technique only if adjustment selection techniques can be used in the scope and if the scope is densely populated. Otherwise, allow pressing `Shift` MB1 and moving the pointer for dragging.

Recommended

When the TRANSFER button is integrated with SELECT on MB1, treat pressing `Ctrl` MB1 and moving the pointer on an unselected element as a selection technique only if group selection techniques can be used in the scope, if `Ctrl` forces use of toggle mode instead of select mode, and if the scope is densely populated. Otherwise, allow pressing `Ctrl` MB1 and moving the pointer for dragging.

Recommended

When the TRANSFER button is integrated with SELECT on MB1, allow the user to press `Ctrl Shift` MB1 and move the pointer on an unselected element for dragging.

During a Drag Operation

Recommended

If the drag pointer's source indicator consists of a graphic image of the source and if the operation is a move, hide the source, especially if the operation indicator is used.

Recommended

When a user pauses on an icon (for a time that the user can set) during a drag–and–drop operation, display and raise the window used to view the corresponding object.

Completing the Drag

Required

The application that contains the source elements of a drag–and–drop operation may limit drops to windows that it manages. It may also treat elements of its own or other applications as invisible for the purpose of determining the target.

Required

When a drag is completed, a drop should be performed at the pointer location as follows:

Where the pointer is when the TRANSFER button is released, if it was pressed while the drag was in progress

Where the pointer is when the TRANSFER button is clicked, if it was not pressed while the drag was in progress
Where the user presses `Return` or `Enter`

In any case, the state of the `Ctrl` and `Shift` modifiers at the time of the drops determines the operation to be performed.

Required

When a drag is in progress and the user presses `Cancel`, cancel the drag.

Recommended

When a drag is in progress and the user presses `Help`, post a Help dialog that describes the result of a drop at the pointer location. The Help dialog should allow the drop to be performed (if it would likely succeed) or canceled (via a `Cancel` choice) or additional Help to be obtained (via a Help choice).

Recommended

If different operations can be performed as a result of a drop (for example move, copy, or link), provide push buttons labeled with all the choices in the Help dialog.

Optional

The Help dialog can contain a Resume choice if the drag can be resumed.

Dropping an Element

Required

When the user attempts to drop an element on an element that does not support the operation, cancel the drag or display a message dialog similar to the one that would be used if `Help` were pressed at that location.

Required

When a user drops a source element onto the same position it currently occupies, the application should not perform an action.

Required

If the source elements of a drag−and−drop move operation are selected, select the elements transferred to the target scope if the source and target scopes are the same.

Required

If the source elements of a drag−and−drop move operation are selected, select the elements transferred to the target scope if the target uses normal mode and allows the number of elements transferred to be selected.

Required

Move focus to the target control in a drag−and−drop operation.

Recommended

If the user drops an element at an inappropriate drop site, design your application to display a "snap back" effect and to display an error message indicating the reason the drop was not allowed. Within the error message, state the context of the error (for example, trying to perform action A on object B), what happened (for example, could not connect to system X), and how to correct the problem (for example, direct the user to press the Help push button for information).

Recommended

When the user selects the target of a drag−and−drop link operation, make a link (in an application−specific manner) from the selected elements to the source elements.

Essential Related Topics

For more information, see the **Context−Sensitive Help** , Copy To (Dialog Choice) , Drag/Move (Action Choice) , Direct Manipulation ,

Emphasis (Cue) , Move To (Dialog Choice) , Pointer , and Window Menu reference pages.

Supplemental Related Topics

For more information, see the Data Transfer , Keyboard (Device) , Mouse (Device) , Pointer (Predefined) , and Pop–Up Menu (Menu Type) reference pages.

Drag/Move (Action Choice)

NAME

Drag/Move (Action Choice) -Reference

Description

The drag/move action choices include the following:

Drag

An action choice that initiates drag and drop of the file or object being viewed or of specified elements within the window.

Drag (in the File menu)

Initiates a drag of the underlying object or file being viewed in this window.

Drag (in the Edit menu)

Initiates a drag of the selected elements.

Move

Has the following definitions:

An action choice that allows the user to move a window
An action choice that initiates drag and drop, allowing a user to move specified elements within the window
An action choice used in the context of a specific transfer mechanism to indicate that the transfer should result in a move

Move (in the window menu)

Moves the window or icon with which the menu is associated.

Move (in the File and Edit menus)

Has the same effect as Drag (in the File and Edit menus).

When to Use

Required

Include Move in the window menu. When used from a window, move the window, and when used from a window icon, move the icon.

Recommended

Include Drag in the File menu to support drag and drop of the file or object being viewed.

Recommended

Include Drag or Move in the Edit menu to allow use of drag and drop for selected elements without requiring use of the mouse buttons.

Guidelines

Required

In the File and Edit menus, use Move if the selected elements can only be moved; use Drag if other operations are possible as well.

Required

When the user is moving a window, the only visible target should be the workspace. That is, a window can be dropped anywhere, but should always be treated as being dropped on the workplace, not on some other element.

Required

When the user initiates a drag−and−drop operation by selecting Drag or Move from a menu, and the pointer position after choosing the menu entry is outside the bounding box of the element or elements being moved, or pointer warping is not enabled, then do one of the following:

Start the drag at the current pointer position and drag the elements there.

Either wait until the user moves the pointer into the bounding box (within the window) of the elements being moved and start dragging them at that point, or wait until the user presses the directional keys to move the elements.

Recommended

When a drag and drop transfer is initiated by selecting Drag or Move from a menu and pointer warping is enabled, warp the pointer to an appropriate location according to the following:

If the source indicator of the drag pointer consists of a graphic smaller or dissimilar to the elements being moved, warp the pointer to the center of the elements.

If the drag pointer consists solely of an image or outline of the elements being moved or uses a state and/or operation indicator placed at an arbitrary position relative to them, warp the pointer to the center of the elements.

If the drag pointer consists of a state and/or operation indicator placed at a corner or edge of an image or outline of the elements being moved, warp the pointer to that corner or edge of the elements.

Essential Related Topics

For more information, see the **Drag−and−Drop Transfer** , Edit (Menu) , File Menu , and Pointer reference pages.

Supplemental Related Topics

For more information, see the Window Menu reference page.

Drop–Down Combination Box (Control)

NAME

Drop–Down Combination Box (Control) –Reference

Description

A drop–down combination box is a combination text–list control in which the text field is a text–entry field that is always visible, but the list box is hidden until the user performs an action to display it. Drop–Down Text–Entry Field illustrates a drop–down text–entry field and Drop –Down List illustrates a drop–down list.

Figure 1 Drop–Down Text–Entry Field

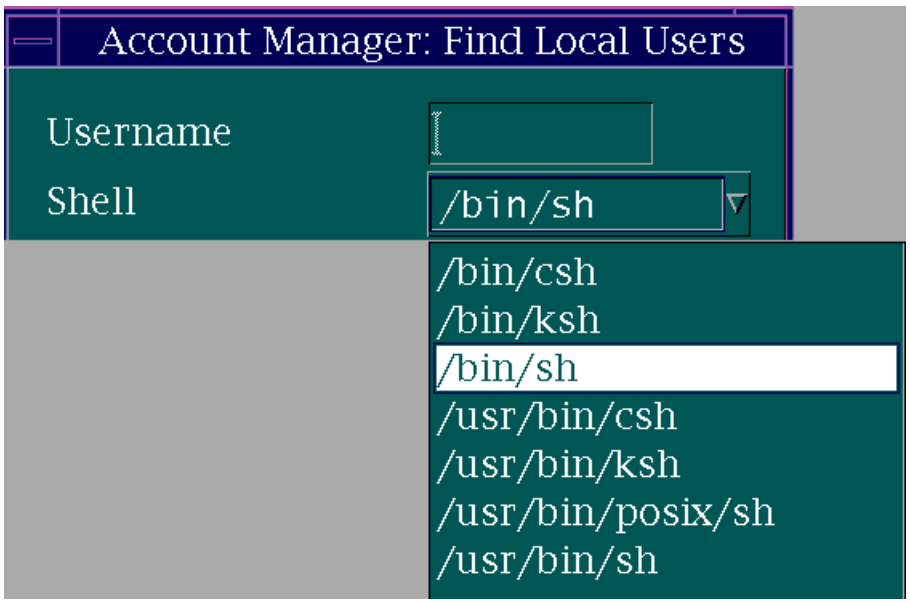
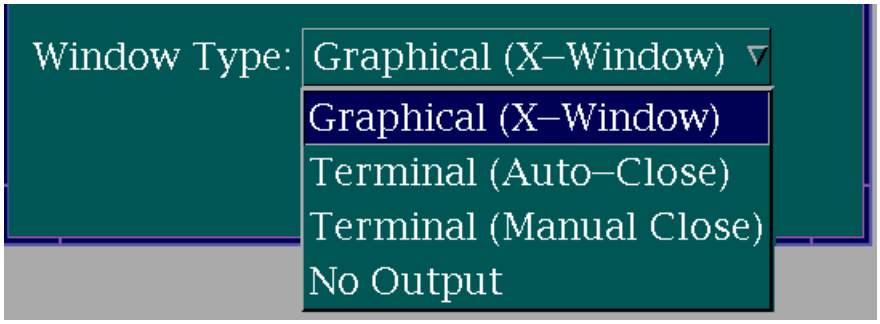


Figure 2 Drop–Down List



When to Use

Recommended

Use a drop–down combination box when a window does not have enough space to contain a combination box.

Guidelines

Behavior

Required

If the drop–down combination box has focus and the list box is displayed, remove the list box if the user presses `Cancel` or `Escape` and leave focus in the text–entry field.

Required

Do not allow the list box in a drop–down combination box to take focus.

Essential Related Topics

For more information, see the Combination Box (Control), Combination Text–List Control (Control Type), Control, Drop–Down List (Control), and List Box (Control) reference pages.

Supplemental Related Topics

For more information, see the Emphasis (Cue) and Text–Entry Field (Control) reference pages.

Drop–Down List (Control)

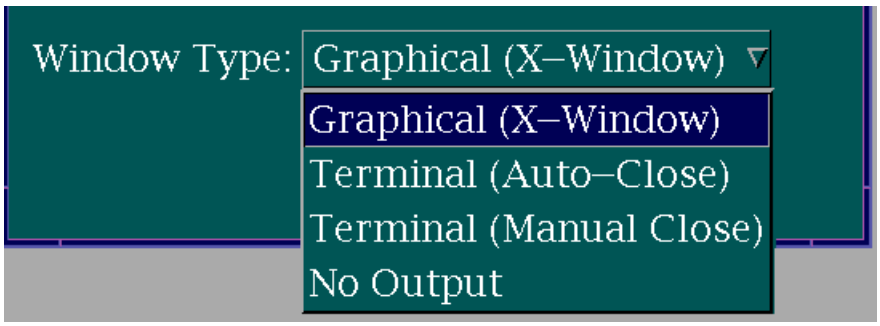
NAME

Drop–Down List (Control) –Reference

Description

A drop–down list is a combination text–list control in which the text field is a text–display field that is always visible, but the list box is hidden until the user performs an action to display it.

Figure 1 Drop–Down List



When to Use

Recommended

Use a drop–down list instead of a list box when the selected item is not changed frequently or when space is so limited that the window does not have enough space to display a list box.

Guidelines

Recommended

Display an initial value from the list box in the text–display field.

Required

If the drop–down combination box has focus and the list box is displayed, remove the list box if the user presses `Cancel` or `Escape` and leave focus in the text–display field.

Behavior

Required

Do not allow the list box in a drop–down list to take focus.

Essential Related Topics

For more information, see the Combination Box (Control), Combination Text–List Control (Control Type), Control, Drop–Down Combination Box (Control), First–Letter Cursor Navigation, and List Box (Control) reference pages.

Supplemental Related Topics

For more information, see the Keyboard (Device) , Mouse (Device) , and Scroll Bar (Control) reference pages.

Edit (Menu)

NAME

Edit (Menu) -Reference

Description

The Edit menu includes the following choices:

Edit

A cascading choice that appears as a menu−bar item. It provides access to other menu items that allow a user to modify the contents of a selection scope.

Duplicate

An action choice that duplicates selected objects and displays the duplicates in the same selection scope.

Insert

This choice can be:

A dialog choice

Leads to a file selection dialog in which users can select a file or object to be inserted into a specified location or into an object or objects displayed within the window.

A cascading choice

May appear as a menu−bar item. It provides access to other choices that enable a user to insert elements into a specified location or into an object or objects displayed within the window.

Select All

An action choice that causes all of the elements in a scope of selection to be selected.

Deselect All

An action choice that deselects the selected elements (possibly excluding the cursored element) in the scope of selection.

Guidelines

In conjunction with the Menu Guidelines reference page, use Edit Menu Choices to decide which choices to include in the Edit menu and how to organize them.

Table 1 Edit Menu Choices

Menu Choice	Mnemonic	Keyboard Function	
Recommended	U	Undo	Undo, `Ctrl Z`, `Alt Backspace`
Recommended	R	Repeat/Redo	`Shift Undo`, `Ctrl Y`, `Alt Shift Backspace`
Recommended		Drag	
Optional		Move (only if Drag is not used)	
Optional		Size	
Optional		Duplicate	
Required	T	Cut	Cut, `Ctrl X`, `Shift Delete`
Required	C	Copy	Copy, `Ctrl C`, `Ctrl Insert`
Optional	K	Copy Link	
Optional		Copy Special ...	
Required	P	Paste	Paste, `Ctrl V`, `Shift Insert`

Optional	L	Paste Link	
Optional		Paste Special ...	
Optional	I	Insert ...	
Optional		Primary Copy	`Alt Copy`, `Alt Ctrl C`, `Alt Ctrl Insert`
Optional		Primary Move	`Alt Cut`, `Alt Ctrl X`, `Alt Shift Delete`
Optional		Primary Link	
Recommended	E	Clear	Delete (only if Delete entry not used)
Required	D	Delete	Delete
Recommended	S	Select All	`Ctrl A`, `Ctrl /`
Recommended		Deselect All	`Ctrl \`
Optional	`Ctrl F`	Find...	
Optional		Replace....	
Optional		Select Pasted	
Optional		Reselect	

Required

For operations with multiple shortcut keys, show the one that augments a printing character with `Ctrl` (for example, `Ctrl A` for Select All).

Required

Do not display the shortcut keys for Primary Copy or Primary Move in their respective menu entries.

Optional

If all Edit menu choices apply to the selected elements, the Edit menu is not needed, and you can place all choices in the Selected menu.

Guidelines for Specific Menu Entries

Required

When the user selects the Select Pasted choice, select the element last transferred into the selection scope.

Required

Allow a Reselect choice only when nothing is selected in the current selection scope. When the user chooses Reselect, reselect the previously selected elements within a selection scope.

Required

Provide a Duplicate choice to copy the selected elements in the same selection scope. If the elements are named, selecting Duplicate should automatically generate names for them that do not conflict with the names of other objects in the scope.

Required

Provide an Insert choice to allow the user to specify a file or object to be included at the cursor position. Either the name, reference, or contents can be included. If multiple choices are reasonable, the dialog should allow the user to determine the choice. If different kinds of elements can be inserted, then either provide a cascaded choice or make Insert a menu–bar item.

Essential Related Topics

For more information, see the Clear (Action Choice), Copy (Choice), Cut (Choice), Delete (Action Choice), Paste (Choice), and Undo (Choice) reference pages.

Supplemental Related Topics

For more information, see the Help Menu and Selected Menu reference pages.

Emphasis (Cue)

NAME

Emphasis (Cue) -Reference

Description

Emphasis is highlighting, color change, or some other visual indication of the condition of an element or choice, and the effect of that condition on the user's ability to interact with it. Emphasis can also give the user additional information about the state of an element or choice.

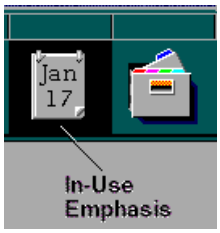
The types of emphasis are:

- In−use
- Selected
- Unavailable
- Source
- Target
- Ready
- Default
- Interacted
- Focus

In−Use Emphasis (Cue)

In−use emphasis is a visual cue that indicates that an object is in use, for example, if a view of the object is being displayed in a window.

Figure 1 In−Use Emphasis



When to Use

Recommended

When a user opens a window on an object represented by an icon, display in−use emphasis on the icon.

Recommended

If the user invokes a task that uses an object represented by an icon, display in−use emphasis on the icon.

Guidelines

Required

Remove in−use emphasis from an icon for an object when all windows that contain a view of that object have been closed and all tasks that use that object have completed.

Recommended

Display in−use emphasis on each instance of an icon for an object.

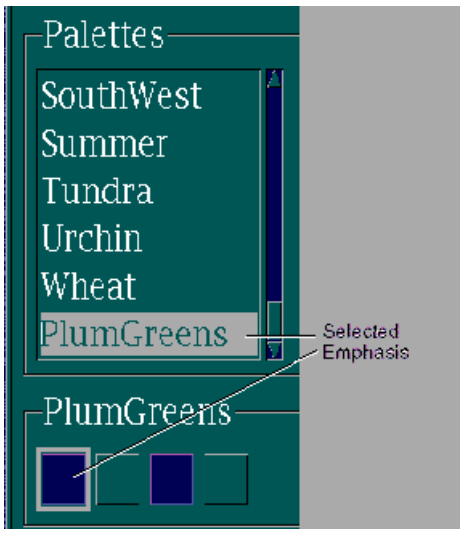
Supplemental Related Topics

For more information, see the **Icon** reference page.

Selected Emphasis (Cue)

Selected emphasis is a visual cue that indicates that an element is selected.

Figure 2 Selected Emphasis



When to Use

Required

Display selected emphasis on the primary selection, whether or not the control containing it has focus.

Required

Display selected emphasis on a persistent selection when the control containing it has focus.

Recommended

Display selected emphasis on a nontextual persistent selection, whether or not the control containing it has focus.

Guidelines

Visual Guidelines

Recommended

Display selected emphasis by changing the foreground and background colors of the selected element to those specified by the operating environment.

Recommended

Design the rendition graphics for elements that use a variety of colors to have distinct background pixels, preferably around the perimeter of the image.

Recommended

If your application supports a secondary selection mechanism in addition to a primary selection (for example, during quick transfer), display selected emphasis for the secondary selection by using the foreground color to draw a box around a graphic element or to underline a text element.

Optional

If a selected element uses a variety of colors in its rendition graphic, display selected emphasis by changing the background pixels to the background color specified by the operating environment.

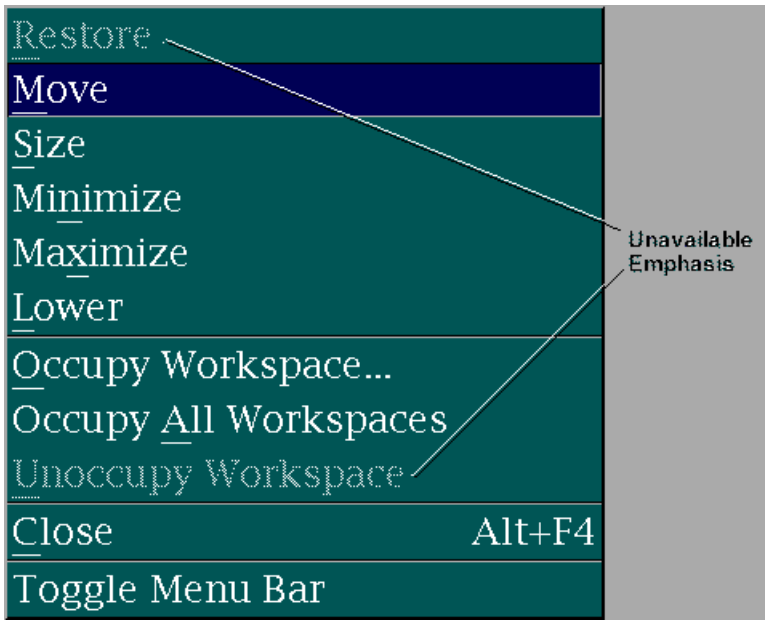
Essential Related Topics

For more information, see the Selection reference page.

Unavailable Emphasis (Cue)

Unavailable emphasis is a visual cue that indicates that a choice or control is not available. Unavailable emphasis is sometimes called *''dimmed emphasis''* or *''dimming''* in some environments.

Figure 3 Unavailable Emphasis



When to Use

Required

Display unavailable emphasis on choices or controls that represent operations that cannot be activated in the current context.

Guidelines

Required

If a choice is never available to a particular user, do not display the choice rather than displaying it with unavailable emphasis. For example, if the system administrator assigns read–only access to a user, destructive choices such as Cut or Delete should not be displayed.

Required

Do not include unavailable choices in lists and option menus.

Recommended

Provide a warning signal when a user attempts to interact with a choice or control that is displayed with unavailable emphasis.

Recommended

When the contents of a list are variable from one presentation to the next, such as a list of documents, avoid displaying them with unavailable emphasis; instead, do not include choices in the list that are unavailable.

Recommended

If a user attempts to choose an element that is currently displayed with unavailable emphasis, indicate in the information area that the choice cannot be activated and that requesting help will explain why it is unavailable.

Visual Guidelines

Recommended

Display unavailable emphasis by dimming the choice or control that the user cannot interact with. For example, indicate a menu choice that is unavailable by changing every other pixel to the background color.

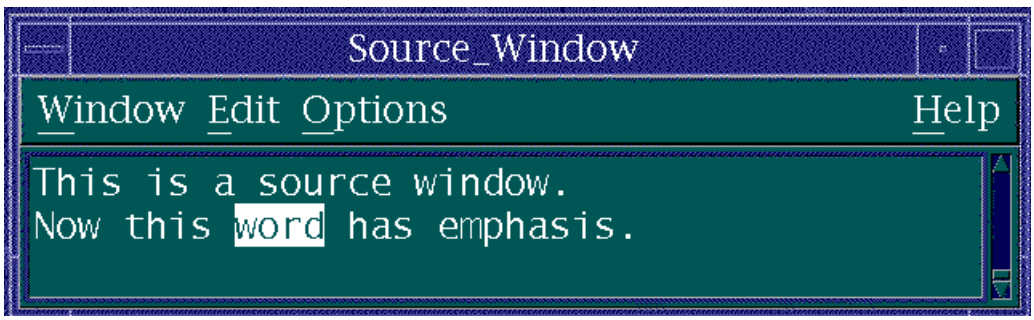
Supplemental Related Topics

For more information, see the Choice reference page.

Source Emphasis (Cue)

Source emphasis is a visual cue that indicates the element from which a user made a request or initiated a transfer operation.

Figure 4 Source Emphasis



When to Use

Recommended

Display source emphasis on the source elements during a drag–and–drop operation.

Recommended

Display source emphasis on an element while a pop–up menu is posted from that element.

Recommended

Display source emphasis on an element if an action message that applies to that element is displayed.

Optional

Display source emphasis on a selectable element if the pointer is on the element and the user is not in the middle of a selection operation.

Guidelines

Visual Guidelines

Recommended

Display source emphasis by reducing the contrast of the element being manipulated. For example, change an icon for an element by changing every other pixel to the background color, thus making the icon appear to dim.

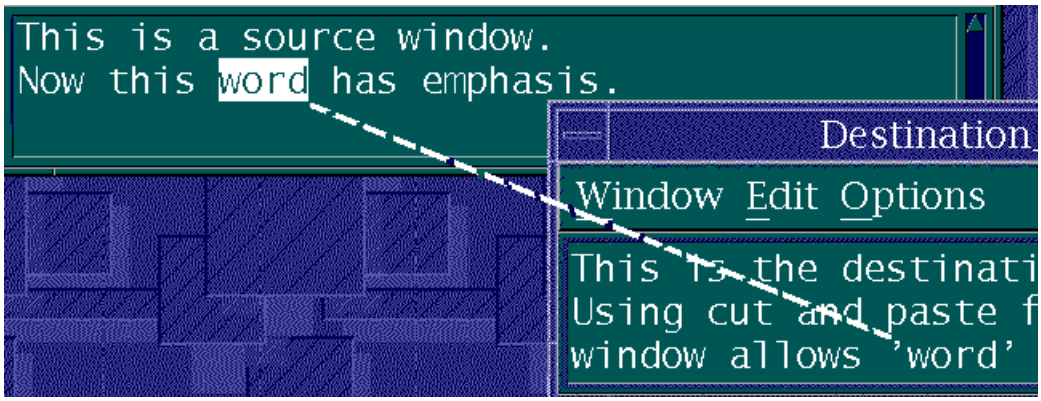
Essential Related Topics

For more information, see the Direct Manipulation reference page.

Target Emphasis (Cue)

Target emphasis is a graphical cue that indicates an element representing an object (or other application entity) that will receive the results of a transfer operation.

Figure 5 Target Emphasis



When to Use

Required

During a drag–and–drop operation to a target element in which a drop is likely to succeed, display target emphasis on the target element when the pointer is on that element.

Recommended

During a drag–and–drop operation, display source emphasis on the source element.

Guidelines

Recommended

When a target element is an icon that represents an object, do not display target emphasis on other icons for the object.

Recommended

Design target emphasis so that it will be visible in addition to any other forms of emphasis that could be currently visible on an element. For example, display both target emphasis and selected emphasis on a selected element that is the target of a drag−and−drop operation.

Visual Guidelines

Recommended

Display target emphasis as a solid line around the target element.

Recommended

When a target element is a window that contains a view of a container, display target emphasis as a solid thin line around the inside of the window adjacent to the border.

Optional

When elements in a selection scope are maintained in some order, use a graphic element as a marker placed between the items to indicate that the element (or elements) being dragged will be inserted between those items. The placement of the marker should indicate that the dragged element (or elements) will be added after the element that precedes the marker and prior to the element following the marker.

Essential Related Topics

For more information, see the Drag−and−Drop Transfer reference page.

Ready Emphasis (Cue)

Ready emphasis is a graphical cue for an element, used when the SELECT button is pressed, to indicate that the choice represented by that element will be activated or toggled. Ready emphasis is called "armed emphasis" in some environments.

Figure 6 Ready Emphasis

When to Use

Required

Display ready emphasis on a control if the action will be activated or toggled when the user completes the current action.

Guidelines

Required

When the pointer is located on an element that represents a choice and the user presses the SELECT button, display ready emphasis to indicate that releasing SELECT activates or toggles the choice. Modify the emphasis as follows if the user moves the pointer:

If the pointer is moved outside of the element while SELECT is pressed, change the element's appearance back to its normal condition.

If the pointer is moved back inside the element while SELECT is still pressed, redisplay the ready emphasis on the element to indicate pending activation or toggling.

Required

Remove the ready emphasis from the selected element when the user moves the pointer from the element, whether or not the action that would activate or toggle the choice was completed or when the user cancels the action.

Recommended

Ready emphasis for a value indicates the value of the choice if it were toggled.

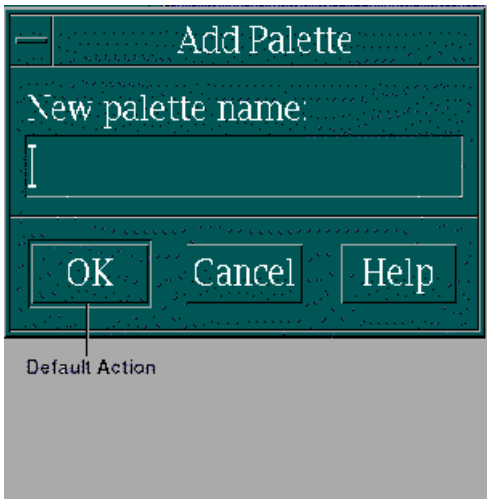
Essential Related Topics

For more information, see the Selection reference page.

Default Emphasis (Cue)

Default emphasis is the emphasis on a choice that indicates that it would be activated if the user requested the default action.

Figure 7 Default Emphasis



When to Use

Required

Display default emphasis on the push button whose action corresponds to the current default action.

Guidelines

Visual Guidelines

Optional

Provide default emphasis around a push button by drawing a border around it.

Essential Related Topics

For more information, see the Default Action reference page.

Interacted Emphasis (Cue)

Interacted emphasis is emphasis on a control that indicates that it is the last control containing an editable selection scope on the user's display with which the user interacted. In explicit mode, it is the last such control that had focus; in implicit mode, it is the last such control to which a key or mouse button press or release was directed.

When to Use

Required

In explicit mode, display interacted emphasis on the last control within a selection scope that had input focus.

Required

In implicit mode, display interacted emphasis on the last control within a selection scope that received input.

Guidelines

Recommended

In text, show interacted emphasis by displaying the cursor.

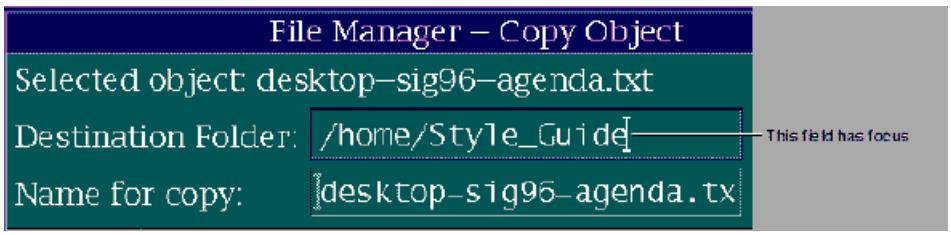
Essential Related Topics

For more information, see the Input Focus reference page.

Focus Emphasis (Cue)

Focus emphasis is emphasis on a control that indicates it is the control to which input will be directed. Always use focus emphasis when an explicit focus policy is in use. Focus emphasis is optional when an implicit focus policy is in use since the pointer location specifies the control that will receive input.

Figure 8 Focus Emphasis



When to Use

Required

When an explicit focus policy is in use, use focus emphasis to indicate which control has focus.

Optional

When an implicit focus policy is in use, you can use focus emphasis to indicate which control has focus.

Essential Related Topics

For more information, see the Input Focus reference page.

Exit (Choice)

NAME

Exit (Choice) –Reference

Description

Exit is an action choice that ends the current application and all windows associated with it. This action is equivalent to closing all primary windows of the application and ending the application.

When to Use

Required

Provide the Exit choice in the File menu. If you do not provide a File menu, make Exit the last choice in the pull–down menu for the first menu–bar item on the menu bar.

Required

Provide the Exit choice to allow the user to end the current application.

Guidelines

Required

When the user chooses the Exit choice, prompt the user to save any unsaved changes before ending the current application.

Recommended

Provide the Exit choice even though it is similar to the Close choice in the window menu to ensure that users can end the application even if they are running a noncompliant window manager.

Essential Related Topics

For more information, see the Close (Choice) and File Menu reference pages.

Supplemental Related Topics

For more information, see the Menu Guidelines reference page.

File Menu

NAME

File Menu –Reference

Description

The File menu includes the following choices:

File

A cascading choice that appears as a menu–bar item. It provides access to other menu items that allow a user to invoke actions for storage, data transfer, and display of the object, file, or data being viewed in the window.

Print

A dialog choice that prepares and arranges data to be printed.

Print Setup

An action choice that opens a window in which the user can set the parameters for printing.

Revert

An action choice that resets a view of an object or file to the state it was in when the object or file was last opened or last changed, whichever is more recent.

When to Use

Required

If you provide any of the functionality specified for the choices defined for a File menu, provide a File choice on the menu bar.

Guidelines

In conjunction with the Menu Guidelines reference page, use File Menu Choices to decide which of the choices listed to include in the File menu and how to organize them.

Table 1 File Menu Choices

Menu Choice	Mnemonic	Keyboard Function	
Required	N	New	`Ctrl N`
Required	O	Open ...	`Ctrl O`
Required	S	Save	`Ctrl S`
Required	A	Save As ...	
Optional		Revert	
Optional		Copy To ...	
Optional		Move To ...	
Optional		Drag	
Recommended	P	Print ...	`Ctrl P`
Recommended	R	Print Setup	
Recommended	C	Close	
Required	X	Exit	

Recommended

If an application does not have a state or contents that needs to be named and persistently stored, then do one of the following:

Replace the File choice's label with one identifying the application.

If the application manages a single type of element and uses a Selected menu whose label has been replaced with the name of that

type, eliminate the File menu and place Close (if needed) and Exit at the end of the Selected menu.

Optional

The File menu can include (at the end, preceded by a separator) a list of files previously viewed by the user that are no longer viewed in the current window (or optionally in any open window of the application). This list generally has a maximum length and can be optionally limited to the files viewed since starting the application or the session.

Guidelines for Specific Menu Entries

Required

When the user chooses Revert, modify the displayed state to revert to the state last saved or, if the state has not been saved, to the initial state.

Recommended

Do not support Revert if significant aspects of the data being viewed depend upon data provided continually (or intermittently) through an external source. If the user has indicated (through property settings) that the view should not automatically be changed as new data arrives, or if new data does not arrive automatically, provide a Refresh choice that explicitly obtains the new data, if necessary, and refreshes the view.

Recommended

If there is a selection, the Print dialog allows the user to specify that just the selected region is to be printed.

Optional

You can use Revert as a dialog or cascading choice if more than one kind of reversion is supported.

Essential Related Topics

For more information, see the Close (Choice), Exit (Choice), Menu (Control), Menu Guidelines, New (Action Choice), Open (Choice), and Save/Save As (Action Choice) reference pages.

Supplemental Related Topics

For more information, see the Help (Menu/Action Choice) and Menu Bar (Menu Type) reference pages.

File Selection (Dialog)

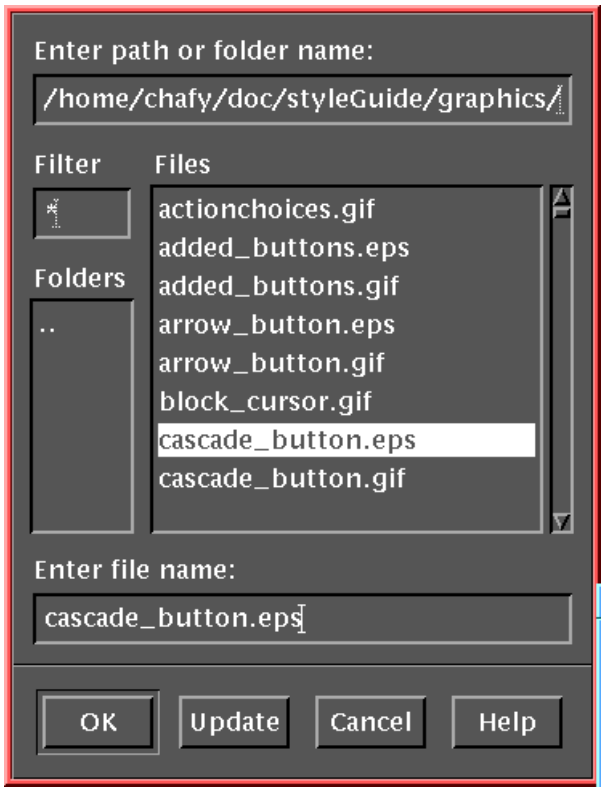
NAME

File Selection (Dialog) -Reference

Description

A file selection dialog is a dialog that allows the user to specify a file or element.

Figure 1 File Selection Dialog



When to Use

Use a file selection dialog when the user needs to open a file for processing.

Guidelines

Required

Allow the user to continue interacting with the application when a file selection dialog is displayed.

Required

The file selection dialog box must be composed of at least the following:

A text component for displaying and editing a directory mask (or filter) used to select the file to be displayed. The directory mask must be a string that specifies the base directory to be examined and a search path.

A list component for displaying file names. The list should display all files and subdirectories in the base directory that match the search pattern. The list must use either the single or browse selection model.

A text component for displaying and editing a file name.

A group of push buttons labeled OK, Filter, Cancel, and Help. If the file selection dialog is used to perform a specific action to the selected files, OK can be replaced by a label that describes the action to be done.

Required

The user must be able to select a new directory to examine by scrolling through the list of subdirectories and selecting the desired directory or by editing the directory mask. Selecting a new directory from the directory list must not change the search pattern. A user must be able to select a new search pattern by editing the directory mask.

Required

Allow the user to navigate the text within each list component by pressing `[uarr ]`, `[darr ]`, `Ctrl Begin`, and `CtrlEnd`. Allow the navigation actions `PageUp` and `PageDown` to move the cursored element within each list.

Required

The contents of the directory in the text−entry field must correspond to the contents of the directory list. The contents of the file name text must also correspond to the contents of the file name list.

Required

The user must be able to select a file by scrolling through the list of file names and selecting the desired file or by entering the file name directly into the text−entry field. Selecting a file from the list causes that file name to appear in the file selection text−display field.

Required

The user can select a new file as many times as desired. The application must not process the file until one of the following events occurs:

The user activates the OK button.

The user presses `Enter` or `Return` while the file name text−entry field has focus.

The user presses `Enter` or `Return` while the cursor is on an item in the file list.

The user double−clicks the SELECT button on an item in the file list.

Required

The file selection dialog must initiate a directory and file search when any of the following occurs:

The file selection dialog box is initialized

The user activates the Filter push button

The user double−clicks the SELECT button or presses `Enter` or `Return` on an item in the directory list

The user presses `Enter` or `Return` while the directory mask text−entry field has focus

Essential Related Topics

For more information, see the **Dialog (Choice Type)** , Text Display Field , and Text−Entry Field (Control) reference pages.

Supplemental Related Topics

For more information, see the Dialog (Window) reference page.

Find (Choice and Dialog)

NAME

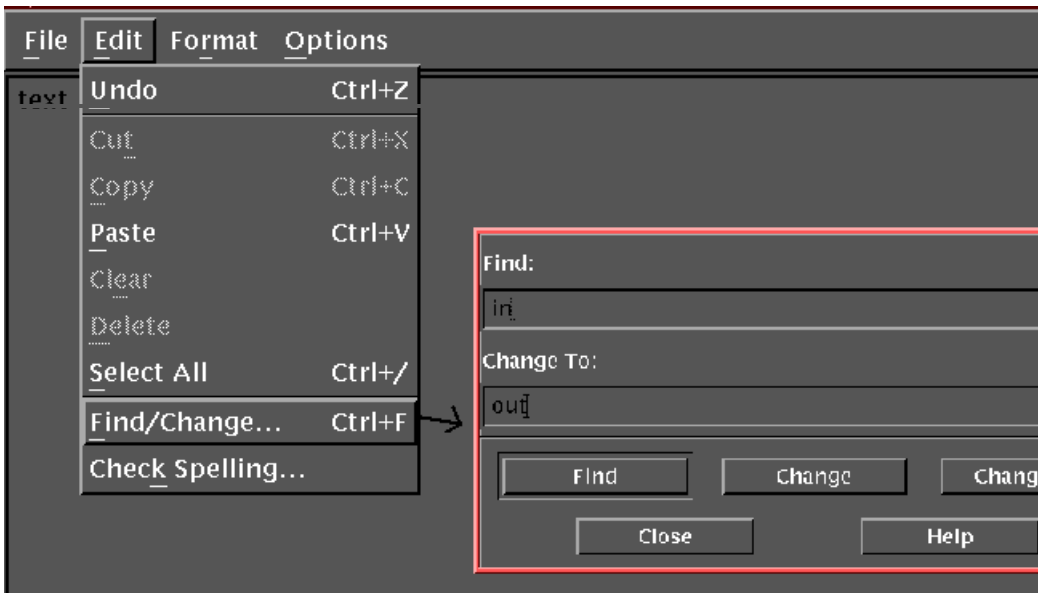
Find (Choice and Dialog) -Reference

Description

The Find choice can be one of the following:

- A dialog choice that initiates a search for objects or data associated with the contents of the current window
- A dialog that allows the user to specify the criteria to be used for the search
- An action choice within the Find dialog

Figure 1 Find Choice and Dialog



When to Use

Required

Provide a Find choice in the Edit menu to search for elements that match specified criteria. These elements may either be:

Within the view itself

Within objects that are represented in the view

In other objects, with the data in the view used, in part, to identify the other objects and/or the search criteria

Required

When Find is chosen, display a Find dialog window that allows a user to specify the parameters of the search. For example, when using the Find choice to search for text, allow a user to specify the text to be found, the direction of the search, sensitivity to text capitalization, and similar parameters.

Guidelines

Finding Matches

Required

If Find is used to find one match at a time, and more than one match meets the criteria, keep the Find dialog displayed.

Required

When the user specifies new criteria (and matches are to be found one at a time) and chooses the Find push button, find the first match and display it in the dialog.

Required

When the Find dialog remains displayed after the user chooses Find within it, find the next match when the user chooses Find again.

Recommended

If you want to allow every match found to be replaced, then use a Replace choice instead.

Recommended

If Find or Replace is used to find matches only within objects that are represented in the view, place the choice on the Selected menu.

Viewing Matches

Recommended

If the find operation is used to search for a single element, when the next match is found, move the active cursor to the matching element.

Recommended

If the find operation is used to search for a combination or sequence of elements (such as a word made up of text characters), select the next matching combination or sequence found.

Recommended

If Find is used to search for an element (or elements) outside of the current window, open a window to view the matching element within the object in which the next match is found. Then remove the window in which the previous match was found (unless it was the one from which the find operation was initiated).

Recommended

If Find is used to search for an element (or elements) within objects represented in the current window, move the cursor in the current window to the object in which the next match is found. Also, open a window to view the matching element within that object.

Recommended

Avoid changing the contents of a window as a result of a find operation, except to scroll the next match into sight and to indicate that it matches the search criteria.

Additional Behavior

Recommended

When Find is available in the Edit menu to search for elements within objects represented in the current window, the semantics for Find (for example, which objects it applies to) match those for choices in the Selected menu.

Recommended

When Find is used to search for elements within the current window, provide a choice within the Find dialog that indicates whether the search should be restricted to the currently selected elements.

Recommended

If Find is used to search for a combination or sequence of elements (such as a word made up of text characters) within a single selection scope, and discontiguous selections are supported in that scope, provide a choice within the Find dialog that determines when the next matching combination or sequence is found and selected, according to the following:

The current selection should first be deselected

The newly selected matching elements should be added to the current selection

Recommended

If the find operation is used to find matches within a single selection scope, include a choice within the Find dialog to determine whether to find matches one at a time (the default) or find all the matches at once. If the user chooses this option, Find selects all the matching elements (or combinations or sequences of elements).

Essential Related Topics

For more information, see the **Edit Menu** and View Menu reference pages.

Supplemental Related Topics

For more information, see the Object and Selection reference pages.

First–Letter Cursor Navigation

NAME

First–Letter Cursor Navigation –Reference

Description

First–letter cursor navigation is an internal navigation technique used in a selection scope in which entering a printing character moves the cursor in the scope to the next item whose textual label begins with that character.

Figure 1 First–Letter Cursor Navigation

When to Use

Required

Provide first–letter cursor navigation in a selection scope only if the selection scope does not support a keystroke–based mechanism such as mnemonic navigation.

Recommended

Provide first–letter cursor navigation in all selection scopes that associate textual labels with each selectable item.

Guidelines

Behavior

Required

Skip white spaces that begin a textual label to determine the first character of a textual label used to match an item.

Recommended

If the current language environment supports characters with multiple cases, such as uppercase and lowercase, allow the user to press letters of any case to match a specific textual label.

Key Pressing

Recommended

If the current language environment supports characters with multiple cases, such as uppercase and lowercase, allow the user to press letters of any case to match a specific textual label.

Required

When a user types a printing character in a scope by using first−letter cursor navigation, move the location cursor in the scope to the next item whose textual label starts with that letter.

Required

When a user enters a printing character that has no valid match, do not move the location cursor in the selection scope.

With Combination Text−List Controls

Required

When a printing character entered into a text−display field that has focus is used for first−letter cursor navigation of an associated list, select the item matched and place its text contents into the text−display field. If there is no valid match, generate a warning signal and do not make any changes to the appearance of the list or text−display field.

Required

When a printing character entered into a text−display field that has focus is used for first−letter cursor navigation of an associated list, and the list is not displayed, do one of the following:

If pressing `[darr ]` would display the list, make entering the character display it as well.

Navigate in the list as indicated and select the cursored list item, displaying it in the text field without displaying the list.

Recommended

When focus is in a text−display field that supports first−letter cursor navigation of an associated list, pressing `Ctrl [darr ]` and `Ctrl [uarr ]`, respectively, moves the cursor, selects the next or previous matching item that matches the prefix, and places its text contents into the text−display field.

Essential Related Topics

For more information, see the **Combination Box (Control)** , Cursor , and Drop−Down Combination Box (Control) reference pages.

Supplemental Related Topics

For more information, see the Container (Control) , Drop−Down List (Control) , and List Box (Control) reference pages.

Gauge (Control)

NAME

Gauge (Control) –Reference

Description

A gauge is a graphical display, generally used for values that the user cannot directly change. For example, you can use a gauge to display the percent complete in an in–progress message. The gauge has the same required and optional elements as a slider.

Figure 1 Gauge

When to Use

Recommended

Use a gauge in an in–progress message to show the degree to which a task is completed. For example, use a gauge to indicate how many files have been copied relative to the number that remain to be copied.

Recommended

Use a gauge when it is useful to show the user a value relative to a range of possible values but when the value cannot be changed.

Recommended

Use a gauge whose value can be shown when the displacement of the value from the minimum value is more important than the value itself.

Guidelines

Required

When the value of the gauge cannot be directly changed, use a zero–width slider arm. Fill the track between the position of the minimum value and the current value to achieve a thermometer–like effect.

Required

When the value of the gauge can be directly changed, do the following:

Use a slider arm whose appearance provides some indication (such as a thumb or handle) that it can be manipulated.
Fill the track between the position of the minimum value and the slider arm to achieve a thermometer–like effect.
Use the same pattern or color in the trough as in the slider arm so that they appear to be one continuous whole.
Place the slider arm so that the edge opposite to the filled area is at the position of the current value.

Required

Do not display arrow buttons in a gauge if the value in the gauge cannot be changed.

Required

All appropriate guidelines that apply to sliders apply to gauges whose values can be directly changed by the user.

Essential Related Topics

For more information, see the In–Progress Message , Scroll Bar (Control) , and Slider (Control) reference pages.

Group Box

NAME

Group Box –Reference

Description

A group box is a rectangular box drawn around a group of controls to indicate that the controls are related and to provide a heading for the group.

Figure 1 Group Box

When to Use

Recommended

Use a group box only when a group heading or white space alone does not visually distinguish groups of controls in a window.

Guidelines

Required

When you use a group box, provide a group heading as well.

Required

Do not allow a tab group to include elements from both inside and outside a group box.

Recommended

Avoid using a group box around a field of push buttons.

Recommended

Avoid using a group box around a single field. For example, avoid using a group box around a single list box.

Recommended

Place the group heading associated with a group box at the top edge of the group box.

Supplemental Related Topics

For more information, see the **Control** and Group Heading (Label Type) reference pages.

Group Heading (Label Type)

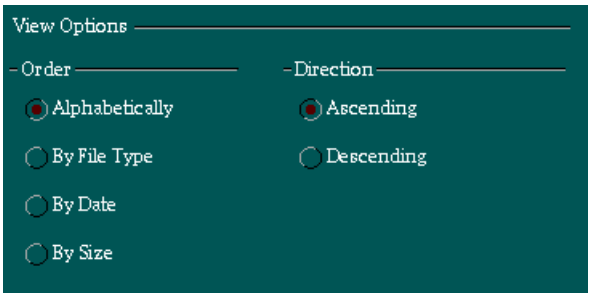
NAME

Group Heading (Label Type) -Reference

Description

A group heading is a textual label that identifies a group of related fields.

Figure 1 Group Heading



When to Use

Recommended

Provide a group heading for each tab group that consists of multiple controls.

Recommended

Use a group heading to visually distinguish and name a group of related fields.

Guidelines

Recommended

Place a group heading above each group of fields.

Recommended

Visually indicate which items are associated with the heading. If the operating environment supports different weights of type, use bolder type for the group heading, and align the fields under the heading. Otherwise, indent the fields in each group under their associated group heading or use a group box.

Recommended

Left–align group headings vertically with the left edge of other group headings. For information on bidirectional and vertical language support, see Chapter 11.

Recommended

If a group heading appears in a window that can be scrolled horizontally, scroll the heading with the group it identifies.

Recommended

If a group heading appears in a window that can be scrolled vertically, scroll the heading with the group it identifies. If this is not the

desired behavior, consider using a column heading instead.

Supplemental Related Topics

For more information, see Chapter 11 and the **Control** and Label reference pages.

Help (Menu/Action Choice)

NAME

Help (Menu/Action Choice) –Reference

Description

Help is information provided to the user about the application. It uses the following choices:

Help

A cascading choice that appears as a menu–bar item. It provides access to other menu items that provide various kinds of information related to the use of the application.

Index

An action choice that presents an alphabetic listing of help topics for an application.

Keyboard

An action choice that presents information about the application's use of keys, including function keys, shortcut keys, and mnemonics.

Mouse

An action choice that presents information about the application's use of the mouse.

On Item

An action choice that allows the user to indicate the element for which help is desired.

Overview

An action choice that provides a brief overview of each action and task that a user can perform within a window.

Reference

An action choice that provides access to reference pages for the application.

Table of Contents

An action choice that displays a table of contents of the help information available.

Tasks

An action choice that enables a user to determine how to perform specific tasks.

Tutorial

An action choice that gives a user access to online educational information.

Using Help

An action choice that gives a user information about how the help system works.

About

An action choice that displays a secondary window that contains information about an application, such as its copyright notice, logo, and version.

When to Use

Required

Provide a Help menu−bar item in the menu bar of all windows that have a menu bar and provide help information.

Recommended

Place a Help push button in a window that does not have a menu bar.

Guidelines

Required

In conjunction with the **Menu Guidelines** reference page, use Help Menu Choices to decide which choices to include in the Help menu and how to organize them.

Table 1 Help Menu Choices

Menu Choice	Mnemonic	Keyboard Function	
Recommended	O	Overview	
Optional	I	Index	
Recommended	C	Table of Contents	
Recommended	T	Tasks	
Recommended	R	Reference	
Optional	L	Tutorial	
Optional	K	Keyboard	
Optional	M	Mouse	
Recommended	O	On Item	`ShiftF1` or `ShiftHelp`
Required	U	Using Help	
Required	A	About *application−name*	

Required

Do not display the shortcut key for On Item (for example, `Shift F1` or `Shift Help`) in the menu entry if it would significantly increase the width of the Help menu.

Recommended

Provide help in the form most appropriate and useful to a user. For example, use text, graphics, animation, or any other technique or combination of techniques that conveys the information clearly and concisely.

Recommended

When possible, place a help window so that it does not cover the element for which help was requested.

Recommended

Display help information in a separate primary window.

Recommended

If the help system provides access to information that spans multiple applications, then use a separate help application to display the help information. If necessary, have a request for help start the application.

Optional

If help for both mouse and keyboard is provided, use one searchable entry for Mouse and Keyboard (M for example) instead of separate entries for each.

Indexes

Recommended

Do not include the words "Help for" in index entries. For example, do not label an entry Help for Cut.

Keyboard Help

Recommended

In keyboard help information, display all key assignments for the window or component from which the user requested help.

Recommended

In keyboard help information, indicate to a user which keys are available in the current state of the window.

Recommended

List shortcut key assignments in keyboard help information.

Recommended

If a user adds or changes key assignments, display the new or changed assignments in keyboard help information, along with any unchanged assignments.

On Element

Required

When the user chooses On Item, initiate context−sensitive help mode by changing the shape of the pointer to the question mark pointer. When the user moves the pointer to the element for which help is needed and clicks the SELECT button, exit from the mode and present any context−sensitive help available for the element. When the user presses `Cancel` or `Escape`, exit from the mode without presenting help.

Recommended

While in context−sensitive help mode, when the user clicks the MENU button while the pointer is on a cascading menu choice or at a position where a pop−up menu is defined, display the menu, allowing the user to obtain context−sensitive help for the menu entries. Remove the menus when the mode is exited.

About

Recommended

About should display a window that indicates minimally the name and version of your application. Display the icon or some other signature graphic for your application.

Supplemental Related Topics

For more information, see the Menu Guidelines reference page.

Icon

NAME

Icon –Reference

Description

An icon is a pictorial representation of an object that can be manipulated directly. An icon typically consists of an image and a label. The image might be text, a frame from a video object, or a graphic representation of a printer or document, and may include animation. Icons for the same object may be displayed in various sizes. The operating environment and the container for a collection of icons may support distinct small, medium, and large icons that represent the same object but differ in detail and size.

Figure 1 Window and Application Icons

Window Icon:

Application Icons:

When to Use

Required

Use an icon to represent objects that the user can place in a container.

Recommended

Do not use an icon to represent a function. Place function choices on menus and in push buttons or tools.

Recommended

Use large icons only for window icons.

Guidelines

Required

Define an icon to consist of an image and an optional textual label as follows:

Image

Use a common, task–related symbol associated with the object it represents.

Label

If you provide a label, place the label below the image. If the image is small relative to the size of the label, place the label to the right of the image. For information on bidirectional and vertical language support, see Chapter 11.

Required

Ensure that the image has a border that prevents the image and the background from merging.

Designing Icons

Recommended

Use a readily identifiable shape or outline for the icon. This helps to improve user recognition and reduce a user's dependence on other identifiers such as the label or color. For example, an icon of a printer clearly identifies its purpose; the label can be reduced to a brief identifier for the specific printer.

Recommended

Design an icon so that it shows the important states associated with the object it represents. Such states might include:

The object needs attention, for example, the printer is out of paper.
Some threshold has been reached, for example, a mailbox is full.
The object can only be read, for example, the user does not have the authority to modify a document.

Recommended

Design an icon so that it shows the important characteristics associated with the object it represents. Such characteristics might include:

The security classification of an object, for example, a document is confidential or of restricted distribution.
The urgency of some task associated with the object, for example, an electronic mail item that must be responded to immediately.
The size of the object, for example, the number of mail messages in a mail container.

Recommended

If you show shadows on objects to add depth to the image, assume the light source is from the front top–left and place the shadows to the right, below and behind the object.

Recommended

When several elements or objects are used together within one icon image, visually unite them. For example, overlap the elements to present a more visually unified whole.

Recommended

If your application will be presented in more than one language, avoid using words, body parts, or figures that involve signing with hands or fingers in the icons.

Designing Small Icons

Required

Limit the detail in a small icon to avoid a cluttered appearance. For example, do not display a count of the number of objects in the container.

Recommended

Do not use an algorithmically reduced copy of the original icon as a small icon. Instead, use a separate graphic to display a small icon that shows fewer characteristics so that the characteristics can be more easily distinguished.

Recommended

Use a small icon to represent the general type of an object when space is limited.

Changing the Icon of an Object

Recommended

Allow a user to customize the image of an icon that represents an object. For example, allow a user to change the icon to make it more meaningful, recognizable, or personal.

Using Color in Icons

Recommended

Use colors in an icon to enhance user recognition and to intentionally link or group related elements. To avoid a busy, cluttered screen and to reduce the chance of unintentionally indicating a relationship between objects, do not overuse different colors.

Recommended

If you use colors in icons, also provide a separate set of icons that can be used on gray scale or monochrome displays.

Icon Labels

Required

If an icon represents a named object, label the icon with the name of the object.

Required

If direct editing can be used to change the textual label of an icon, include a Properties choice in the icon's pop−up menu, and include a text−entry field in the properties window through which the label can be changed.

Recommended

Support direct editing to allow mouse users to change the textual label of an icon.

Recommended

Visually separate the textual label from the icon image.

Recommended

When the icon has focus, expand the width of the label to display the entire text. For example, when the icon for the "1999 Financial Goals" document has focus, display the entire document name as the icon label.

Supplemental Related Topics

For more information, see the **Label** , Object , and Text−Entry Field (Control) reference pages.

In–Progress Message

NAME

In–Progress Message –Reference

Description

An in–progress message is a message that informs the user about the status of a task, such as copying a set of files. If you normally use a separate window for an in–progress message, you can also display an information message in that window.

Figure 1 In–Progress Message

When to Use

Recommended

Display an in–progress message when the user would benefit from knowing that a task is taking place and the status of that task.

Recommended

Display an in–progress message if a task is proceeding normally and your application can monitor the completion status of the task without interrupting it and the status can be displayed visually.

Recommended

When a wait pointer has been displayed for more than an interval specified in the operating environment (typically 15 seconds), display an in–progress message in addition to the wait pointer.

Guidelines

Required

An in–progress message in a secondary window should contain at least a graphic image, a textual label or message, and the OK and Close choices.

Required

Do not interfere with the user's interaction with an application just because an in–progress message is being displayed; for example, normally make the in–progress message window a modeless window.

Required

Close an in–progress message window when a request made by a user completes normally.

Required

If information that would have been displayed in an information message is displayed in an in–progress message, do not close the window until the user explicitly requests to close it, such as by activating the Close push button.

Recommended

Show a gauge and continually update it to accurately represent the known progress of a request made by a user.

Recommended

Display the label for the gauge in an in–progress message above or to the left of the gauge. For information on bidirectional and vertical

language support, see Chapter 11.

Providing Controls in In−Progress Messages

Required

Provide a Close push button to allow a user to close the in−progress message window without affecting the task.

Recommended

Provide a Stop (or Cancel) push button to allow a user to end the task and close the window.

Recommended

Provide a Pause push button that allows the user to suspend a task and a Resume push button that allows a user to resume the task that has been suspended. Display unavailable emphasis on one or the other, depending on whether the task is running or has been suspended.

Recommended

Provide a Help push button to provide access to help information.

Visuals

Required

Display the hourglass symbol at the left of an in−progress message. For information on bidirectional and vertical language support, see Chapter 11.

Essential Related Topics

For more information, see the **Action Message** , Gauge (Control) , Information Message , Message , Pointer (Predefined) , Push Button (Predefined) , and Warning Signal reference pages.

Include (Choice and Dialog)

NAME

Include (Choice and Dialog) –Reference

Description

Include is a dialog choice used either to determine which elements are displayed within a window or to identify the subset of elements to be displayed, added, or removed from a view.

When to Use

Recommended

Provide an Include choice in the View menu when it is useful to view subsets of elements contained within the view.

Required

When Include is chosen, display an Include dialog window that allows the user to specify a subset of elements to be displayed within the view.

Guidelines

Required

When an element is added to the data on which a view filtered by an include action is displayed, and the element matches the criteria for the view, display the element in the view.

Recommended

When the user specifies, in the Include dialog, the properties of a subset of the elements that can be viewed, allow the user to choose whether to:

Replace the elements in the current view (for example, show the specified subset of all the elements instead)
Restrict the current view (for example, show the specified subset of the currently viewed elements)
Add elements to the view (for example, add the specified subset of all the elements to the view)
Remove elements from the view (for example, remove the specified subset of the currently viewed elements from the view)

Recommended

In the window in which the results of the Include operation are displayed, provide a status area to list the criteria used to create the subset, as well as the number of elements displayed and the total number of elements from which they were chosen.

Recommended

Provide a choice within the Include dialog to allow the result of the Include operation to be shown in a separate window instead of in the current view.

Essential Related Topics

For more information, see the **View** and View Menu reference pages.

Supplemental Related Topics

For more information, see the Object and Selection reference pages.

Information and Message Areas (Area)

NAME

Information and Message Areas (Area) –Reference

Description

The information area is a specific part of a window in which information about the current application task context and other ephemeral task –related messages are displayed.

The message area is a specific part of a window in which the user can get detailed information or directions about performing a task in the current context.

Your application can have one area that performs both these functions only if there is no conflict.

Figure 1 Information and Message Area

When to Use

Recommended

Provide an information or message area for each primary window and each nonmodal secondary window.

Recommended

Provide a message area for a window that contains rarely used choices.

Recommended

Provide a message area if the task to be performed in the viewing area is complex, requiring many steps to be performed precisely.

Recommended

Do not provide an information or message area for a message window.

Guidelines

Recommended

Do not scroll the information or message area when the window's viewing areas are scrolled.

Recommended

Do not provide scroll bars for the information area and do not allow a user to scroll information in the information area.

Recommended

Provide scroll bars for the message area if there is not enough space to display the complete message.

Recommended

Do not allow a user to interact with the information or message areas, except to transfer data from the area.

Recommended

If you display any text in an information area, make it a textual label.

Recommended

If you display any text in a message area, put it in a text–display field.

Recommended

Remove information from the information area as soon as it is no longer relevant to the current state of the window or the current position of the cursor.

Layout

Recommended

If you include an information or message area, design it to run from border to border across the width of the window.

Recommended

Make the information area no taller than necessary to accommodate brief, but meaningful, information. For example, if the information area displays text, accommodate at most two lines of text.

Recommended

If the message area displays text, make it two to five lines high.

Recommended

Place the information and message areas at the bottom of the client area of a window. If you provide both, place the message area above the information area.

Recommended

Separate the areas from the rest of the window. For example, use a solid line or a horizontal scroll bar to separate the information area from the rest of the window.

Recommended

If the user reduces the width of a window with an information or message area and the information can wrap in the area, wrap the information; otherwise, clip the area.

Displaying Ephemeral Messages in the Information Area

Recommended

Do not use the information area to display information that a user must see; display that type of information in a message area.

Recommended

Use the message area for information that the user must see but does not need to acknowledge. If the user must acknowledge the message, use a message dialog; if the user does not need to acknowledge the information and it is brief, use an information area.

Recommended

Place information about the normal completion of a process in the information area. For example, indicate in the information area that a file was successfully saved.

Recommended

For views that can support more than one selection model, if it is unclear from the context which ones are supported, place in the information area information about the selection models supported. For example, if only single selection is supported, display a label similar to the following:

```
Select only one
```

Optional

When the active cursor is on a choice, briefly describe in the information area what will happen when a user activates or toggles that choice.

Optional

When the active cursor is on an element other than a choice, briefly suggest in the information area how to perform available actions on the element.

Optional

If a default action is available but no push button with default emphasis is visible, briefly indicate the default action in the information area.

Optional

When the user presses the SELECT button on a push button in an application, use the information area to describe the effect of activating the button. Remove the information when the user releases SELECT.

Displaying Messages in the Message Area

Optional

As the user follows the instructions in the message area, if your application can determine that the message has been followed, scroll the message area so that the next instruction is visible.

Essential Related Topics

For more information, see the **Information Message** reference page.

Supplemental Related Topics

For more information, see the Choice , Clipboard , In–Progress Message , and Status Area (Area) reference pages.

Information Message

NAME

Information Message –Reference

Description

An information message is a message that indicates to a user that a condition or an event has occurred. For example:

```
Document has been deleted
```

Figure 1 Information Message Dialog

When to Use

Required

Display an information message when a situation has occurred that the user may need to know about, but which has no follow–up actions the user can request.

Required

Display an information message when there is additional information about the status of normal completion of a task or about the current status of the task.

Required

If an in–progress message for the task is currently displayed in its own secondary window, do not use an information message. Instead, augment the in–progress message window with the message information.

Guidelines

Required

Display the information symbol appropriate to your operating environment to the left of the information message. For information on bidirectional and vertical language support, see Chapter 11.

Required

Design the information message window to be modeless.

Providing Controls in Information Messages

Required

Provide an OK push button that allows a user to acknowledge the information displayed in the information message and to remove the message.

Recommended

Provide a Help push button that allows the user to request help information.

Visuals

Required

Display the (i) symbol as the graphic image at the left of an information message. For information on bidirectional and vertical language support, see Chapter 11.

Essential Related Topics

For more information, see the **Action Message** , In–Progress Message , Message , and Push Button (Predefined) reference pages.

Supplemental Related Topics

For more information, see the Information and Message Areas (Area) reference page.

Input Focus

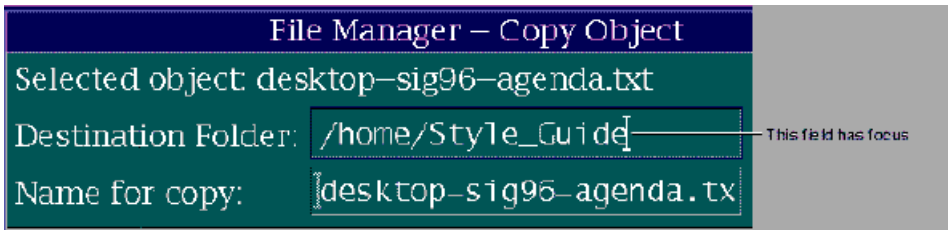
NAME

Input Focus -Reference

Description

Input focus, or focus, identifies the place to which keyboard input is directed.

Figure 1 Focus on a Text–Entry Field



When to Use

Required

When the user has set the input focus in the workspace to track the pointer, use an implicit focus policy to move focus from window to window by always ensuring that the window under the pointer has the focus.

Required

When the user has set the input focus in the workspace to be explicitly changed, use an explicit focus policy to move focus from window to window.

Required

When the user has set the input focus within a window to track the pointer, use an implicit focus policy to move focus from control to control within that window by always ensuring that the control under the pointer has the focus.

Required

When the user has set the input focus within a window to be explicitly changed, use an explicit focus policy to move focus from control to control, except within controls that are spring sensitive.

Guidelines

Required

When an explicit focus policy is in use in a workspace, and the user presses the SELECT button (optionally augmented by `Shift` and/or `Ctrl`) in a window, move the focus to that window.

Required

When an explicit focus policy is in use in a window, and the user presses the SELECT button (possibly augmented by `Shift` and/or `Ctrl`) in a control that can take focus, in addition to performing any action the control binds to the button press, move the focus to the control, unless the control is used to change the visible portion of some area (for example, a scroll bar).

Required

When a window that uses an explicit keyboard focus policy acquires focus through either keyboard navigation or a button press (but not by pressing SELECT [possibly augmented by `Shift` and/or `Ctrl`] in a control in that window) causing that control to take focus, then move the focus to the control that last had focus in the window if the following are true:

- The window had the focus at some time in the past (whether or not it has been minimized since then)
- The control that last had focus in the window is still able to receive input focus

Otherwise, move focus to either of the following:

- The control with which the user is most likely to want to interact
- The initial control of the first tab group in the window

Required

Do not change the state, appearance, or behavior of a control when the user uses the keyboard to move the input focus to the control, except for showing focus emphasis. For example, when the user moves through a set of radio buttons with the directional keys, do not change the state of any of the radio buttons.

Required

When the user presses `Ctrl` SELECT on a control that can take focus, move focus to the control but do not perform any other action unless pressing `Ctrl` SELECT already has a predefined meaning in that control.

Recommended

When the user uses a pointing device to manipulate a control such as a scroll bar or a split bar that changes the visible portion of some area, do not move the focus to that control.

Recommended

Do not change the input focus as a result of a direct manipulation operation except for those that use the SELECT button, possibly augmented by `Shift` and/or `Ctrl`.

Essential Related Topics

For more information, see the **Active Window** reference page.

Supplemental Related Topics

For more information, see the Cursor , Keyboard (Device) , Mouse (Device) , and Window Navigation reference pages.

Input Models (CDE)

NAME

Input Models (CDE) –Reference

Description

Input models specify the way data is entered or manipulated in a CDE application.

Input devices have different actions depending on which part of the interface the user is interacting with. Usually, mouse users can access windows and controls more easily than keyboard users, due to the inherent flexibility in mouse manipulation. Keyboard users must use specific keys to move the cursor in the application.

Guidelines

Keyboard Focus Model

Required

Only one window at a time can have keyboard focus. Highlight the window that has focus. Within the window that has keyboard focus, only one component at a time can have focus.

The keyboard focus determines which component on the screen receives keyboard events. This rule prevents confusion about which window and component have focus.

Required

If your application uses an explicit focus policy, when the user presses the SELECT button, do not move focus to a component that is not traversable or does not accept input.

An explicit focus policy requires the user to explicitly select which window or component receives keyboard focus. Generally, the user gives focus to a window or component by pressing the SELECT button over it. However, this policy must not allow the user to give focus to a component that is not traversable or does not accept input.

Required

If your application uses an explicit focus policy, highlight (with a location cursor) the component with the keyboard focus.

The user needs to know the location of the keyboard focus to be able to control an application.

Input Device Model

Required

Support methods of interaction for keyboard–only users. All features of your application should be available from the keyboard.

Some users may not have access to a pointing device. These users need to be able to access the full functionality of the application from the keyboard. Also, advanced users may use the keyboard to perform some tasks more quickly than with a pointing device.

Required

Use the following bindings for mouse buttons:

SELECT

Used for selection, activation, and setting the location cursor, SELECT is the leftmost button, except for left–handed users, where it can be the rightmost button.

TRANSFER

Used for moving and copying elements, TRANSFER is the middle mouse button, unless dragging is integrated with selection or the mouse has fewer than three buttons.

MENU

Used for popping up menus, MENU is the rightmost button, except for left−handed users, where it can be the leftmost button, or unless the mouse has fewer than three buttons. If the mouse has one button, bind the MENU button to `Alt` SELECT.

Required

Do not warp the pointer unless you have given the user a means of disabling the behavior.

The pointer position is intended only as input to applications, not as an output mechanism. An application warps the pointer when it changes the pointer's position. This practice is confusing to users and reduces their sense of control over an application. Warping the pointer can also cause problems for users of absolute location pointing devices (for example, a puck on a graphics tablet).

Required

Components or applications developed for CDE must adhere to Motif key bindings (see Appendix A).

Essential Related Topics

For more information, see the **Keyboard (Device)** and Mouse (Device) reference pages.

Supplemental Related Topics

For more information, see the Pointer (Predefined) reference page.

Internal Navigation

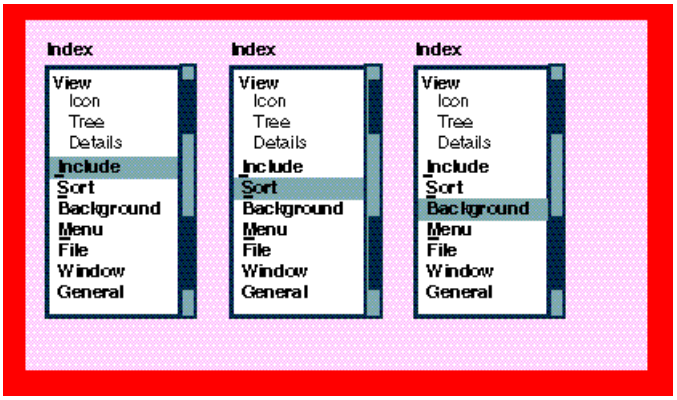
NAME

Internal Navigation –Reference

Description

Internal navigation is the use of the directional keys (and related keys) for internal purposes within a control –– primarily to move the active cursor within the control, but also for other purposes such as changing a value displayed or represented by a control.

Figure 1 Internal Navigation Within a List Box



When to Use

Recommended

Use internal navigation when the active cursor needs to be moved within the control.

Recommended

Use internal navigation when manipulating elements within a control.

Guidelines

Required

If a control uses internal navigation to move an element cursor among elements within a control, follow all the rules for control navigation, proceeding as if the elements in the control were controls within a tab group.

Graphics Cursor

Required

Allow the directional keys to move a graphics cursor one unit in the indicated direction, where a unit is generally a pixel.

Required

Allow the directional keys augmented by `Ctrl` to move a graphics cursor multiple units in the indicated direction.

Required

Allow `Home` (or `Begin`) and `End` to move a graphics cursor to the leftmost and rightmost ends, respectively, of the data being viewed in the control. For information on bidirectional and vertical language support, see Chapter 11.

Required

Allow `Ctrl Home` (or `Ctrl Begin`) and `Ctrl End` to do one of the following:

Move the graphics cursor to the topmost and bottommost ends, respectively, of the data being viewed (recommended).

Move the graphics cursor to the top leftmost and bottom rightmost ends, respectively, of the data being viewed. For information on bidirectional and vertical language support, see Chapter 11.

Required

Do not allow a graphics cursor to wrap from one edge to the opposite edge.

Positional Indicator

Required

When the purpose of a control is to allow a positional indicator to be set to a value (for example, an arm in a slider), then when the control has focus, allow directional keys to move the positional indicator one unit (not necessarily one pixel) in the corresponding direction.

Required

When the purpose of a control is to allow a positional indicator to be set to a value, allow directional keys augmented by `Ctrl` to move the positional indicator multiple units in the indicated direction.

Recommended

If a control with a positional indicator has major and minor tick marks, and if minor tick marks indicate each single unit, then allow directional keys augmented by `Ctrl` to move to major tick marks.

Home and End Keys with Positional Indicators

Required

If a control with a positional indicator has only one degree of freedom, then do one of the following:

Make both vertical and horizontal navigation have the same effect.

Support either vertical *or* horizontal navigation.

Required

If a control with a positional indicator has only one degree of freedom, then allow `Home` (or `Begin`) to move the positional indicator to its initial position, and `End` to move it to its final position.

Required

If vertical and horizontal navigation have different effects on a positional indicator, then allow `Home` (or `Begin`) to move the indicator as far as it can go toward the initial or final horizontal positions. Allow `Ctrl Home` (or `Ctrl Begin`) and `Ctrl End` to move the indicator as far as it can go towards the initial or final vertical position.

Changing Values

Required

If you use horizontal or vertical navigation to switch to a value from among a linear sequence of values, then make `Home` (or `Begin`) switch to the first such value, and `End` switch to the last such value.

Recommended

If the user can use a control to set a value from among a linear sequence of values, then when the control has focus, allow horizontal navigation, vertical navigation, or both, to switch from one value to another.

General

Recommended

If additional internal navigation techniques are required, extend the navigation model as needed, but do not deviate from it unnecessarily.
Some situations that may require additional navigation techniques are:

- Controls in which the elements are densely packed or are organized into layers
- Controls that use three–dimensional navigation
- Controls that organize elements hierarchically

Essential Related Topics

For more information, see the **Control Navigation** , Cursor , Mnemonic , and Tab Group reference pages.

Supplemental Related Topics

For more information, see the Control , Keyboard (Device) , Push Button (Predefined) , Scroll Bar (Control) , and Window Navigation reference pages.

Keyboard (Device)

NAME

Keyboard (Device) –Reference

Description

The keyboard is a device, consisting of systematically arranged keys, that allows a user to type information, move the cursor, or activate functions assigned to keys.

This book uses a model keyboard. Refer to Appendix B for details.

When to Use

Required

Use equivalent, though not necessarily identical, mouse and keyboard techniques to provide access to all functions.

Guidelines

Required

Use `Ctrl`, `Shift`, and `Alt` only to modify the function of other keys or key combinations on the keyboard or pointing–device actions.

Required

Use `Alt` in conjunction with printing characters only to provide access to mnemonics in an application or for predefined actions. For information on bidirectional and vertical language support, see Chapter 11.

Required

If the functions listed in Appendix B are provided, assign them to the keys listed in the table. Do not assign these functions to other keys. Only the user should reassign them.

Recommended

Use `ScrollLock` only to augment navigation keys for navigating with the scroll bar.

Essential Related Topics

For more information, see the Control Navigation , Internal Navigation , Tab–Group , and Window Navigation reference pages.

Supplemental Related Topics

For more information, see the Clear (Action Choice) , Control , Copy (Choice) , Cursor , Delete (Action Choice) , Menu (Control) , Mnemonic , Selection , Shortcut Key , Text–Entry Field (Control) , and Window Navigation reference pages.

Label

NAME

Label –Reference

Description

A label includes the following:

Label

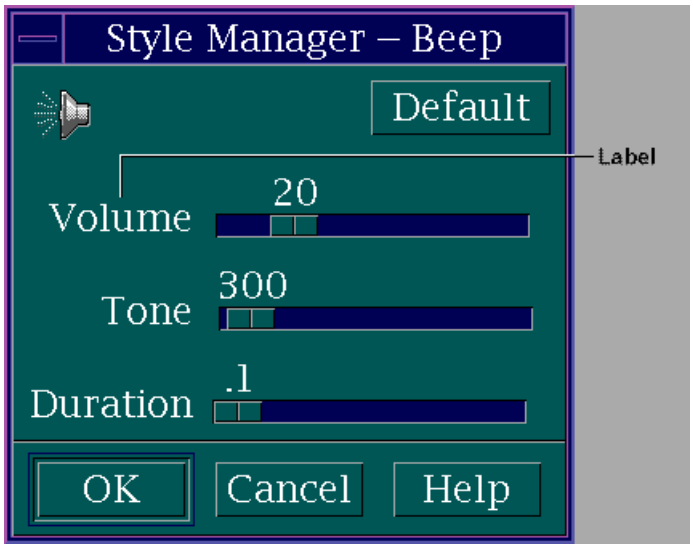
Is a noneditable element that contains text or a graphic that identifies a control or provides additional information about the control. Labels can also be names that identify objects.

Static text (control)

Is a textual label that can take focus and to which a user can navigate by using control navigation.

Label is an example of a label.

Figure 1 Label



When to Use

Recommended

Use a label to provide additional information or to explain the function of a control or group of controls. For example, use a label to describe the required date format for a text field that will contain a date.

Recommended

Provide a label for each tab group unless the window contains only one tab group and the window title serves as the label.

Recommended

Provide a label for each text field unless the fields appear in rows or columns and associated row or column headings serve as the labels.

Guidelines

If a label contains a significant amount of text and the user is likely to want to use it as a data transfer source, provide a means for both mouse and keyboard users to do so.

Required

Do not treat a static text control as a tab group.

Required

Do not support internal navigation or selection within a static text control.

Layout

Required

If an element contains a choice described by a textual label, place the label within the element (for example, a push button or radio button) unless some other textual information needs to be placed there (for example, an option menu button).

Recommended

If you provide a label to describe a control and it cannot be placed within the control, place it to the left or above the control. For information on bidirectional and vertical language support, see Chapter 11.

Recommended

If you provide a label as a name of an object, place it below the object. If the object is a small icon, place it to the right of the object. For information on bidirectional and vertical language support, see Chapter 11.

Recommended

If you provide a label to indicate the type of information or descriptive information about an element, place it to the right of the element. For information on bidirectional and vertical language support, see Chapter 11.

Recommended

When you provide a label for a group of controls in a column, place the label so that it cannot be confused with a column heading.

Recommended

If a label is displayed above a text field, align the label with the left edge of the field. For information on bidirectional and vertical language support, see Chapter 11.

Recommended

For controls that support more than one selection model, and if it is unclear from the context that this is the case, use a label to provide information about the selection model currently supported. For example, provide a label for a list box that uses multiple selection that indicates that an item can be selected or deselected by clicking on it.

Optional

In high−volume data−entry windows, consider using right−aligned edges of labels next to left−aligned text fields. This leaves a narrow vertical column of space between the labels and their associated text fields to allow a user to quickly scan the choices in the window. For information on bidirectional and vertical language support, see Chapter 11.

Quick Transfer

Required

If a label supports direct editing, support quick transfer to transfer the range of text identified by the quick transfer operation.

Optional

If a label does not support direct editing, allow a quick transfer beginning and ending in the label or field to use the entire contents of the label or field as its source.

Visuals

Recommended

Make labels visually less important than the data to which they refer, through color, size, typeface, or a combination.

Recommended

Capitalize the first letter of the first and last words of a label and all other words except articles, coordinating conjunctions, prepositions, and the "to" in infinitives.

Supplemental Related Topics

For more information, see the Control , Group Heading (Label Type) , and Selection Models reference pages.

List Box (Control)

NAME

List Box (Control) -Reference

Description

A list box is a control that contains a list of items that a user can select.

Figure 1 List Box

When to Use

Recommended

Use a list box to display a list of items in which the number of items may vary. Items may either be choices, or they may identify objects by their names.

Guidelines

Recommended

Display the items in a list box in an order that is meaningful to a user, such as alphabetic order, numeric order, chronological order, or some other order. For example, display modem baud rates in numeric order.

Behavior

Required

When the cursor is on a list item that labels a viewable object, make presenting that object the default action.

Required

Design list boxes with either a single, browse, multiple, or extended selection model.

Recommended

Provide first–letter cursor navigation in a list box.

Navigation

Required

In a scrollable list box, do not allow the cursor to wrap.

Required

Display the active cursor as an element cursor on the appropriate list item.

Required

When the user presses [uarr] or [darr], navigate to the previous and next items in the list box, respectively.

Required

If a list box can be scrolled horizontally, make it a tab group.

Required

If a list box can be scrolled horizontally, then when focus is in the list box, allow [larr] and [rarr] to scroll left and right by one character width, and Ctrl [larr] and Ctrl [rarr] to scroll left and right by a larger amount.

Recommended

Design a list box to be a tab group.

Visuals

Required

Provide the list box with an area for presenting a list of labels as items.

Required

Provide vertical scroll bars when some of the list items are not visible in the list box.

Required

Arrange the list items vertically.

Required

Use different cursor visuals when the list box is in add mode from the one used in normal mode.

Recommended

Design a list box to be large enough to display a minimum of six list items at a time to provide context to the user.

Recommended

Make list boxes at least wide enough to display list items of average width.

Recommended

When a user increases the size of the window in which the list box is displayed, increase the size of the list box so that more items can be seen.

Recommended

When a user decreases the size of the window in which the list box is displayed, decrease the size of the list box so that fewer items can be seen, unless reducing the size of the list box will prevent the list box from displaying two list items.

Recommended

Provide horizontal scroll bars when items are wider than the list box.

Recommended

In normal mode, use a solid box around the item as the cursor visual.

Recommended

In add mode, use a dashed box around the item as the cursor visual.

Optional

If a window can be decreased in size so much that two items in the list cannot be seen, consider allowing the window to be scrolled or

using a drop–down list box instead of a list box.

Essential Related Topics

For more information, see the **Control** and Drop–Down List (Control) reference pages.

Supplemental Related Topics

For more information, see the Combination Box (Control), Drop–Down Combination Box (Control), First–Letter Cursor Navigation, Label, Scroll Bar (Control), and Selection reference pages.

Margin Selection Techniques

NAME

Margin Selection Techniques –Reference

Description

Margin selection techniques are variants of other group techniques in which the user selects or deselects a group of elements by acting on elements in the margin of the control managing the selection scope. For example, selecting elements in the margin of a spreadsheet might select an area consisting of all elements in the corresponding rows or columns.

You can use any of the standard techniques (including adjustment) as a margin technique. For example:

- A margin area point technique could involve clicking on a single margin element to select or toggle elements in a single row or column of a spreadsheet.
- A margin area swipe technique (for example, pressing the SELECT button and moving it through a range of margin elements) could be used to select a range of rows or columns.

You can also apply margin techniques to margin elements. For example, if margin elements are used with text, then clicking the SELECT button in the margin (a margin range point technique) might select a line, but double–clicking SELECT in the margin (a multilevel margin range point technique) might select an entire paragraph.

When to Use

Required

If margin elements can take focus, place them in a separate selection scope and a separate tab group from the selection scope they control.

Guidelines

Required

After using a margin point technique, do the following:

Define the selection region to consist of the elements corresponding to the margin element that was used.
Place the anchor point at the point in the region nearest to the margin element.
Identify the anchor element to be the element, if any, within the selection region nearest to the anchor point.
Define the anchor region to be the selection region.
If the margin element cannot take focus, place the active cursor at one of the following:

The anchor point, if it can be placed there
If an element cursor, on the anchor element
If there is no anchor element, where the cursor previously was in the scope

Required

When using the margin click or swipe techniques:

Define the selection region to consist of all the elements that correspond to the region of margin elements identified by the technique.
Place the anchor point at the point in the region nearest to the position in the margin at which the technique was initiated.
Define the anchor element to be the element in the selection region nearest to the anchor point.
Define the anchor region to consist of the elements corresponding to the anchor element of the margin's selection region.
If the margin elements cannot take focus, place the active cursor at one of the following:

The pointer position or element farthest from the margin's anchor element, if it can be placed there
Where the cursor previously was in the scope

Required

When using the margin point, swipe, or click techniques in select mode:

Select all the elements in the selection region.
Deselect all other elements in the scope.

Required

When using the margin point, swipe, or click techniques in toggle mode, toggle all elements in the selection region, based on the toggling policy.

Essential Related Topics

For more information, see the **Multilevel Selection Techniques** , Selection Modes , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Selection Models and Selection Policies reference pages.

Maximize (Choice)

NAME

Maximize (Choice) -Reference

Description

Maximize is an action choice that enlarges a window to its largest possible size. If appropriate, the Maximize choice may also be represented by a maximize button in the title bar of a window.

When to Use

Required

Provide a Maximize choice in the window menu when a user can change the size of a window.

Guidelines

Required

Enlarge the window to the largest size possible for the current view when the user chooses Maximize from the window menu or its associated window icon.

Required

Before maximizing a window, save its state (including its size and position) for use when the window is restored.

Required

Do not make the Maximize choice available when the window is maximized.

Recommended

If the user does not override the default maximum size of the window, make it smaller than the size of the workspace.

Maximize Button

Required

If you provide a Maximize choice, also provide a maximize button that implements the Maximize choice on the title bar when the window is not currently maximized, and the user has not specified otherwise.

Required

Replace the maximize button with a restore button that implements the Restore choice when a window is maximized.

Essential Related Topics

For more information, see the Window Menu reference page.

Supplemental Related Topics

For more information, see the Minimize (Choice) , Restore (Choice) , and Size (Choice) reference pages.

Menu (Control)

NAME

Menu (Control) -Reference

Description

A menu is a control that contains a group of choices, generally of any type. Menus are either pop–up menus, tear–off menus, menu bars, or cascaded menus.

A cascaded menu is a menu displayed from a cascading choice and may be a pull–down menu, an option menu, or a submenu.

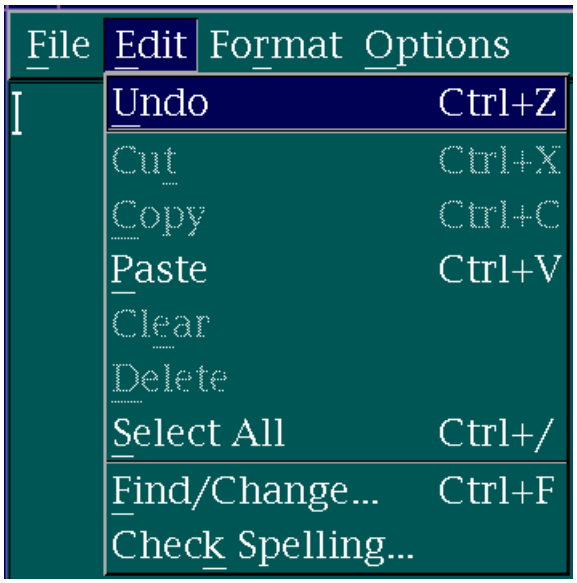
A submenu is a cascaded menu displayed from a tear–off menu, a pop–up menu, or another cascaded menu.

A menu system is a spring–loaded system whose cascaded controls are all menus.

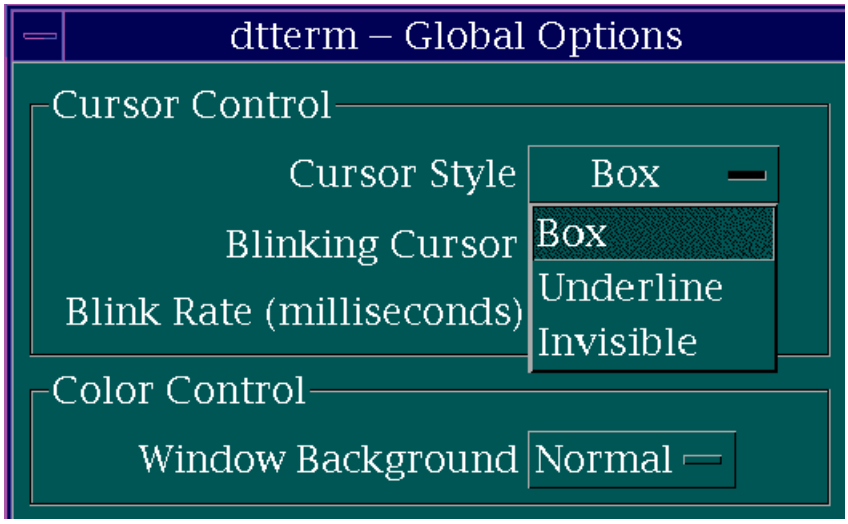
Menu Types illustrates the various menu types.

Figure 1 Menu Types

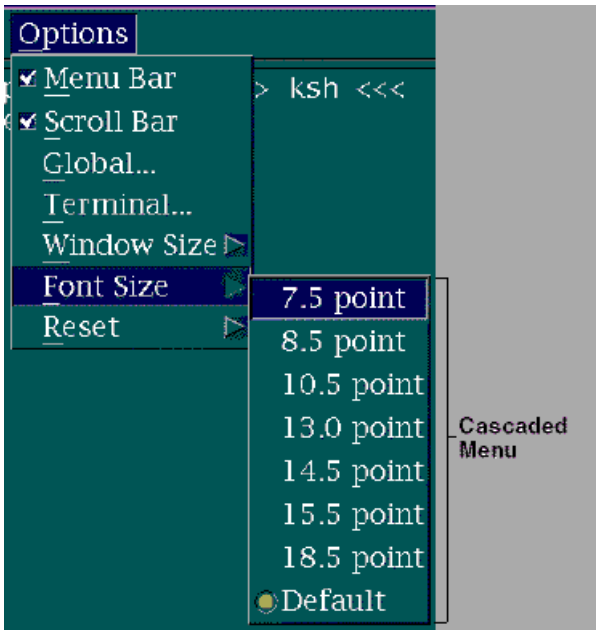
Pull–down menu:



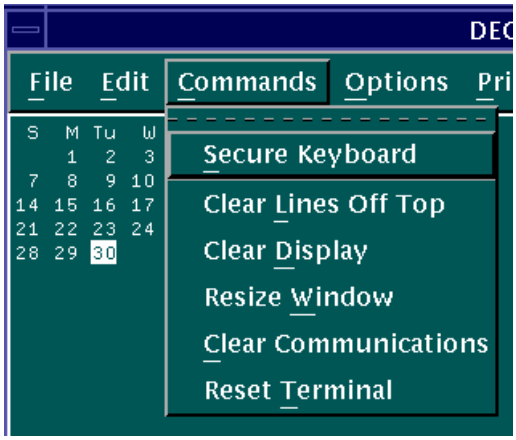
Option menu:

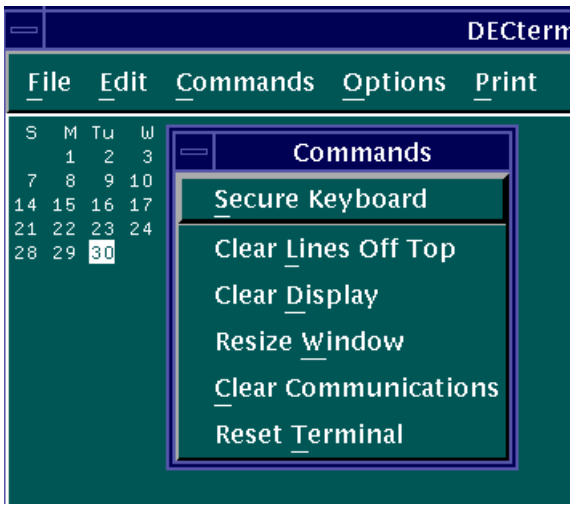


Cascaded menu:



Tear–off menu:


DEC
File
Edit
Commands
Options
Pri
S M Tu W
1 2 3
7 8 9 10
14 15 16 17
21 22 23 24
28 29 30
Secure Keyboard
Clear Lines Off Top
Clear Display
Resize Window
Clear Communications
Reset Terminal


DECterm
File
Edit
Commands
Options
Print
S M Tu W
1 2 3
7 8 9 10
14 15 16 17
21 22 23 24
28 29 30
Commands
Secure Keyboard
Clear Lines Off Top
Clear Display
Resize Window
Clear Communications
Reset Terminal

Pop–up menu:

When to Use

Recommended

Use a menu to present choices, especially standard choices, in a consistent manner.

Recommended

Use a menu to provide choices when there is no space for the choices in the window itself, or when the display of the choices would be awkward or would remove attention from the main controls that the user needs to perform a task.

Guidelines

The following guidelines apply to languages that read from left to right. For information on bidirectional and vertical language support, see Chapter 11.

Required

Place only action, cascading, dialog, and value choices in a menu.

Organization

Recommended

Place related choices together in a menu.

Recommended

Use separators to distinguish groups of related choices.

Recommended

Place application–specific choices in a menu following a group of related predefined choices.

Recommended

Allow a user to add, remove, or reorder choices on a menu.

Optional

Allow a user to add or remove cascaded menus and to reorganize the allocation of choices to menus.

Submenus

Recommended

Avoid placing more than 15 choices in a menu, unless requested by the user. Use submenus to reduce the number of choices in a menu.

Recommended

Avoid using more than one level of submenus.

Recommended

Design submenus to contain at least three choices. If only one or two choices would be in the submenu, place the choices in the parent menu except where the menu can be customized by the user to include additional choices.

Labels

Recommended

Do not use the label of a cascading choice as the label of a choice within its corresponding cascaded menu. Instead, consider the label as the title for the choices in the corresponding menu. Choose a label for the cascading choice that accurately describes the purpose of the menu.

Mnemonics

Required

Provide the predefined mnemonic for each predefined textual choice in a menu.

Required

Provide a unique mnemonic for each application−specific textual choice in a menu, unless there is no meaningful unique mnemonic.

Availability of Choices

Required

Except in the Selected menu or in a pop−up menu (or a menu torn off from them), do not add or remove choices from a menu to indicate unavailability of choices. Instead, display unavailable choices with unavailable emphasis.

Required

If a choice is not available to a particular user due to an external factor over which the user has no control, do not include or reserve space for the choice in a menu.

Required

Display unavailable emphasis on action, value, and dialog choices in a menu that cannot be activated or toggled in the current context.

Size

Required

A cascaded menu should be just large enough so that the labels of all the choices in the menu are fully displayed.

Layout

Required

Organize menus in either rows or columns.

Required

If you provide cascading choices in a menu other than the menu bar, organize the menu in columns.

Navigation

Required

Support the use of directional keys for navigation in a menu system when the menu containing the focus is not spring sensitive.

Recommended

Support the use of directional keys for navigation when focus is in a menu system and no mouse button is pressed.

Navigate Down

Required

In a menu organized in columns, make `[darr ]` move the cursor to a choice in the row below, wrapping at the bottom row within the menu.

Required

In a menu organized in rows, but not on a cascading choice, make `[darr ]` move the cursor to the choice below, wrapping at the bottom edge within the menu.

Navigate Up

Required

In a menu organized in columns, make `[uarr ]` move the cursor to a choice in the row above, wrapping at the top row within the menu.

Required

In a menu organized in rows, but not on a cascading choice, make `[uarr ]` move the cursor to the choice above, wrapping at the top edge within the menu.

Navigate Left

Required

In a menu organized in rows or columns, whose parent is not a menu, make `[larr ]` move the cursor to the choice to the left, wrapping at the left edge within the menu.

Required

In a menu organized in rows or columns, but not at the left edge, make `[larr ]` move the cursor to the choice to the left.

Required

At the left edge of a menu organized in rows or columns, whose parent menu is organized in columns, make `[larr ]` remove the menu and move the cursor to its associated cascading choice.

Required

At the left edge of a menu organized in rows or columns, cascaded from a menu−bar item, make `[larr ]` remove the menu and activate the menu−bar item to the left, first wrapping within the menu bar, if necessary.

Navigate Right

Required

In a menu organized in columns, on a cascading choice, make `[rarr ]` activate the choice.

Required

In a menu organized in rows or columns, whose parent is not a menu, but not on a cascading choice, make [rarr] move the cursor to the choice to the right, wrapping at the right edge within the menu.

Required

In a menu organized in rows or columns, but not at the right edge, and not on a cascading choice, make [rarr] move the cursor to the choice to the right.

Required

At the right edge of a menu organized in rows or columns, whose cascaded menu ancestors are all organized in columns, and which is ultimately descended from a menu–bar item, but not on a cascading choice, make [rarr] remove all the cascaded menus and activate the menu–bar item to the right, first wrapping within the menu bar, if necessary.

Visual Guidelines

Required

When a cascading choice is used in a menu whose choices are aligned vertically, display a right–pointing arrow to the right of the choice and align the right edge of the arrow with the right edge of the menu.

Focus and Default Choices

Required

If a spring–loaded menu is displayed and is not spring sensitive, give it focus unless the focus is in a descendant spring–loaded menu.

Required

When focus moves to a menu that has just been displayed, place the active cursor on the default choice in the menu. If not determined in any other specified way, make the default choice the first choice in the menu or, if that is a tear–off choice, the following choice.

Placement

Required

When a submenu is displayed from a menu–bar item, place it, if possible, immediately below the menu–bar item, with the left edges of the menu and menu–bar item aligned.

Required

When a submenu is displayed from a menu organized as a single column, place the submenu, if possible, so that its left edge is aligned with the right edge of the parent menu.

Recommended

When a submenu is displayed from a menu whose choices are aligned vertically, place the submenu, if possible, so that its default menu item is directly to the right of the right arrow in the cascading choice from which the submenu was displayed.

Keyboard Activation and Toggling

Required

When the active cursor is on an available choice in a menu and the user presses Select, Space, Ctrl Space, Enter, CtrlEnter, or keypadEnter, activate or toggle the choice.

Essential Related Topics

For more information, see the Cascading (Choice Type) , Choice , and Spring–Loaded (Control Type) reference pages.

Supplemental Related Topics

For more information, see the Emphasis (Cue) , Menu Bar (Menu Type) , Mnemonic , Pop–Up Menu (Menu Type) , Pull–Down Menu (Menu Type) , and Tear–Off Menu (Menu Type) reference pages.

Menu Bar (Menu Type)

NAME

Menu Bar (Menu Type) –Reference

Description

A menu bar is a menu displayed across the top of the client area of a window.

A menu–bar system is a spring–loaded system that consists of a menu bar and any menus cascaded from it.

Figure 1 Menu Bar

When to Use

Required

Provide a menu bar if you provide any of the action choices included in the File menu, other than Close or Exit.

Required

Provide a menu bar if you provide any of the action choices included in the View menu.

Recommended

Provide a menu bar if you provide any of the action choices included in the Help menu.

Recommended

Provide a menu bar when a window will provide more than six action choices.

Guidelines

Required

Place only cascading choices in a menu bar.

Required

A cascading choice in a menu bar leads to a pull–down menu.

Menu–Bar Content

Required

In conjunction with the **Menu Guidelines** reference page, use Menu–Bar Choices to decide which choices to include in a menu bar and how to organize them.

Table 1 Menu–Bar Choices

	Mnemonic	Item	Keyboard Function
Required	F	File	`AltF`
Recommended	S	Selected	`AltS`
Required	E	Edit	`AltE`

Optional	I	Insert	AltI
Required	V	View	AltV
Recommended	O	Options	AltO
Optional	o	Tools	AltO
Required	W	Window	AltW
Required	H	Help	AltH

Recommended

Add any application–specific menu–bar choices between the last standard menu–bar choice that is provided and the Help choice.

Recommended

When a window has both a menu bar and push buttons, all action choices available on push buttons should also be included on a pull –down menu (or a menu cascaded from one) available in the window.

Directional Navigation

Required

In a menu bar, make [larr] move the cursor to the choice to the left, wrapping at the left edge within the menu bar.

Required

In a menu bar, make [rarr] move the cursor to the choice to the right, wrapping at the right edge within the menu bar.

Required

In a menu bar, make [darr] activate the cursored cascading choice, display the associated pull–down menu, and move focus to it.

Navigation to and from the Menu Bar

Required

When focus is in a control in a window and the user presses F10 or Shift Menu, navigate to the first item in the menu bar.

Required

If a window has a menu bar, but focus is in a different spring–loaded system originating from the client area of that window, then make pressing F10 or Shift Menu deactivate the spring–loaded system before navigating to the first item in the menu bar.

Required

If the focus is in a menu–bar system, make pressing F10 or Shift Menu deactivate the system.

Placement and Layout

Required

Place the menu bar at the top of the client area of a window and extend it the width of the entire window.

Required

Align menu–bar items horizontally and lay them out left to right. For information on bidirectional and vertical language support, see Chapter 11.

Recommended

Allow a user to hide the menu bar if the functionality it provides is available through other means (for example, push buttons and pop–up menus).

Visual Guidelines

Required

Do not display a graphic (in particular, a down−arrow graphic) along with a cascading choice in a menu−bar item.

Essential Related Topics

For more information, see the Cascading (Choice Type) , Pull−Down Menu (Menu Type) , and Spring−Loaded (Control Type) reference pages.

Supplemental Related Topics

For more information, see the Choice , Edit (Menu) , File Menu , Help (Menu/Action Choice) , Options (Menu) , Selected Menu , and View Menu reference pages.

Menu Cascade Button (Control)

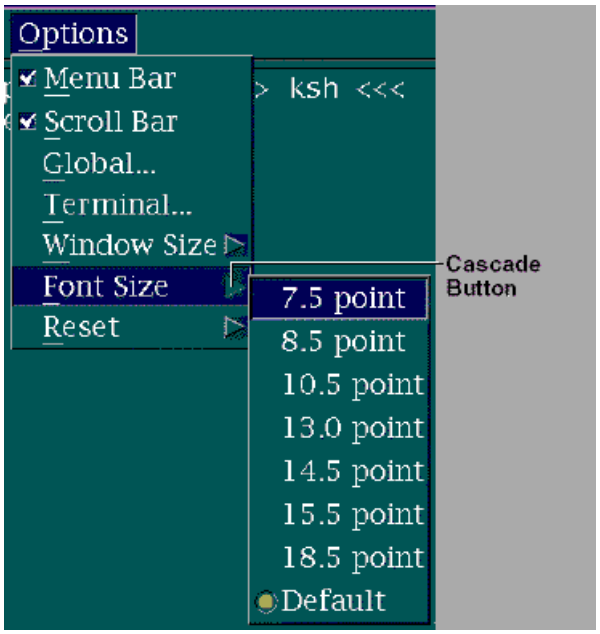
NAME

Menu Cascade Button (Control) –Reference

Description

A menu cascade button is a control that contains a cascading choice from which a pull–down menu can be displayed.

Figure 1 Menu Cascade Button



When to Use

Recommended

Use a menu cascade button to provide choices that apply to a control or group of controls within a window.

Guidelines

Required

Do not use menu cascade buttons at the top of a window in place of a menu bar.

Required

Do not use menu cascade buttons to provide choices that apply to the window or the controls in a window as a whole.

Keyboard Navigation

Required

When focus is on a menu cascade button and the user presses `Select`, `spacebar`, or `Ctrl spacebar`, display the associated pull–down menu and move focus to it.

Optional

When focus is on a menu cascade button and the user types `[darr ]`, you can display the associated pull–down menu and move focus to it.

Visual Guidelines

Required

Except in a window menu button, display a down–arrow graphic to the right of the cascading choice in a menu cascade button. For information on bidirectional and vertical language support, see Chapter 11.

Essential Related Topics

For more information, see the **Cascading (Choice Type)** , Choice , and Pull–Down Menu (Menu Type) reference pages.

Menu Guidelines

NAME

Menu Guidelines –Reference

Description

The following menu guidelines specify how to interpret the menu description included in the reference pages for the pop–up and standard pull –down menus.

Guidelines

Inclusion

Required

> If a choice for a described menu is marked *Required*, then your menu should include that choice if that functionality can be reasonably provided by your application.

Recommended

> If a choice for a described menu is marked *Recommended*, then your menu should include that choice if that functionality can be reasonably provided by your application.

Optional

> If a choice for a described menu is marked *Optional*, then your menu can include that choice if that functionality can be reasonably provided by your application.

Usage

Required

> If your menu includes a choice whose functionality is the same as any choice listed for that menu, use the predefined label specified for the choice.

Required

> If your menu includes two or more of the choices listed for that menu, they should appear in the order listed.

Required

> The choice in the described menus are separated into groups. If your menu includes items listed in different groups, use a separator to separate the items.

Required

> If a choice for a described menu has a mnemonic or shortcut key specified, then if the choice appears in your menu, use that mnemonic or shortcut key.

Required

> If a choice for a described menu has a mnemonic specified, but that choice does not appear in your menu, do not use the mnemonic for other choices in the menu.

Recommended

> If a choice for a described menu has a shortcut key specified, but that choice does not appear in your menu, do not define that shortcut key in your application.

Essential Related Topics

For more information, see the **Edit (Menu)** , File Menu , Help (Menu/Action Choice) , Mnemonic , Pop–Up Menu (Menu Type) , Selected Menu , Shortcut Key , and View Menu reference pages.

Message

NAME

Message –Reference

Description

A message is a persistent cue that consists of information displayed in a secondary window in response to an unexpected event, a situation in which something undesirable could occur, or when there is additional status information on a process that has completed. The five types of messages are error message, information message, in–progress message, question message, and warning message.

When to Use

Recommended

Use a message to report unexpected or undesirable situations to the user that require some user action or decision.

Recommended

Use a message to indicate that a task has completed successfully, but there is additional information available about the completion status of the task.

Recommended

Use a message to indicate that a process is in progress.

Guidelines

Message Text

Required

Phrase message text so that a user clearly understands what caused the message as well as what action, if any, can be taken to correct the situation that caused the message.

Required

When wording the messages displayed by your application, do not assume that the user has any expert knowledge about computer systems in general or the operating system in particular. You can assume that the user has knowledge about basic interface terms that could have been learned through tutorials, help information, and user documentation. However, avoid terminology that is typically understood only by an expert or knowledgeable computer user unless the application is specifically targeted at computer professionals. Also, messages returned to your application by the underlying system should not be passed directly to the user, but instead should be interpreted into language the novice user can understand.

Recommended

Avoid phrasing messages in a way that requires a Yes or No response from the user. If you use Yes and No push buttons, avoid using negatives in the message text. For example, do not use the message:

```
Are you sure you don't want to save the file?
```

Instead, use:

```
File has been modified.
Choose 'Discard' to throw away changes or
'Save' to save the file and then quit.
```

Message Windows

Required

If a window associated with the task or element for which the task is being performed is open, display the message in a secondary window that depends on the associated window.

Required

If no associated window is open for a task being performed and for which an urgent message must be displayed, then display the message in a secondary window and, optionally, display the task's window if such a window exists.

Required

If no associated window is open for the task being performed and for which a nonurgent message needs to be displayed, then place the message in a secondary window, but do not display it. Optionally, change the appearance of the icon through which the window can be displayed if such a window exists.

Recommended

If you provide a message identifier in a message window, place it in the bottom rightmost corner of the message and display it in a smaller font than the rest of the message text. For information on bidirectional and vertical language support, see Chapter 11.

Providing Controls in Message Windows

Recommended

Provide a Help push button in all messages, except those that contain trivial or self–explanatory messages. Design your application with both the expert and novice user in mind. The novice user must be able to access additional explanatory information, the circumstances under which it was displayed, and what the user should do in response to the message.

Recommended

Use **Push Buttons for Various Message Types** to determine which predefined push buttons to use in messages.

Table 1 Push Buttons for Various Message Types

Type of Window	Push Buttons (Listed in relative order of appearance)	Result of Close Choice in Window Menu	Result of `Enter` Key	Result of `Esc` Key
Error message Use for more complex decisions. The user can cancel the operation or choose a destructive action.	*action name* One or more push buttons that contain the names of the actions must be available. (optional) Retry Use Retry if it is practical. Cancel Use Cancel if it is practical. Affected user data must be returned to its original state or left in a useful state. Help The help window must have at least two topics from the set: *action*, Retry, Cancel.	Cancel (if supported, otherwise a nondestructive action)	Retry (if supported, otherwise a nondestructive action)	Cancel (if supported, otherwise a nondestructive action)
Question message Use when the user can resolve the problem in a nondestructive way and will not lose data.	Yes Yes and No each represent an action, or Yes represents an action and No is Cancel. No Help	Choice of Yes/No that does not lose data	Choice of Yes/No that does not lose data	Choice of Yes/No that does not lose data
Information message	OK Help	OK	OK	OK
In–progress message	Close Stop (optional) Pause and Resume (both optional) Help	Close	Pause if supported, otherwise Close	Stop if supported, otherwise Close
Warning message	*Action name* One or more push buttons that contain the names of the actions must be available. (optional) Continue The window must have at least one action that continues the request and one action that cancels the request.	Cancel (if supported, or else a nondestructive action)	Nondestructive action	Cancel (if supported, otherwise a nondestructive action)

Cancel Help

Essential Related Topics

For more information, see Chapter 11 and the Action Message , Information and Message Areas (Area) , Information Message , In–Progress Message , and Push Button (Predefined) reference pages.

Supplemental Related Topics

For more information, see the Warning Signal reference page.

Minimize (Choice)

NAME

Minimize (Choice) –Reference

Description

Minimize is an action choice that removes a window and all of the secondary windows associated with that window from the workspace and displays the window icon that represents that window. If appropriate, the Minimize choice may also be represented by a minimize button in the title bar of a window.

When to Use

Required

Provide the Minimize choice in the window menu of a primary window.

Required

Do not provide the Minimize choice for secondary windows.

Guidelines

Required

Remove the window from the display and display its window icon, if not already displayed, when the user chooses Minimize from the window menu or the associated window icon.

Required

Before minimizing a window, save its state, including its size and position, along with the state of each of its secondary windows to be used when the window is restored.

Required

Do not make the Minimize choice available when the window is minimized.

Required

Even if a window has been minimized, design the application to continue to perform any application tasks initiated through that window.

Recommended

When a window contains a view of an object, maintain in−use emphasis on all icons that represent the object even when the window is minimized.

Recommended

When a user minimizes a window that has been minimized previously, and its window icon is not displayed, display the window icon where it was last displayed before being restored.

Minimize Button

Required

If you provide a Minimize choice and the user has not specified otherwise, also provide a minimize button that implements the Minimize choice on the title bar.

Essential Related Topics

For more information, see the **Window Icon** , Window Icon Box , and Window Menu reference pages.

Supplemental Related Topics

For more information, see the Emphasis (Cue) , Maximize (Choice) , Restore (Choice) , and Size (Choice) reference pages.

Mnemonic

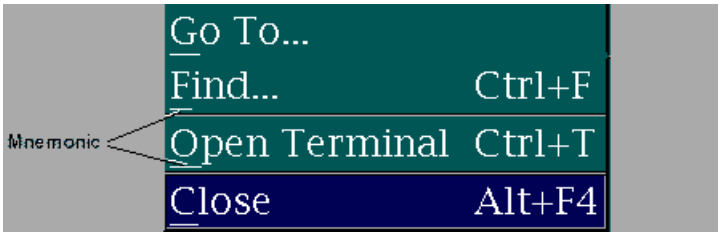
NAME

Mnemonic -Reference

Description

A mnemonic is a single character represented by a key on the keyboard that is associated with a textual label and used to navigate to the group, control, or choice associated with it, and possibly to activate or toggle it.

Figure 1 Mnemonic



When to Use

The following guidelines apply to languages that read from left to right. For information on bidirectional and vertical language support, see Chapter 11.

Required

Provide a predefined mnemonic for each predefined textual choice in a menu.

Required

Provide a mnemonic for each textual choice in a menu, unless you cannot find a meaningful unique mnemonic.

Recommended

Provide a mnemonic for the group heading associated with a tab group.

Recommended

Provide a mnemonic for the label associated with a text field.

Recommended

Provide a mnemonic for each push button with a textual label.

Recommended

Provide a mnemonic for each radio button and check box with a textual label in a tab group.

Guidelines

Uniqueness

Required

Make mnemonics unique among all choices in a menu.

Required

Make mnemonics unique among all radio buttons and check boxes in a tab group.

Required

Make mnemonics unique among all push buttons, group headings, and menu–bar items.

Required

If a mnemonic is an alphabetic character, allow the user to press either the uppercase or lowercase character.

Assignment and Display of Mnemonics

Required

The character used as a label's mnemonic must appear in the label and be highlighted within it, generally by underlining it.

Recommended

If choices are duplicated within an application, assign the same mnemonic to the choices.

Recommended

For textual labels that do not have a mnemonic defined, assign mnemonic characters by applying the first applicable guideline that follows:

Use the first character of the label, or the first character of one of the words in a multiple–word label, unless those characters have been assigned as mnemonics for other choices.

Use a consonant in the label.

If all the consonants in a label have been used as mnemonics for other labels, use any other character in the choice name.

If all the characters in a label have been used as mnemonics for other labels, add a unique character to the label text and make it the mnemonic. Display the character in parentheses after the label.

Optional

A group of labels that require unique mnemonics may have their mnemonics generated by sequentially assigned numbers or letters to the labels, prepending the uniquely assigned character to its corresponding label (along with any bracketing or separator characters) and underlining it.

Mnemonic Behavior

Required

When the active cursor is in a menu and the user types the mnemonic of a choice within that menu, move the cursor to the choice and activate or toggle it.

Recommended

When the active cursor is in a tab group and the user types the mnemonic of a choice within that tab group, activate or toggle the choice. If an explicit focus policy is in use, move focus to the choice.

Recommended

When the cursor is in a window or a spring–loaded system popped up or cascaded from that window, and the user presses `Alt` plus a mnemonic associated with a control in the window, activate or navigate to that control after first deactivating any spring–loaded system that contains the cursor. For more information, see `Alt` Mnemonic Behavior.

`Alt` Mnemonic Behavior

Required

When the user presses `Alt` plus a mnemonic associated with a menu–bar item, activate that menu–bar item and move focus to the menu pulled down from it.

Required

When the user presses `Alt` plus a mnemonic associated with an option menu or menu cascade button, pull down the corresponding menu and move focus to it.

Recommended

When the user presses `Alt` plus a mnemonic associated with a push button, activate the push button. If using an explicit focus policy, and the focus is not in the data area affected by the push button's action, move focus to the push button.

Recommended

When the user presses `Alt` plus a mnemonic associated with a text field or tab group (other than for a control described previously) and an explicit focus is in use, move focus to that tab group or text field.

Essential Related Topics

For more information, see the **Edit (Menu)** , File Menu , Help (Menu/Action Choice) , Selected Menu , View Menu , and Window Menu reference pages.

Supplemental Related Topics

For more information, see the First–Letter Cursor Navigation , Keyboard (Device) , and Shortcut Key reference pages.

Mouse (Device)

NAME

Mouse (Device) –Reference

Description

A mouse includes the following:

Pointing device

> An input device, such as a mouse, trackball, or joystick, used to move a pointer on the screen. The operating environment may specify a mode of operation in which the directional keys act as a pointing device with other keys acting as pointer buttons.

Mouse

> A commonly used pointing device that has one or more buttons that a user presses to interact with the operating environment.

This book uses a model mouse as the pointing device (see Appendix C for more information).

Guidelines

Required

> Provide support for a mouse as a pointing device.

Required

> Support the following virtual mouse buttons:
>
> SELECT
>
> > The virtual pointer button used for selection, activation, toggling, and forms of direct manipulation not assigned to other virtual mouse buttons.
>
> ADJUST
>
> > The virtual pointer button used for adjusting a selection.
>
> TRANSFER
>
> > The virtual pointer button used for data transfer operations.
>
> MENU
>
> > The virtual pointer button used to obtain pop–up menus.

Required

> The period of time between button release and the subsequent press in a multi–click event is user adjustable, and the default is not less than 100 milliseconds.

Required

> Do not assign double–click functions to choices on which the user will typically perform multiple single clicks, such as on push buttons.

Three–Button Mouse Bindings

Required

Support a 3–button mouse.

Required

Treat a 2–button mouse as a 3–button mouse if it has been configured so that the third button is available by chording the other two buttons (for example, with MB1+MB2 treated as MB3).

Required

On a 3–button mouse, bind MENU to MB3.

Required

Allow the user to choose whether the bindings on a 3–button mouse support one of the following:
SELECT and TRANSFER integrated on MB1 and TRANSFER bound to MB2 (recommended)
SELECT bound to MB1 and TRANSFER bound to MB2
SELECT and TRANSFER integrated on MB1 and ADJUST bound to MB2
SELECT and TRANSFER integrated on MB1 and nothing bound to MB2

Required

If you support binding TRANSFER to MB2 and also integrate TRANSFER with SELECT on MB1, then whenever MB1 is pressed and moved, interpret the action as a drag–and–drop operation. Ensure that when MB2 is pressed and the pointer is moved along the same path, it will have an identical effect.

Required

If you support binding ADJUST to MB2, design your application so that using ADJUST is identical to using `Shift` SELECT.

Optional

If nothing is bound to MB2, your application can use it for application–specific functionality; however, this functionality must be accessible in other ways if the user chooses a different binding for MB2.

Two–Button Mouse Bindings

Required

Support a 2–button mouse.

Required

Allow the user to choose whether the bindings on a 2–button mouse support one of the following:
SELECT and TRANSFER integrated on MB1 and MENU on MB2
SELECT and TRANSFER integrated on MB1, TRANSFER bound to MB2, and MENU bound to `Alt` MB1
SELECT bound to MB1, TRANSFER bound to MB2, and MENU bound to `Alt` MB1

One–Button Mouse

Recommended

Support a 1–button mouse, with SELECT and TRANSFER integrated on MB1 and MENU bound to `Alt` MB1.

Essential Related Topics

For more information, see Appendix C and the **Data Transfer** and Pointer (Predefined) reference pages.

Supplemental Related Topics

For more information, see the Default Action and Pointer reference pages.

Move To (Dialog Choice)

NAME

Move To (Dialog Choice) -Reference

Description

Move To includes the following:

Move To

A dialog choice that leads to a file selection dialog in which users can specify where files or objects should be moved.

Move To

On the File (or Edit) menu, a choice that allows the user to update the underlying file or object being viewed and to move it elsewhere while still keeping open a view of it.

Move To

On the Selected menu, allows the user to move the selected objects to a specified place.

Guidelines

Required

When appropriate, the File Selection dialog should allow the user to differentiate between moving the link or reference to a file or object, moving its underlying storage location, or both.

Recommended

When the user chooses Move To in the File menu, the File Selection dialog should allow the user to move the file or object to the trash, in which case the window is closed or replaced by a view onto a new object.

Recommended

When the user chooses Move To in the File menu, the File Selection dialog should allow the user to move the file or object to the clipboard, in which case the window is closed or replaced by a view onto a new object.

Recommended

When the user moves multiple files or objects, the File Selection dialog should allow the user to name only the place to which the files or objects are to be moved, but not to provide new names for each of them. If any new names are needed, they should be generated from their current names in a way that does not conflict with any names at their destination.

Essential Related Topics

For more information, see the **File Menu** and Selected Menu reference pages.

Supplemental Related Topics

For more information, see the Menu Guidelines reference page.

Multilevel Selection Technique

NAME

Multilevel Selection Technique –Reference

Description

Multilevel selection techniques are mouse–based variants of other group techniques in which elements are selected in parts.

Within scopes whose elements could be seen to be divided into groups (at various levels), multilevel selection techniques can be used to select a higher–level part of many elements at a time by using multi–press or multi–click on the SELECT or ADJUST buttons.

The number of clicks or presses control the level. For example, within text, the following can occur:

- Double–click selects a word.
- Double–press, moving the pointer, and releasing selects parts of words at a time.
- Higher numbers of presses or clicks may select lines, sentences, paragraphs, or all the text in the scope.

In these examples, the use of the multilevel technique selects ranges of characters –– that is, text supports multilevel range techniques.

Any of the standard techniques (including adjustment) may be used as a multilevel technique. For example:

- A multilevel range point technique would involve a multiple click to select a part of many elements (for example, double–pressing the SELECT button to select a word).
- A multilevel range adjust swipe technique would adjust the selection by a part of many elements at a time (for example, double–pressing the ADJUST button, moving the pointer, and releasing the ADJUST button to adjust a text selection a word at a time).

In addition, you can use `Ctrl` to augment multiple presses or clicks of the SELECT button to force use of toggle mode. Multilevel Selection Technique Methods lists the techniques available.

Table 1 Multilevel Selection Technique Methods

Technique	Method
Multilevel point technique	Multi–click SELECT
Multilevel point technique, forcing toggle mode	Multi–click `Ctrl` SELECT
Multilevel swipe technique	Multi–press SELECT, move pointer, release SELECT
Multilevel swipe technique, forcing toggle mode	Multi–press `Ctrl` SELECT, move pointer, release SELECT
Multilevel click technique	Multi–click SELECT, move pointer, click ADJUST
Multilevel click technique, forcing toggle mode	Multi–click `Ctrl` SELECT, move pointer, click ADJUST
Multilevel adjust click technique	Multi–click ADJUST
Multilevel adjust swipe technique	Multi–press ADJUST, move pointer, release ADJUST

Guidelines

Required

After using a multilevel point technique:

Identify the pointer position as the anchor point.

Define the selection region to consist of the part of elements (depending on the number of clicks), if any, containing the pointer.

Define the anchor element to be the element in the region nearest to the pointer.

Define the anchor region to be the selection region.

Place the active cursor as follows:

At the anchor point, if it can be placed there

On the anchor element, if it is an element cursor

Where the cursor previously was in the scope, if there is no anchor element

Required

When using a multilevel click or swipe technique:

Identify the pointer position at initiation of the technique as the anchor point.

Define the selection region to consist of all elements that would be selected when using the corresponding click or swipe technique.
Expand the selection region if all elements in its part (depending on the number of presses) are also included in the region.
Define the anchor element to be the element in the selection region nearest to the pointer.
Define the anchor region to be the selection region that would be obtained if a multilevel point technique had been used instead at the point at which the technique was initiated.
Place the active cursor as follows:
At the pointer position when the technique was finished, if it can be placed there
On the element at the other end of the range from the anchor element, if you are using an element cursor, the region is not empty, and range technique is in effect
On an element within the selection region, if you are using an element cursor, the region is not empty, and area technique is in effect
On the last element touched, if you are using an element cursor, the region is not empty, and touch technique is in effect
Where the cursor previously was in the scope, if none of the above are true

Required

When using a multilevel point, swipe, or click technique in select mode:
Select all the elements in the selection region.
Deselect all other elements in the scope.

Required

When using a multilevel point, swipe, or click technique in toggle mode, toggle all elements in the selection region, based on the toggling policy.

Required

When using an adjustment technique to adjust a selection initially made using a multilevel technique, expand the selection region so that if an element is in the region, all elements in its part (depending on the number of times the SELECT button was pressed in the initial technique) are also included in the region.

Required

When using a multilevel adjust click or swipe technique:
Identify the pointer position at initiation of the technique as the anchor point.
Define the selection region to consist of all elements that would be selected when using the corresponding adjust click or swipe technique, but then expanded so that one of the following occurs:
If an element is added to the region, also include all elements in its part (depending on the number of times the ADJUST button is pressed) in the region.
If an element is removed from the region, place it back into the region, unless all elements in its part (depending on the number of times the ADJUST button was pressed) are also included in the region.
Define the anchor element to be the element in the selection region nearest to the pointer.
Define the anchor region to be the selection region that would be obtained if a multilevel point technique had been used instead at the point at which the technique was initiated.

Essential Related Topics

For more information, see the Adjustment Techniques , Selection Modes , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Selection Models and Selection Policies reference pages.

Multipage Control (Control Type)

NAME

Multipage Control (Control Type) –Reference

Description

Multipage control includes the following:

Multipage control

A kind of control that shows one or more pages at a time from among a larger set of pages, each page containing one or more controls.

Paging control

Associated with multipage control, switches pages within a multipage control.

When to Use

Recommended

Use a multipage control to organize groups of controls that can be logically subdivided into groups, and that should not or need not all be displayed at once.

Recommended

Use a multipage control instead of placing related controls in separate windows connected by dialog choices on push buttons.

Recommended

Use a multipage control instead of scrolling an area that contains a large number of controls.

Recommended

Use a multipage control to organize groups of controls in a window according to some application–specific order, for example, sorted in ascending order by page number.

Recommended

Do not use a multipage control when a set of controls does not fit within a viewing area. Use a More choice instead.

Guidelines

Tab Navigation

Required

Organize the controls within each page of a multipage control into one or more tab groups.

Required

Do not use tab group navigation to switch pages of a multipage control.

Required

When forward tab group navigation is in use, traverse all tab groups on a visible page before navigating off the page.

Required

When tab group navigation moves focus into a multipage control, place focus, initially, in one of the visible pages.

Paging Key Navigation

Recommended

If focus is within a page in a multipage control, then allow pressing `PageUp`, `Ctrl PageUp`, or `PageLeft` (or `PageDown`, `Ctrl PageDown`, or `PageRight`) to switch to an appropriate subsequent (or preceding) page, unless the control uses the key within the page that has focus.

Paging Control

Recommended

When a small fixed set of distinct pages are to be used in a multipage control, use a list, drop–down list, or option menu as the paging control.

Recommended

When a small set of pages, some or all of which are similar, are to be used in a multipage control, use a spin box as the paging control.

Recommended

When a multipage control does not itself support organization of the pages in groups, use an option menu with cascading menus as the paging control.

Essential Related Topics

For more information, see the **Control** and Notebook (Control) reference pages.

Navigation (CDE)

NAME

Navigation (CDE) -Reference

Description

Navigation is the movement of pointers and cursors within the interface to indicate where actions should occur.

Guidelines

Mouse−Based Navigation

Required

When the keyboard focus policy is explicit, allow the user to press the SELECT button on a component to move focus to it, except for components, such as scroll bars, that are used to adjust the size and location of other elements.

Required

When the pointer is on a menu, allow the user to press the SELECT button to activate the menu in a spring−loaded manner.

Required

When the pointer is on an element with an inactive pop−up menu and the context of the element allows the pop−up menu to be displayed, allow the user to press the MENU button to activate the pop−up menu in a spring−loaded manner.

Required

If the user takes an action to display a pop−up menu, and a menu can be displayed for both an inner element and an outer element that contains the inner element, display the pop−up menu for the internal element. Be sure that the pop−up menu for an internal element is always accessible.

Required

Once a pop−up menu is displayed, make the MENU button behave just as the SELECT button does for any menu system.

Required

Make the SELECT button available from within posted pop−up menus. It should behave just as in any menu system.

Required

When a menu is popped up or pulled down, place the location cursor on the menu's default entry or on the first entry in the menu, if there is no default entry.

Required

Remove a spring−loaded menu system when the mouse button that activated it is released, except when the button is released on a cascading button in the menu hierarchy.

Required

While a spring−loaded menu is popped up or pulled down, moving the pointer within the menu system moves the location cursor to track the pointer.

Required

When a spring−loaded menu is popped up or pulled down and the pointer rests on a cascading button, pull down the associated menu and make it traverse. Remove the associated menu, possibly after a short delay, when the pointer moves to a menu item outside of the menu or

its cascading button.

Required

When a spring−loaded menu that is part of the menu bar is pulled down, moving the pointer to any other element on the menu bar unposts the current menu system and posts the pull−down menu associated with the new element.

Required

When a spring−loaded menu is popped up or pulled down, and the button that activated the menu system is released within a component in the menu system, activate that component. If the release is on a cascading button or an option button, activate the associated menu in a posted manner if it was not posted prior to the associated button press.

Required

When the pointer is in an area with a pop−up menu, allow the user to click the MENU button to activate the menu in a posted manner if it was not posted prior to clicking the MENU button.

Required

Once a pull−down or option menu is displayed, pressing the SELECT button in the menu system causes the menu to behave as a spring −loaded menu. This allows the user to switch easily between using a posted menu and a spring−loaded menu.

Required

If a button press unposts a menu and that button press is not also passed to the underlying component, do not pass subsequent events up to and including the button release to the underlying component.

When a button press unposts a menu, the press can be passed to the underlying component. Whether or not it is passed to the underlying component, the press can have additional effects, such as raising and giving focus to the underlying window. If the press is not passed to the underlying component, events up to and including the release must not be passed to that component.

Required

Once a pop−up menu is displayed, pressing the SELECT button or pressing the MENU button in the menu system causes the menu to behave as a spring−loaded menu.

Optional

Pressing or clicking the MENU button on a menu−bar item displays the menu.

Required

Pressing or clicking the MENU button on an option button displays the option menu.

Required

Pressing the SELECT button on a text field causes the text cursor to be inserted at the mouse cursor position.

Keyboard−Based Navigation

Required

In a text component, display the text cursor differently when it has keyboard focus than when it does not. In a text component, the text cursor serves as the location cursor and, therefore, must indicate whether the component has keyboard focus.

Required

If a text component indicates that it has lost the keyboard focus by hiding the text cursor and if the component subsequently regains the focus, the cursor should reappear at the same position it had when the component lost focus. The text cursor should not change position when a text component loses and then gains keyboard focus.

Required

In a small component, such as a sash, indicate that it has keyboard focus by filling it. No other meaning should be associated with the filled

state.

Required

Pressing `Ctrl Tab` moves the location cursor to the next field, and pressing `Ctrl Shift Tab` moves the location cursor to the previous field. Unless `Tab` and `Shift Tab` are used for internal navigation within a field, pressing `Tab` also moves the location cursor to the next field, and pressing `Shift Tab` also moves the location cursor to the previous field.

Required

Pressing `Tab` (if not used for internal navigation) or `Ctrl Tab` moves the location cursor forward through fields in a window according to the following rules:

If the next field is a control, pressing `Tab` (if not used for internal navigation) or `Ctrl Tab` moves the location cursor to that control.

If the next field is a group, pressing `Tab` (if not used for internal navigation) or `Ctrl Tab` moves the location cursor to a traversable component within the group.

If the next field contains no traversable components, pressing `Tab` (if not used for internal navigation) or `Ctrl Tab` skips the field.

Required

Pressing `Shift Tab` (if not used for internal navigation) or `Ctrl Shift Tab` moves the location cursor backward through fields in the order opposite to that of `Tab` (if not used for internal navigation) and `Ctrl Tab`.

Required

When a window acquires focus, place the location cursor on the control that last had focus in the window, providing that all the following conditions are met:

The window uses an explicit keyboard focus policy.

The window acquires the focus through keyboard navigation or through a button press other than within the client area of the window.

The window had the focus at some time in the past.

The control that last had focus in the window is still traversable.

Required

Wrap field navigation between the first and last fields in the window.

Required

When `[darr ]` and `[uarr ]` are used for component navigation within a field, they should behave according to the following rules:

In a left−to−right language environment, pressing `[darr ]` moves the location cursor through all traversable controls in the field, starting at the upper left and ending at the lower right, then wrapping to the upper left. If the controls are aligned in a matrix−like arrangement, `[darr ]` first traverses one column from top to bottom, then traverses the column to its right, and so on. In a right−to−left language environment, pressing `[darr ]` moves the location cursor through all traversable controls, starting at the upper right and ending at the lower left.

Pressing `[uarr ]` moves the location cursor through all traversable controls in the field in the order opposite to that of `[darr ]`.

Required

When `[rarr ]` and `[larr ]` are used for component navigation within a field, they should behave according to the following rules:

In a left−to−right language environment, pressing `[rarr ]` moves the location cursor through all traversable controls in the field, starting at the upper left and ending at the lower right, then wrapping to the upper left. If the controls are aligned in a matrix−like arrangement, `[rarr ]` first traverses one row from left to right, then traverses the row below it, and so on. In a right−to−left language environment, pressing `[rarr ]` moves the location cursor through all traversable controls, starting at the lower left and ending at the upper right.

Pressing `[larr ]` moves the location cursor through all traversable controls in the field in the order opposite to that of `[rarr ]`.

Required

If a control uses `[rarr ]` and `[larr ]` for internal navigation, pressing `Begin` moves the location cursor to the leftmost edge of

the data or the leftmost element in a left−to−right language environment. In a right−to−left language environment, pressing `Begin` moves the location cursor to the rightmost edge of the data or the rightmost element.

Required

If a control uses `[rarr ]` and `[larr ]` for internal navigation, pressing `End` moves the location cursor to the rightmost edge of the data or the rightmost element in a left−to−right language environment. In a right−to−left language environment, pressing `End` moves the location cursor to the leftmost edge of the data or the leftmost element.

Required

If a control uses `[darr ]` and `[uarr ]` for internal navigation, pressing `Ctrl Begin` moves the location cursor to one of the following:

The first element
The topmost edge of the data
In a left−to−right language environment, the topmost left edge of the data; in a right−to−left language environment, the topmost right edge of the data

Required

If a control uses `[darr ]` and `[uarr ]` for internal navigation, pressing `Ctrl End` moves the location cursor to one of the following:

The last element
The bottommost edge of the data
In a left−to−right language environment, the bottommost right edge of the data; in a right−to−left language environment, the bottommost left edge of the data

Optional

Each time a new window is opened, place keyboard focus in the first field or location within the window or in a default location, if this is appropriate for the particular window.

Required

Pressing `Tab` key moves input focus between push buttons within a group.

Required

The `Ctrl`, `Shift`, and `Alt` keys should modify only the function of other keys or key combinations.

Optional

Pressing `Alt` only to provides access to mnemonics.

Menu Traversal

Required

If the user traverses to a menu while the keyboard focus policy is implicit, temporarily change the focus policy to explicit and revert to implicit whenever the user traverses out of the menu system.

Menus must always be traversable, even when the keyboard focus policy is generally implicit.

Required

Pressing `F10` activates the menu bar−system if it is inactive and the location cursor is placed on the first traversable cascading button in the menu bar. If there are no traversable cascading buttons, the key should do nothing.

Required

When the keyboard focus is in an element with an inactive pop−up menu and the context of the element allows the pop−up menu to be displayed, pressing `menu` activates the pop−up menu and places the location cursor on the default item of the menu or on the first traversable item in the pop−up menu, if there is no default item.

Required

When the keyboard focus is in an option button, pressing `Select` or `Spacebar` displays the option menu and places location cursor on the previously selected item in the option menu; or, if the option menu has been pulled down for the first time, places the location cursor on the default item in the menu. If there is an active option menu, pressing `Return`, `Select`, or `Spacebar` selects the current item in the option menu, unposts the menu system, and returns the location cursor to the option button.

Required

Pressing `[darr ]`, `[uarr ]`, `[larr ]`, and `[rarr ]` traverses through the items in a menu system.

Required

When a menu traversal action traverses to the next or previous component in a menu or menu bar, the order of traversal and the wrapping behavior should be the same as that of the corresponding component navigation action within a field.

Required

If your application uses any two–dimensional menus, do not include any cascading buttons. Cascading buttons in a two–dimensional menu restricts the user's ability to use the keyboard to navigate to all of the elements of the menu.

Required

When focus is on a component in a menu or menu–bar system, pressing `[darr ]` behaves in the following way:

If the component is in a vertical or two–dimensional menu, pressing `[darr ]` traverses down to the next traversable component, wrapping the component within the menu if necessary.

If the component is in a menu bar, and the component with the keyboard focus is a cascading button, pressing `[darr ]` displays its associated pull–down menu and traverses to the default entry in the menu or, if the menu has no default, to the first traversable entry in the menu.

Required

When focus is on a component in a menu or menu–bar system, pressing `[uarr ]` behaves in the following way:

If the component is in a vertical or two–dimensional menu, pressing `[uarr ]` traverses up to the previous traversable component, wrapping within the menu if necessary, and proceeds in the order opposite to that of pressing `[darr ]`.

Required

When focus is on a component in a menu or menu–bar system, pressing `[larr ]` behaves in the following way:

If the component is in a menu bar or two–dimensional menu, but not at the left edge, pressing `[larr ]` traverses left to the previous traversable component.

If the component is at the left edge of a menu bar, pressing `[larr ]` wraps within the menu bar.

If the component is at the left edge of a vertical or two–dimensional menu that is the child of a vertical or two–dimensional menu, pressing `[larr ]` unposts the current menu and traverses to the parent cascading button.

If the component is at the left edge of a vertical or two–dimensional menu that is the child of a menu bar, pressing `[larr ]` unposts the current menu and traverses left to the previous traversable entry in the menu bar. If that entry is a cascading button, pressing `[larr ]` posts its associated pull–down menu and traverses to the default entry in the menu or, if the menu has no default, to the first traversable entry in the menu.

Required

When focus is on a component in a menu or menu–bar system, pressing `[rarr ]` behaves in the following way:

If the component is a cascading button in a vertical menu, pressing `[rarr ]` displays its associated pull–down menu and traverses to the default entry in the menu or, if the menu has no default, to the first traversable entry in the menu.

If the component is in a menu bar or two–dimensional menu, but not at the right edge, pressing `[rarr ]` traverses right to the next traversable component.

If the component is at the right edge of a menu bar, pressing `[rarr ]` wraps within the menu bar.

If the component is not a cascading button and is at the right edge of a vertical or two–dimensional menu, and if the current menu has an ancestor cascading button (typically in a menu bar) from which pressing `[darr ]` displays its associated pull–down menu, pressing `[rarr ]` unposts the menu system pulled down from the nearest such ancestor cascading button and traverses right from that cascading button to the next traversable component. If that component is a cascading button, pressing `[rarr ]` displays its associated pull–down menu and traverses to the default entry in the menu or, if the menu has no default, to the first traversable entry

in the menu.

Required

Allow all menu traversal actions, with the exception of menu posting, to traverse to tear−off buttons in the same way as for other menu entries. Traversal of tear−off buttons needs to be consistent with traversal of other menu items.

Required

If your application uses `F10`, `Menu`, or `Cancel` to unpost an entire menu system and an explicit focus policy is in use, move the location cursor back to the component that had it before the menu system was posted.

Scrollable Component Navigation

Required

Any scrollable components within your application should support the appropriate navigation and scrolling operations. The page navigation keys `Page Up`, `Page Down`, `Control Page Up` (for `Page Left`), and `Control Page Down` (for `Page Right`) scroll the visible region by a page increment. The user needs to be able to view and access the entire contents of a scrollable component.

Required

When scrolling by a page, leave at least one unit of overlap between the old and new pages.

Required

Any keyboard operation that moves the cursor to or in the component, or that inserts, deletes, or modifies items at the cursor location should scroll the component so that the cursor is visible when the operation is complete. The user needs to be able to see the results of moving the location cursor or the effects of operating on the contents of the scrollable component.

Required

If a mouse−based scrolling action is in progress, pressing `Cancel` cancels the scrolling action and returns the scrolling device to its state prior to the start of the scrolling operation.

Essential Related Topics

For more information, see the **Input Focus** , Internal Navigation , and Menu (Control) reference pages.

Supplemental Related Topics

For more information, see the Keyboard (Device) and Mouse (Device) reference pages.

New (Action Choice)

NAME

New (Action Choice) –Reference

Description

New is an action or dialog choice that creates a new file or object of the type that an application uses. When chosen from the File menu, the newly created file or object is displayed as well.

Guidelines

File Menu

Required

The New choice in the File menu either replaces the contents of the current window or opens a new window (which is recommended).

Required

When the user chooses New from the File menu, and the contents of the window are to be replaced, prompt the user with a message to save the window contents if possible.

Recommended

Allow the user to set a property to determine whether New reuses the current window or opens a new window.

Selected Menu

Required

The New choice in the Selected menu creates a new object of the single type managed and displayed by the application, and includes it in the selection scope displayed within the window.

Required

When the user chooses New from the Selected menu, give the newly created object a default name that does not conflict with the name of an existing object and allow the user to change the name.

Optional

If your application allows the user to create multiple types of objects, allow New to be used in the Selected menu as a cascading choice that displays a menu of the types of objects that can be created.

Essential Related Topics

For more information, see the File Menu , Selected Menu , and Window Title reference pages.

Supplemental Related Topics

For more information, see the Menu Guidelines and Save/Save As (Action Choice) reference pages.

New Window (Choice)

NAME

New Window (Choice) –Reference

Description

New Window is an action or cascading choice that displays a view of the currently viewed data in a new window.

When to Use

Required

Provide a New Window choice in the View menu if your application allows a user to display a new window that contains another view of the data viewed in the current window.

Guidelines

Required

When New Window is an action choice, choosing it displays a new window that contains the same view of the data as the current window, or a default view.

Required

When New Window is a cascading choice, its cascaded menu must list the views available to be shown in the new window.

Recommended

Make New Window an action choice when a user is most likely to want the new window to show the view that will be chosen (for example, either the same view or a default view).

Recommended

Make New Window a cascading choice when a user is most likely to want the new window to show a different view than is being viewed in the current window.

Recommended

If you use a cascading choice for New Window, and you cannot show additional windows of some views, display choices for those views in the cascaded menu with unavailable emphasis.

Recommended

Use a property that the user can set to determine whether New Window uses the view in the current window or a default view.

Multiple Windows for Viewing Data

Required

Do not change the underlying data being viewed when the user opens a window that contains a new view.

Recommended

If you use multiple windows to display the same type of view of a file or object, and the user makes a File menu choice in one of the windows to change the file or object being viewed, only that window is affected; the other windows contain the original file or object.

Essential Related Topics

For more information, see the **View** and View Menu reference pages.

Supplemental Related Topics

For more information, see the Object reference page.

Notebook (Control)

NAME

Notebook (Control) –Reference

Description

A notebook is a multipage control for a linearly ordered set of pages that supports the use of notebook tabs as paging controls. It consists of the following elements:

Notebook tab

A kind of action choice used as a paging control within a notebook to make the page corresponding to the notebook tab visible.

Major tabs

A linear sequence of notebook tabs that identifies the important pages in a notebook or organizes the notebook into sections.

Minor tabs

A linear sequence of tabs that identifies important pages within the sections determined by the major tabs.

Bookmark tabs

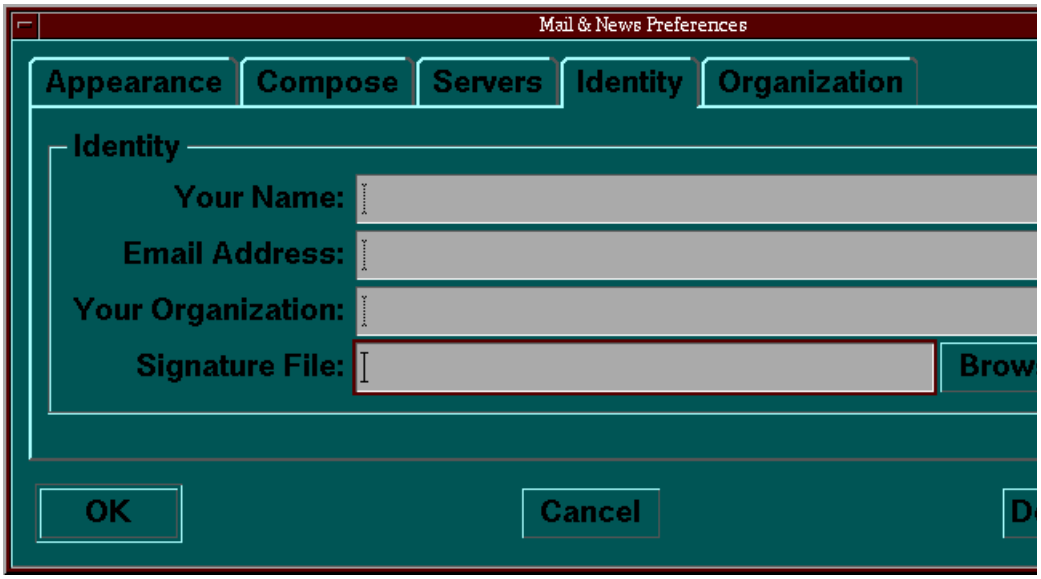
A set of tabs that corresponds to pages, dynamically identified as a result of user action.

Section

An associated contiguous group of pages in a notebook.

Notebook is an example of a simple notebook.

Figure 1 Notebook



When to Use

Recommended

Use a notebook when more than a small number of pages are to be used in a multipage control.

Recommended

Use a notebook when the pages in a multipage control are best organized in sections.

Guidelines

Notebook Layout

Required

Provide a spin box paging control for paging through the notebook.

Required

If there is insufficient space to display a group of notebook tabs, the layout is clipped and arrow buttons (action choices that cannot take focus) are displayed on the ends of the group to scroll the visible set of tabs.

Recommended

When the pages in a notebook can be organized into sections, provide a set of major tabs to directly access the first page of a section.

Recommended

When some pages within a section associated with a major tab have greater importance than other pages in the section, provide a set of minor tabs to directly access the important pages.

Recommended

Lay out the major tabs vertically along the right side of the notebook page and lay out the minor tabs horizontally below the notebook page in order of the corresponding pages. For information on bidirectional and vertical language support, see Chapter 11.

Optional

Provide one or more sets of bookmark tabs, additional paging controls, and other controls for displaying status in a notebook.

Behavior of Notebook Tabs

Required

The action associated with a notebook tab displays its corresponding page.

Required

Allow directional keys to navigate to all the notebook tabs in a group, including those that are not currently visible (navigating to one makes it visible).

Required

When the user uses the keyboard to activate a notebook tab, leave focus in the notebook tab.

Optional

When the user uses the mouse to activate a notebook tab, move focus to the page displayed.

Tab Groups

Required

Organize the major tabs in a notebook into a single tab group.

Required

Organize the minor tabs corresponding to pages within a section associated with a major tab into a single tab group.

Required

Organize any paging controls (other than notebook tabs) into one or more sets of tab groups.

Required

Organize any bookmark tabs into one or more sets of tab groups.

Tab Navigation Within Single−Page Notebooks

Required

Make the minor tabs for a section of a single−page notebook visible only when the currently visible page is within that section.

Required

When tab group navigation is used to traverse to a group of major or minor tabs in a single−page notebook, place focus on one of the following:

- The notebook tab for the currently visible page
- The nearest notebook tab preceding the currently visible page
- The first notebook tab in the group

Required

Traverse the tab groups within a single−page notebook in the following order:

- Tab group(s) of the visible page
- Any nontab paging controls
- Bookmark tabs
- Major tabs
- Minor tabs
- Traversable status controls

Essential Related Topics

For more information, see the Multipage Control (Control Type) reference page.

Object

NAME

Object -Reference

Description

An object is an element that visually represents something having behavior, contents, or both, not solely revealed by the visual representation. Objects are often represented as icons, but may be represented in other ways as well, for example, as list items. Objects generally have pop−up menus associated with them and can be used as the source of drag−and−drop operations.

When to Use

Recommended

Use objects as part of an object−oriented methodology to allow users to most naturally perform a task.

Guidelines

Required

In the pop−up menu associated with an object, provide access to the actions the user will need to perform with the task.

Required

Make the default action of an object be the action the user is most likely to want to perform with the task.

Required

Allow a user to change the default action of an object.

Required

Make a default action open a window that displays a view of the object (showing its contents), present an accessible user interface, or both. This is not required if the object is best presented in a nonvisual medium (for example, it contains audio and its default action is Play).

Required

Allow an object to be named or otherwise identified uniquely within a scope that contains multiple objects.

Required

If the default action opens a view on an object, and the view is already open, use properties that the user can set to determine whether to do one of the following:

Surface the view (recommended by default)
Open a new view on the object

Required

When the user opens a view of an object, do not remove the icon from which the view was opened.

Recommended

When the user opens a view of an object, display in−use emphasis on the icon for the object. If multiple icons identify the same object, provide in−use emphasis on all of them.

Recommended

If an object can be viewed in multiple ways, allow the user to specify a default view for each object.

Recommended

Allow the user to open multiple windows, each of which displays a different view of an object.

Recommended

Allow the user to open multiple windows that contain the same view of an object.

Recommended

When the user opens a view of an object, and the object is currently in use, use properties that the user can set to determine when changes made in one view are visible in another view.

Supplemental Related Topics

For more information, see the **Data Transfer** , Drag and Drop Transfer , Icon , Pop–Up Menu , and View reference pages.

Open (Choice)

NAME

Open (Choice) –Reference

Description

In the File menu, Open is a dialog choice that leads to a file selection dialog from which the user can select a file or object to open. In the Selected menu, Open is an action choice that opens the selected objects.

Guidelines

File Menu

Required

When the user selects the Open dialog choice from the File menu, either open a new window or replace the contents of the current window.

Required

When the user chooses Open from the File menu, and the contents of the window are to be replaced, prompt the user with a message to save the window contents if possible.

Recommended

Allow the user to set a property to determine whether Open reuses the current window or opens a new window by default.

Recommended

The File Selection dialog for the Open choice (from the File menu) should allow the user to indicate whether to replace the contents of the current window or to open a new window, overriding the default.

Selected Menu

Required

When the user chooses Open from the Selected menu, open each selected object in a separate window.

Essential Related Topics

For more information, see the File Menu and Selected Menu reference pages.

Supplemental Related Topics

For more information, see the Menu Guidelines and Window Menu reference pages.

Option Menu (Menu Type)

NAME

Option Menu (Menu Type) -Reference

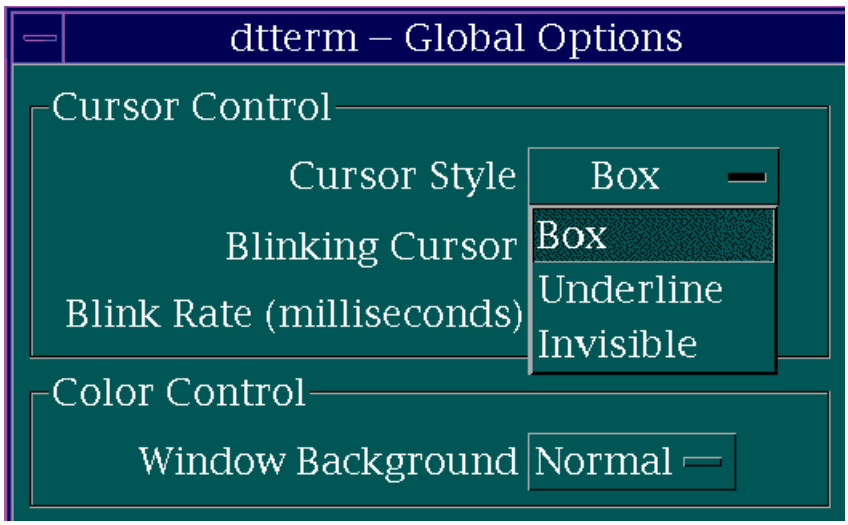
Description

An option menu is a menu that contains only value choices and cascading choices. It allows the user to choose a value from a group of values that may be hierarchically organized.

An option menu button is a button that contains a cascading choice for displaying an option menu and static text that displays the last value chosen from the corresponding option menu or its descendants.

Option Menu and Button illustrates an option menu and option menu button.

Figure 1 Option Menu and Button



When to Use

Required

Use an option menu to let the user choose a value from a fixed group of possible values.

Guidelines

Required

Place only cascading and value choices in an option menu.

Required

A cascading choice in an option menu leads to another option menu.

Required

All of the value choices in an option menu and its cascaded descendants are mutually exclusive; the user can choose only one at a time.

Keyboard Navigation

Required

When focus is on an option menu button and the user presses `Select`, `Space`, or `Ctrl Space`, display the associated option menu and move focus to it.

Optional

When focus is on an option menu button, and the user presses `[darr ]` or `Alt [darr ]`, display the associated option menu and move focus to it.

Side Effects

Recommended

If the user makes a value choice in an option menu that invokes an action as a side effect, limit the effect to changing views rather than changing the underlying data being viewed.

Default Choice

Required

If a value choice in an option menu corresponds to the value shown in its base option menu button, make it the default choice for that menu.

Recommended

If a cascading choice in an option menu leads to a menu or a descendant menu that contains the value shown in its base option menu button, make the cascading choice the default choice for the option menu it is in.

Size

Required

Make the width of the static text area of an option menu button the same as the width of the option menu displayed from it.

Placement

Recommended

When the user chooses an option menu button to display an option menu, place the option menu so that the default choice covers the last chosen value shown in the option menu button.

Required

When the user chooses an option menu button to display an option menu, the option menu must not cover the graphic that indicates a cascading choice in the option menu button.

Visual Guidelines

Required

Display the cascading choice in an option menu button as a horizontal bar graphic.

Required

Display the last value chosen in the option menu or its cascaded descendants in the option menu button as static text to the left of the bar graphic. For information on bidirectional and vertical language support, see Chapter 11.

Required

Do not place a graphic next to a value choice in an option menu (for example, to indicate that it is the value that was last chosen).

Essential Related Topics

For more information, see the Cascading (Choice Type) , Choice , Menu (Control) , and Value (Choice) reference pages.

Options (Menu)

NAME

Options (Menu) -Reference

Description

Options is a cascading choice that appears as a menu−bar item. It provides access to other menu items that enable a user to customize an application.

When to Use

Recommended

Provide an Options menu−bar item in each window that allows the user to tailor the appearance or behavior of the application.

Guidelines

Required

Provide choices in the Options menu that are specific to the application overall, not just to a particular view.

Required

Changes resulting from choices made in the Options menu do not change the underlying data being viewed.

Recommended

Provide choices in the Options menu that display dialogs through which the user can set default application options, such as whether New and Open opens a new window or replaces the contents of the current window.

Recommended

Provide choices in the Options menu (or in a dialog displayed via the Options menu) through which the user can set a level of expertise. Allow these settings to control the choices that appear in menus and the degree of explanation and assistance that the application provides.

Required

Do not use choices in the Options menu to set properties of objects or elements; use a Properties choice instead.

Essential Related Topics

For more information, see the Menu Guidelines and Object reference pages.

Supplemental Related Topics

For more information, see the Help (Menu/Action Choice) reference page.

Palette Area (Area)

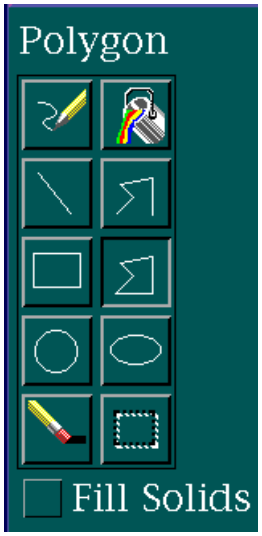
NAME

Palette Area (Area) –Reference

Description

A palette area is an area within a window that provides a place to store commonly used groups of controls. A palette area may also be called a tool area if it contains only tools.

Figure 1 Palette Area



When to Use

Recommended

Provide a palette area when the user may want to interact frequently with specific functional elements of the application. For example, provide a tool palette area when a user may need to use various tools while interacting with a graphical control.

Recommended

Place palette area choices that are critical to the user's task in a highly visible location near the viewing area or elements associated with the task.

Recommended

Provide a palette area to hold action choices when the user may issue one of a small number of commands.

Guidelines

Layout

Required

Lay out the palette area as a control bar (for example, place it just below the window's menu bar) if it can apply to all viewing areas in the window.

Required

If a palette area applies to only one viewing area in a window with multiple viewing areas, place the palette area adjacent to the associated viewing area.

Required

If a palette area is placed adjacent to its associated viewing area, place it on only one side of the viewing area. Use two sides only when there is a large number of critical action choices that must be placed in a logical relationship to an associated fixed element in the view.

Recommended

If a palette area is distributed over two sides of its associated viewing area, use only one side in each dimension.

Customizing the Palette Area

Required

If the user can customize a palette area or remove the action choices in a palette area, use a menu to provide access to the removable choices.

Recommended

If the user can customize a palette area and has removed all controls in it, remove the palette area.

Recommended

Allow the user to customize the palette area by allowing drag and drop.

Essential Related Topics

For more information, see the **Control** , Pointer , and Value Set reference pages.

Paned Box (Control)

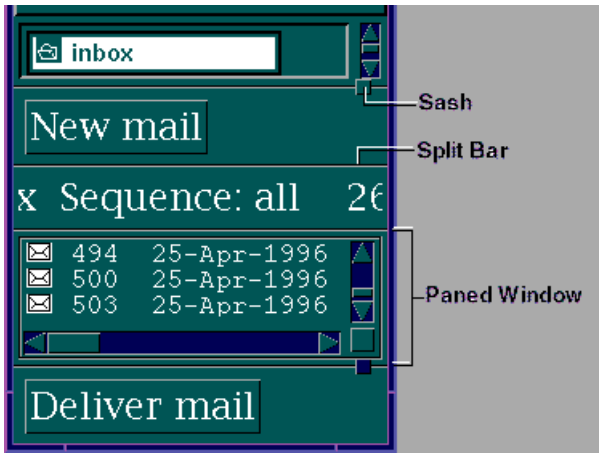
NAME

Paned Box (Control) –Reference

Description

A paned box (or paned window) is a control that is divided by split bars into areas called panes.

Figure 1 Paned Box



When to Use

Recommended

Provide a paned box when it is useful for a user to see related information or more than one view of data in separate panes.

Guidelines

Required

Support paned boxes that can be split either horizontally or vertically.

Required

Provide a split bar between panes of a paned box.

Required

If the paned box is used to separate multiple viewing areas in a window, make the window resizable.

Recommended

Support only one direction for a specific paned box.

Recommended

If your application supports changing the size of a pane, provide a sash on the split bar to do so.

Recommended

If your application supports multiple views of the information displayed in a pane, allow the user to split the pane multiple times in one or the other direction, but do not change the underlying data being viewed when the user does so.

Behavior

Recommended

Provide coordinated scrolling of related panes in a paned box in the dimension orthogonal to the direction of splitting. For example, allow a user to scroll vertically the first pane of a paned box that is split horizontally and have all the other panes scroll with the first pane.

Recommended

Allow panes that contain independent views to be scrolled independently.

Navigation

Required

Design each pane in a paned box to consist of one or more tab groups.

Required

Make each sash in a paned box a tab group.

Required

When the user uses tab−group navigation, traverse through the tab groups in a paned box in the following order:

All tab groups in the first pane
The sash on the split bar of the first pane, if any
All tab groups in the next pane
The sash on the next split bar

Repeat this pattern to the end of the paned box.

Essential Related Topics

For more information, see Chapter 5 and the **Sash (Control)** and Window Navigation reference pages.

Supplemental Related Topics

For more information, see the Control Navigation , Input Focus , Internal Navigation , and Tab Group reference pages.

Persistent Cue

NAME

Persistent Cue –Reference

Description

A persistent cue directs the user's attention to a part of the screen or user interface, indicates a particular state of an object, or alerts the user about potentially serious situations. There are three types of persistent cues:

- Graphical cues (augmentations of existing graphical elements)
- Information provided in window information areas
- Messages in secondary windows

In degraded environments (for instance, in a low–light environment or if the user is visually impaired) persistent cues can be supplemented, replaced, or augmented by cues in other sensory modalities, such as sound.

When to Use

Required

Provide a persistent cue to distinguish the state of some object represented by an element in the user interface or the status of some task that the user requested.

Required

Provide a persistent cue to alert the user about a potentially serious condition in the operating environment.

Required

Provide a persistent cue to attract the user's attention to some area of the interface.

Recommended

Provide a persistent cue to inform the user about features or behavior of the interface element with which the user is interacting.

Recommended

Provide a persistent cue when the user requests help with the application or an element of the interface.

Guidelines

Required

If the persistent cue is intended to alert the user, supplement it with a warning signal.

Required

If a persistent cue represents an internal state that is changing, update the cue as the change happens. If the change is a continuous change, update the cue continuously or at regular intervals.

Required

If a persistent cue blinks (for example, it turns on and off repeatedly) use it only if the element it is on has input focus or if the user must be made aware of a critical condition.

Required

Do not design a new persistent cue to represent a state that already has one defined for it.

Required

If a persistent cue is associated with a task, display the cue while the task is in progress and remove it when the task is done.

Recommended

If a task displays a persistent cue while it is in progress and the task takes a short time (between 2 and 10 seconds), use a special pointer as the persistent cue.

Recommended

If a task displays a persistent cue while it is in progress and the task takes an intermediate to long time (longer than 10 seconds), use a message as the persistent cue.

Recommended

If the element with a persistent cue associated with it is moved, move the persistent cue along with the element.

Recommended

If a persistent cue represents an internal state of the application or operating environment that can change independently, and if it changes such that the cue cannot be displayed, and your application has determined that such a change has occurred, remove the persistent cue.

Recommended

Supplement the persistent cue with a warning signal if the user might ignore the cue inadvertently.

Recommended

If your application can determine that a persistent cue is no longer needed, remove it.

Recommended

Do not change the behavior of an element of the interface just because a persistent cue has been associated with it.

Recommended

Do not design any element of the interface so that it will always have the same persistent cue associated with it. It should be possible for the application to show the element with or without the persistent cue, depending on some user–modifiable internal state.

Recommended

If a persistent cue modifies the appearance of the element it is on, design the cue so that it does not change the size of the element. If this is not feasible, design the element so that the size of the persistent cue is included in its size. If neither option is feasible, use a message instead of a graphical cue.

Essential Related Topics

For more information, see the **Information and Message Areas (Area)** , Message , and Warning Signal reference pages.

Point Technique

NAME

Point Technique –Reference

Description

The point technique is an individual selection technique in which a single point is identified. If the point is on a selectable element, that element is either selected or has its selection state toggled. The following describes the steps for mouse–based and keyboard–based techniques:

Mouse–based point technique

Click the SELECT button at the desired point or on the desired element.

If no mouse–based group technique is supported for the selection scope:
Press the SELECT button on a selectable element.
Release the SELECT button on the same element (with arbitrary movement of the mouse in between).

Mouse–based point technique, forcing toggle mode

Click `Ctrl` SELECT at the desired point, or on the desired element.

If no mouse–based group technique is supported for the selection scope:
Press `Ctrl` SELECT on a selectable element.
Release the SELECT button on the same element with arbitrary movement of the mouse in between).

Keyboard–based point technique

Press `Select`, `Space` (unless in text), or press `Ctrl Space` with the cursor at the desired point or on the desired element.

Guidelines

Mouse–Based Select Mode

Required

When the user is using the point technique in select mode and presses or clicks the SELECT button on a selectable element and then releases SELECT on the same element:
Select that element.
Deselect all other elements in the scope.

Required

When using the point technique in select mode and the user clicks the SELECT button in the background of a selection scope, deselect all elements in the scope.

Mouse–Based Toggle Mode

Required

When the user is using the point technique in select mode and presses `Ctrl` SELECT in place of SELECT, select as if toggle mode were in use.

Required

When the user is using the point technique in toggle mode and presses or clicks the SELECT button on a selectable element and then

releases SELECT on the same element, toggle the selection state of that element.

Required

Allow the point technique in toggle mode to select an element as follows:
If only one element at a time may be selected in a scope, deselect the currently selected element (if any) in the scope.
If any number of elements can be selected in the scope, do not change the selection state of the other elements.

Required

When the user is using the point technique in toggle mode and clicks the SELECT button in the background of a selection scope, it should have no effect on the selection state of the elements in the scope.

Keyboard−Based Normal Mode

Required

When the user is using the point technique in normal mode while the active cursor is on a selectable element and presses `Select`, `Space` (unless in text), or `Ctrl Space`:
Select that element.
Deselect all other elements in the scope.

Required

When the user is using the point technique in normal mode, if the active cursor is in the background of a selection scope and the user presses `Select`, `Space` (unless in text), or `Ctrl Space`, deselect all elements in the scope.

Keyboard−Based Add Mode

Required

When the user is using the point technique in add mode while the active cursor is on a selectable element and presses `Select`, `Space` (unless in text), or `Ctrl Space`, toggle the selection state of that element.

Required

Allow the point technique in add mode to select an element as follows:
If only one element at a time may be selected in a scope, deselect the currently selected element (if any) in the scope.
If selections are required to be contiguous, deselect all other elements in the scope.
If any number of elements can be selected in the scope, do not change the selection state of the other elements.

Required

When the user is using the point technique in add mode while the active cursor is in the background of a selection scope and presses `Select`, `Space` (unless in text), or `Ctrl Space`:
If selections are required to be contiguous, deselect all elements in the scope.
If selections may be discontiguous, do not change the selection state of the elements in the scope.

Selection Region

Required

When the user performs the point technique on a selectable element:
Identify that element as an anchor element for later use in adjustment of the selection.
Define the current selection region to consist of that element.

Required

When the user uses the point technique on a selectable element, identify a point as an anchor point for later use in adjustment of the selection. That point should be:
The pointer position, for the mouse−based technique
The cursor position, for the keyboard−based technique with a text or graphics cursor

The center of the element, for the keyboard–based technique with an element cursor

Required

When the user uses the point technique in the background of a selection scope:
Identify that point as an anchor point for later use in adjustment of the selection.
Define the current selection region to be empty.

Required

After a mouse–based point technique, place the active cursor as follows:
At the anchor point, if it can be placed there
On the anchor element, if using an element cursor
Where the cursor previously was in the scope, if there is no anchor element

Essential Related Topics

For more information, see the **Adjustment Techniques** , Selection Modes , Selection Policies , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Selection Models reference page.

Pointer

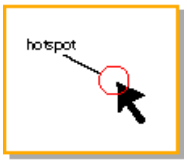
NAME

Pointer –Reference

Description

The pointer is a graphical cue, usually in the shape of an arrow, that a user can move with a pointing device. Users place the pointer over an element they want to work with. A position on the pointer, called the hot spot, tracks the movement of the input device and is the location where input device actions occur.

Figure 1 Pointer Hot Spot



When to Use

Required

Display a pointer whenever the operating environment allows the user to move it.

Recommended

Use different pointer shapes to provide a visual indication of the kind of interaction allowed. For example, use a spray can pointer to indicate that the user can "paint" in a drawing area.

Required

Use the hot spot to target (point to) the window, object, element, or selection under the pointer. If multiple elements are under the pointer, the topmost element is generally considered the element pointed to by the pointer.

Guidelines

Required

When no other element–specific or task–specific pointer is displayed, display the arrow pointer.

Required

Do not create new pointer shapes for operations that already have pointer shapes associated with them.

Required

If the pointer can be warped (that is, moved by the application without reference to any user control of the pointing device), allow the operating environment to disable this feature and prevent the pointer from being warped.

Definition of the Hot Spot

Required

Define a hot spot for each pointer shape.

Recommended

Place the hot spot at a visually distinguishable position of the pointer shape.

Recommended

If the pointer graphic has no visually distinguishable feature, place the hot spot at its center, if it is a symmetrical graphic.

Using the Keyboard to Move the Pointer

Required

For users unable to use a standard pointing device, support a mode whose only effect is to enable the directional keys, possibly modified with `Ctrl`, to act as a pointing device.

Required

Support access for users unable to press a pointer button or move the pointer simultaneously or for users who need to perform more precise placement than possible with a standard pointing device.

Required

If your interface enters a mode on a pointer button press and supports behavior on pointer motion in that mode, and there is no other efficient alternative means of providing the same effect, then do the following:

Provide a means of entering that mode via the keyboard, clicking a pointer button (possibly augmented by modifier keys), or both.

Support both the pointing device and the directional keys as a means of moving the pointer in that mode, with an unmodified directional key moving the pointer by one pixel in the direction indicated, and a directional key modified by `Ctrl` moving the pointer by a larger increment.

Do not change the effect of other keyboard actions. In particular, support `Enter` to complete the operation and to exit from the mode, `Cancel` to cancel the operation and to exit from the mode, and `Help` to obtain help.

Behavior

Recommended

When both the pointer and focus are on a control that is using a text or graphics cursor, hide the pointer until the mouse is moved after either of the following:

The user types a text character that is inserted.

The user uses keyboard navigation to move the cursor.

Recommended

If you provide a warning signal associated with a pointer shape, provide the warning signal only when the pointer changes to that shape. For instance, a drag pointer could change to a "dropping image" and generate a warning signal when it moves over the trash can icon. Do not, however, keep generating the warning signal while the pointer is held over the trash can.

Required

Use the hot spot position of the pointer to track the movement of the input device and the location where input device actions occur.

Required

As the pointer changes shape, indicating a change in the function of the current area, do not change the location of the hot spot on the screen.

Recommended

Do not warp the pointer when the user operates a pointing device.

Essential Related Topics

Pointer (Predefined)

For more information, see the **Pointer (Predefined)** reference page.

Supplemental Related Topics

For more information, see the Direct Manipulation , Keyboard (Device) , and Mouse (Device) reference pages.

Pointer (Predefined)

NAME

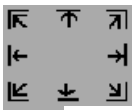
Pointer (Predefined) –Reference

Description

A predefined pointer is a set of standardized pointers that indicate model interactions or operations of the user interface.

Pointer Shapes describes predefined pointer shapes.

Table 1 Pointer Shapes

Appearance	Name	Hot Spot Location	Function
	Arrow	Tip of the arrow	Normal mode
	Cannot	Center	No input allowed
	Wait	Center	Action in progress
	I–beam	Baseline	Text position
	4–directional arrow	Center	Move/Resize
	Resize (8 pointers)	Tip of the arrow	Resize
	Sighting	Center	Precise positioning
	Question	Period	Help mode
	Right arrow	Tip of the arrow	Menu selection

When to Use

Required

Display the arrow pointer to indicate the normal or predominant mode of operation of the user interface.

Required

Display the cannot pointer to indicate that the element under the pointer is not a valid target for a data transfer.

Required

Display the cannot pointer to indicate that action is expected in another visible area before input is accepted in the area under the pointer.

Required

Display a wait pointer to indicate that the user cannot currently interact with the element the pointer is over because some action is in progress in that area.

Required

Display an arrow pointer or an I−beam pointer when the pointer is in a text field.

Required

Display a question pointer when the interface is in context−sensitive help mode.

Required

Display the right arrow pointer when a spring−loaded control is displayed and the interface is waiting for a choice to be made.

Recommended

Display an I−beam pointer when the pointer is being used to initiate a text field in a graphic selection scope.

Recommended

Display an I−beam pointer to indicate that the pointer is over an area where the user can locate a text cursor or select a character sequence.

Recommended

Display the 4−directional arrow pointer if a specific object is being moved without restrictions on its placement.

Recommended

Display the 4−directional arrow pointer after the user initiates a resize operation on a specific element, but before the edge or corner to be resized has been determined.

Recommended

Display one of the eight resize pointers when the pointer, in the predominant mode of operation, is on an element that controls resizing in that direction.

Recommended

Display one of the eight resize pointers when resizing in the indicated direction.

Recommended

Display an arrow or sighting pointer when the pointer is in a viewing area in which the user can specify a precise position for placing a visual element.

Recommended

Display the sighting pointer if the user is in a context in which a precise positioning operation can be initiated, and while the user is performing the operation.

Guidelines

Required

Ignore all mouse button and keyboard actions (except changes in modifier keys) while the cannot pointer or the wait pointer is visible.

Required

Use the predefined pointers when a user performs an operation that uses predefined modal interactions or operations.

Recommended

If a user cannot interact with an element because a current task would normally be affected by that element, display the wait pointer while the pointer is over that element. Do not prevent a user from interacting with elements that will not be affected by a current task.

Essential Related Topics

For more information, see the Cursor and Pointer reference pages.

Pop–Up Menu (Menu Type)

NAME

Pop–Up Menu (Menu Type) –Reference

Description

A pop–up menu is a menu used as a spring–loaded pop–up control that contains context–sensitive choices.

Figure 1 Pop–Up Menu

When to Use

Required

Associate a pop–up menu with every object.

Recommended

Associate a pop–up menu with elements in a selection scope that can be selected.

Recommended

Associate a pop–up menu with the background of a selection scope.

Recommended

Provide a pop–up menu for an element whenever users frequently need to apply actions to that element and if the element is accessible by the predefined shortcut keys.

Optional

A pop–up menu can be associated with controls or elements not associated with selection.

Guidelines

Inclusion

Required

If a choice is not currently available, do not include it in a pop–up menu (except if a window menu is used as a pop–up menu).

Required

If an action choice applies only to objects (for example, choices that also appear on the Selected menu), make the choice available on a pop –up menu.

Recommended

Include an action on a pop–up menu when users frequently need to apply that action to the associated element or selection.

Recommended

The availability and contents of the pop–up menu should depend on the location of the pointer within an element, as well as the current state.

Required

If a choice is not currently available, do not include it in a pop–up menu (except for those used as window menus).

Display

Required

Treat a menu popped up on an unselected character in text as if it were popped up in the background.

Required

If the user attempts to display a pop–up menu where no menu exists, or in a context in which the menu would be empty, do not display any pop–up menu; initiate a warning signal instead.

Required

Keep the relative order of identical choices the same on a pop–up menu as on the pull–down menu. For example, because Paste appears after Copy on the Edit menu, it should appear after Copy on any corresponding pop–up menu.

Required

Do not display a shortcut key for a choice on a pop–up menu unless it has the same result as the corresponding choice from a pull–down menu.

Required

If you include the Help choice on a pop–up menu, make it the last choice on the menu.

Required

On every pop–up menu, provide a title that indicates the function the menu performs or the identity of the element or object associated with it.

Required

Describe which elements have a pop–up menu associated with them and how these pop–up menus can be used in the Help choice available through the window's Help pull–down menu.

Optional

If an unselected element in a selection scope is not an object, and is not displayed as a solid shape, then your application can treat a menu popped up on the element as if it were popped up in the background.

Choice Augmentation

Required

If you include the Delete or Clear choice in a pop–up menu, augment them with the words "to Trash" if and only if all elements to which the operation applies are objects.

Recommended

You can include action choices from pull–down menus other than Edit or Selected in a pop–up menu if there is no possible confusion about semantics. You can use appropriate augmentation of the label if necessary for clarity.

Optional

You can augment action choices in pop–up menus (or menus cascaded from them) with the word "All" to apply the action to all selectable elements in the selection scope.

Optional

You can augment action choices in pop–up menus (or menus cascaded from them) with the word "Selected" or "Selection" (for example, Cut Selection) to apply the action to the current selection in the selection scope.

Pop–Up Menus for Selections and Selectable Elements

Required

In conjunction with the **Menu Guidelines** reference page, use Pop–Up Menu Choices for Selections and Selectable Elements to decide which choices to include in a pop–up menu associated with a selection or a selectable element and how to organize them.

Table 1 Pop–Up Menu Choices for Selections and Selectable Elements

	Mnemonic	Menu Choice
Required	S	Properties
Required	O	Open
Optional	U	Undo
Optional	R	Repeat/Redo
Required		Drag/Move
Optional		Size
Optional		Duplicate
Required		Copy To ...
Required		Move To ...
Optional	T	Cut
Optional	C	Copy
Optional	K	Copy Link
Optional		Copy Special
Optional	P	Paste
Optional	L	Paste Link
Optional		Paste Special
Optional	I	Insert ...
Optional		Primary Copy
Optional		Primary Move
Optional		Primary Link
Recommended	E	Clear {to Trash}
Required	D	Delete {to Trash}
Optional	S	Select All
Optional		Deselect All
Optional		Select Pasted
Optional		Reselect
Optional		Print ...
Recommended		Help

Required

Make Undo or Repeat/Redo available from a pop–up menu (or a menu cascaded from it) associated with a selectable element only if the corresponding action applies to the selection or the selectable element on which the menu was popped up.

Recommended

Include predefined pull–down menu choices on a pop–up menu associated with a selection or a selectable element only if they also appear

on the Selected or Edit menus or clearly apply to the selection.

Pop–Up Menus for Selection Scope Backgrounds

Required

In conjunction with the Menu Guidelines reference page, use Pop–Up Menu Choices for Selection Scope Backgrounds to decide which choices to include in a pop–up menu associated with the background of a selectable scope and how to organize them.

Table 2 Pop–Up Menu Choices for Selection Scope Backgrounds

Menu Choice	Mnemonic	
Optional	U	Undo
Optional	R	Repeat/Redo
Required	N	New
Optional		Duplicate
Optional		Copy To ...
Optional		Move To ...
Optional	T	Cut
Optional	C	Copy
Optional	K	Copy Link
Optional		Copy Special
Recommended	P	Paste
Optional	L	Paste Link
Optional		Paste Special
Optional	I	Insert ...
Optional		Primary Copy
Optional		Primary Move
Optional		Primary Link
Recommended	E	Clear {to Trash}
Required	D	Delete {to Trash}
Optional	S	Select All
Optional		Deselect All
Optional		Select Pasted
Optional		Reselect
Optional		Print All ...
Optional		Print Selected ...
Recommended		Help

Required

Include the New choice in a pop–up menu only if it is associated with the background of a selection scope and if New appears in the corresponding Selected menu.

Required

Include action choices that act on a selection in a pop–up menu associated with the background only if there are elements in the selection scope that are selected.

Recommended

If an action choice in a pop–up menu associated with the background of a selection scope applies to the current selection, augment the choice with the word "Selected" or "Selection." For example, use Cut Selection instead of Cut.

Semantics of Pop–Up Menu Actions for Selection Scopes

Required

If the pop–up menu is associated with a selectable element that is not currently selected and the action label is not augmented with the word "Selected" or "Selection," make the action act as if only the associated element were selected. For example, when the user selects Cut from a pop–up menu on an unselected element, cut just that element to the clipboard.

Required

Help on a pop–up menu associated with an element should have the same effect as pressing `Help` or `F1` when the active cursor is on an element.

Pop–Up Menus for Other Elements and Controls

Required

In conjunction with the Menu Guidelines reference page, use Pop–Up Menu Choices for Other Elements and Controls to decide which choices to include in a pop–up menu associated with controls or elements that are not selectable or that do not include selection scopes and how to organize them.

Table 3 Pop–Up Menu Choices for Other Elements and Controls

Menu Choice	Mnemonic	
Optional	C	Copy
Optional		Copy Special
Optional	P	Paste
Optional		Paste Special
Recommended		Help

Optional

In pop–up menus on elements not related to selection scopes, you can use Copy or Copy Special to place information or values displayed in the element on the clipboard.

Optional

In pop–up menus on elements not related to selection scopes, you can use Paste or Paste Special to set information or values.

Required

Do not change the data or selection state of an element when the pop–up menu associated with it is displayed.

Required

If you provide an action choice (other than Help) on a pop–up menu (or a menu cascaded from it) that is associated with a selection scope, provide access to the same choice from either a menu bar, menu cascade button, push button in the same window, or a shortcut key.

Recommended

When the user invokes a pop–up menu, display source emphasis on the associated element or control.

Mouse Display and Deactivation

Required

When the user presses (not clicks) the MENU button at a location that allows a pop–up menu, display the associated pop–up menu at that location.

Required

When the user clicks the MENU button at a location that allows a pop–up menu and the menu is not currently displayed, display the associated pop–up menu at that location.

Required

When the user clicks the MENU button at a location that allows a pop–up menu and the menu is currently displayed, deactivate the associated pop–up menu at that location.

Keyboard Display and Deactivation

Required

If a pop–up menu is associated with the location of the active cursor and the user presses `Menu` or `Shift F10`, display the associated pop–up menu at that location.

Required

If focus is in a spring–loaded system based on a pop–up menu and the user presses `Menu` or `Shift F10`, deactivate the system.

Placement

Recommended

When a user presses the MENU button to display a pop−up menu, place the menu so that the menu's default choice is to the right of the pointer. For information on bidirectional and vertical language support, see Chapter 11.

Essential Related Topics

For more information, see the Data Transfer , Edit (Menu) , Menu (Control) , Menu Guidelines , Object , Selected Menu , Selection , and Spring−Loaded (Control Type) reference pages.

Supplemental Related Topics

For more information, see the Emphasis (Cue) , Help (Menu/Action Choice) , Shortcut Key , and Warning Signal reference pages.

Prefix Completion

NAME

Prefix Completion –Reference

Description

Prefix completion is an internal navigation technique used in a combination text–list control in which entering a printing character immediately moves the cursor in a text–display field to the next item whose textual label begins with the prefix.

When to Use

Recommended

Use prefix completion in combination text–list controls for index and topic searching.

Guidelines

Behavior

Required

When the user enters a printing character into a text–display field with prefix completion, move the cursor in the associated list to the next item that matches the prefix. Then, select the item and place its text contents into the text–display field. Leave the text cursor in place after the character that was just typed.

Required

When the user enters a printing character into a text–display field with prefix completion, and there is no item matching the prefix, generate a warning signal. Do not make any changes to the appearance of the list or of the text–display field.

Required

When the user enters a printing character in a text–display field with prefix completion, and the associated list is not displayed, do one of the following:

Display the list if `[darr ]` also displays the list.

Navigate in the list as indicated and select the appropriate list item, displaying it in the text field without displaying the list.

Recommended

When focus is in a text–entry field that uses prefix completion, allow `Ctrl [darr ]` and `Ctrl [uarr ]` to move the cursor and select the next or previous item that matches the prefix. Place its text contents into the text–display field, but leave the text cursor in place.

Recommended

Use browse selection in the associated list when prefix completion is in use.

Essential Related Topics

For more information, see the **Combination Text–List Control (Control Type)** , List Box (Control) , Prefix Navigation , Text–Display Field (Control) , and Text–Entry Field (Control) reference pages.

Prefix Navigation

NAME

Prefix Navigation –Reference

Description

Prefix navigation is an internal navigation technique used in a combination text–list control in which entering a printing character immediately moves the cursor in the associated list to the next item whose textual label begins with a prefix.

When to Use

Use prefix completion in combination text–list controls for index and topic searching.

Guidelines

Behavior

Required

When the user types a character in the text–entry field, move the cursor in the associated list to the next item whose label matches the prefix, enter the character into the text–entry field, and place the text cursor after the character. Keep the input focus in the text–entry field.

Required

When the user types a character in the text–entry field and there is no item matching the prefix in the list, generate a warning signal.

Required

If focus is in a text–entry field that uses prefix navigation, and the associated list box is not displayed, display the list box when the user types a printing character.

Recommended

By default, when focus is in the text field, select the cursored list item and place its text in the text field.

Required

When prefix navigation is used to move the cursor in a list, do not automatically select the cursored item unless required to do so by the selection model in use.

Recommended

When focus is in a text–entry field that uses prefix navigation, allow `Ctrl [darr ]` and `Ctrl [uarr ]` to move the cursor and select the next or previous item that matches the prefix. Do not place the matching text in the text field.

Essential Related Topics

For more information, see the **Combination Text–List Control (Control Type)** , List Box (Control) , Prefix Completion , Text–Display Field (Control) , and Text–Entry Field (Control) reference pages.

Primary Transfer

NAME

Primary Transfer -Reference

Description

Primary transfer is a transfer mechanism by which the primary selection is transferred to a destination. The primary transfer operations are primary copy, primary link, and primary move. The following list describes these operations:

Primary Copy

An action choice that copies the contents of the primary selection to a specified location or to an object or objects displayed within the window.

Primary Link

An action choice that places a link to the elements contained in the primary selection in a specified location or in an object or objects displayed within the window.

Primary Move

An action choice that moves the contents of the primary selection to a specified location or to an object or objects displayed within the window.

When to Use

Required

If a scope uses primary selection, allow the selection to be used as the source of a primary transfer.

Required

If an element can be used as a target of a data transfer operation, support primary transfer to it.

Guidelines

Required

The source of a primary transfer is the primary selection.

Required

The target of a primary transfer is the control for which the primary transfer was invoked.

Required

The default operation for a TRANSFER button–based primary transfer is a copy, though it may be overridden with the `Ctrl` and/or `Shift` modifiers.

TRANSFER Button–Based Initiation

Required

When TRANSFER is bound to MB2, allow the user to click the TRANSFER button, optionally augmented with `Shift` and/or `Ctrl`, for primary transfers.

Required

If the user presses the TRANSFER button within a primary selection that can be dragged, initiate the drag.

Required

If TRANSFER is a separate mouse button from SELECT and the user presses the TRANSFER button outside of a primary selection, but at a position where both a drag can be initiated and data can be pasted by a primary transfer, then use the user−specified timeout and motion threshold in the following ways to distinguish drag initiation from primary transfer:

After the user presses the TRANSFER button, if the motion of the pointer exceeds the motion thresholds, initiate a drag.
If the user presses and releases the TRANSFER button within the timeout period, perform a primary transfer.
When the timeout period expires, initiate a drag. But, if the user releases the TRANSFER button before exceeding the motion threshold, cancel the drag.

Behavior

Required

Select the data transferred to a target scope as a result of a primary move operation if the target is using normal mode.

Required

A primary transfer must move the focus to the target control.

Recommended

Duplicate the selected data when the user invokes a primary copy in a linearly organized scope by one of the following:

Using the TRANSFER button with the pointer at the edge of the primary selection
Choosing the operation from a menu popped up at the edge of a primary selection
Choosing the operation from the Edit menu, or pressing the shortcut key for it when the cursor is at the edge of the primary selection

Essential Related Topics

For more information, see the **Data Transfer** and Direct Manipulation reference pages.

Supplemental Related Topics

For more information, see the Mouse (Device) and Pop−Up Menu (Menu Type) reference pages.

Primary Window

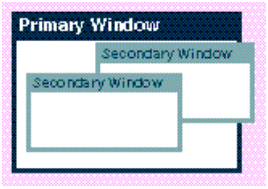
NAME

Primary Window –Reference

Description

A primary window is a window in which the main interaction between the user and an object or application takes place.

Figure 1 Primary Window



When to Use

Required

Use a primary window as the main window of an application.

Required

Use a primary window to display a view of an object.

Guidelines

Required

When a primary window is closed or minimized, remove its secondary windows.

Required

When a primary window is closed, close that window and all of its unshared secondary windows.

Required

When a primary window that has associated secondary windows is opened or restored, restore the secondary windows that apply to the current conditions.

Required

When a primary window is closed, remove its window icon.

Required

When the last primary window of an application is closed, close the application.

Required

When a primary window that is critically needed to work with an application is closed, close the application, even if there are other displayed primary windows of that application.

Recommended

When a primary window is opened or restored, place the windows in the window family in the same position that they were in before the window family was closed or minimized.

Recommended

Provide a sizing border, a title, minimize and maximize buttons, a menu bar, and a window menu in a primary window, unless these have been disabled by the user through an environmental specification, or if the primary window is a component of the workspace.

Essential Related Topics

For more information, see the **Window Icon** , Window Icon Box , and Secondary Window reference pages.

Properties (Choice)

NAME

Properties (Choice) –Reference

Description

Properties is an action choice that displays a property dialog in which the user can display and set properties or characteristics of a file object or other element.

When to use

Required

Use a Properties choice to allow the user to change the properties of a file or object.

Guidelines

Required

Include a Properties choice in the File menu when the user can change the attributes of the selected file, for example, to change a read–only file to read–write.

Required

Use a multipage control to display properties if they cannot all be displayed at once.

Required

If multiple elements are selected, and the user chooses Properties, then for a given property, do the following:

Provide a description that encompasses all selected elements. For example, properties that can take only two values can be displayed as a check box that may either be set, unset, or indeterminate.

Provide a separate page for each element, with each page showing the value for that property.

Recommended

When a user opens the properties of objects selected with a primary selection, allow selections to be made in the properties window. When the user closes the properties window, return the primary selection to the original selected objects.

Optional

When a parent container (or folder) displays an element that represents an object, the properties of the object can be represented by the following:

Aspects of the state or contents of the specified object

Container–specific attributes of the object that are part of the container's state (for example, how the container is to manage the object or objects)

View–specific presentation attributes of the object (for example, the icon for the object when viewed by the user, or when viewed by the user in this container)

Optional

When the selected element represents an object, properties may contain presentation attributes, which include the following:

Whether the object should be represented as an icon or as a live or dead inset

If represented as an icon, the representation, size, color, and so on

For an object represented as an icon or dead inset, the default action (for example, the result of double–clicking on it)

For an icon that represents a link: the link properties, whether the default action should be traversal and, if so, whether to reuse an existing view of the linked data

If the link is to a portion of the object, whether to display only the linked data or the linked data selected in the context of the

containing object

Essential Related Topics

For more information, see the **Selected Menu** reference page.

Supplemental Related Topics

For more information, see the Menu Guidelines reference page.

Pull–Down Menu (Menu Type)

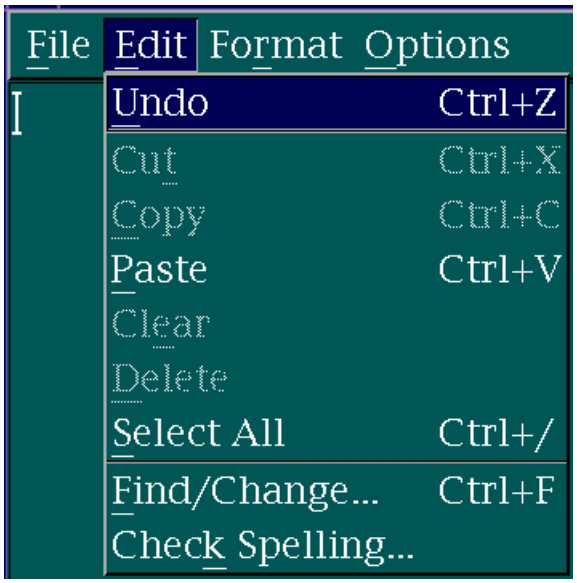
NAME

Pull–Down Menu (Menu Type) –Reference

Description

A pull–down menu is a menu displayed from a cascading choice in a menu bar or a menu cascade button.

Figure 1 Pull–Down Menu



When to Use

Recommended

Provide a pull–down menu for each choice on a menu bar, for the window menu button, and for each menu cascade button in your application.

Guidelines

Recommended

If you provide choices that do not logically fit within any of the predefined pull–down menus, provide your own application–specific menu –bar choices that lead to menus in which you can include those choices.

Recommended

Provide a unique shortcut key assignment for each frequently used application–specific choice in a pull–down menu or a menu cascaded from one.

Optional

You can augment action choices in pull−down menus (or menus cascaded from them) that refer to the current selection in a selection scope with the word "All" to apply the action to all selectable elements in the scope.

Essential Related Topics

For more information, see the **Cascading (Choice Type)** , Menu Bar (Menu Type) , and Menu Cascade Button (Control) reference pages.

Push Button (Control)

NAME

Push Button (Control) –Reference

Description

A push button is a control that represents an action or dialog choice. Push Buttons shows three push buttons.

Figure 1 Push Buttons

When to Use

Required

If you do not provide a menu bar in a window with a viewing area, place all action, cascading, or dialog choices on push buttons in that window, except for those choices that appear on the window menu.

Recommended

Use a push button in a window with a menu bar to provide convenient access to a frequently used action choice, dialog choice, or cascading choice.

Guidelines

Behavior

Recommended

For push buttons that provide action choices that can be used repeatedly, such as the arrow buttons on the scroll bar, do the following:

Repeat the action if the user presses and holds the SELECT button for an initial period of time (the initial delay time specified by the operating environment). If the user continues to press and hold SELECT, continue to repeat the action whenever SELECT is held down for a period of time (the repeat delay time specified by the operating environment).

Represent a push button with ready emphasis while it is active.

Stop repeating the action when the user moves the pointer away from the push button; resume repeating the action if the user moves the pointer back over the push button without releasing SELECT.

If the user moves the pointer from one button of an adjacent pair of arrow buttons that perform opposite actions to the other button, switch the ready emphasis and switch actions.

Recommended

When a push button is persistently unavailable (because of the current configuration of the application or system or, for example, a particular set of companion software is not currently installed), remove the push button rather than showing it as unavailable.

Recommended

If a push button adjusts (or is associated with) a control within a window (rather than the entire window), do not close the window when a user activates the push button.

Layout

Recommended

When a window contains push buttons and a menu bar, place choices in the pull−down menus that provide functions equivalent to push −button functions.

Recommended

Allow a user to hide or remove push buttons if the choices are available through another mechanism, such as the menu bar.

Recommended

Combine existing menu choices to create new push−button choices for frequently used combinations of choices. For example, the Save and the New choices could be combined into a new push−button choice called Save and New that would perform the Save action followed by the New action.

Recommended

If the action indicated by the label on a push button adjusts or is associated with a control within the same window as the push button, place the push button near that control. For example, if the function of a push button is to restore the initial value in a text−entry field, place that push button beside the text−entry field that it affects.

Recommended

Avoid using a push button to change the size of a window; instead, allow the user to change the size of the window with window menu choices and elements of the window frame.

Scrolling

Recommended

If the viewing area of a window can be scrolled and the window contains push buttons that affect the entire window, keep the push buttons visible when the viewing area is scrolled.

Recommended

If a window contains a push button that affects a control in a viewing area, scroll the push button along with the control when the user scrolls the viewing area.

Essential Related Topics

For more information, see the Choice , Control , Default Action , and Push Button (Predefined) reference pages.

Supplemental Related Topics

For more information, see the Dialog (Choice Type) , Emphasis (Cue) , and Size Border (Control) reference pages.

Push Button (Predefined)

NAME

Push Button (Predefined) –Reference

Description

Push buttons can provide predefined functions. The predefined push buttons are Apply, Cancel, Close, Continue, Help, More, No, OK, Pause, Reset, Reset To Default, Resume, Retry, Save As Default, Stop, Undo, and Yes.

When to Use

Required

Use a push button with the predefined label when you provide the function defined as follows:

Apply

An action choice that appears in a window and makes the changes indicated in the window without closing it.

Cancel

An action choice that removes a secondary window without applying any changes made in that window.

Close

An action choice that removes a window and all of the windows associated with it from a screen.

Continue

An action choice that resumes a task that has been interrupted by the application operating environment when the user can proceed as originally requested.

Help

An action choice used on push buttons in secondary windows to provide help specific to that window.

A cascading choice that appears as a menu–bar item and provides access to other menu items that contain information related to the use of the application.

More

An action choice that displays additional controls in a separate window.

An action choice that expands the current dialog to show more value choices and replaces itself with the current dialog.

No

An action choice that indicates a negative response to a question presented in a message.

OK

An action choice that accepts the information in a window and closes it. If the window contains changed information, those changes are applied before the window is closed.

Pause

An action choice that temporarily suspends a task without ending it.

RESET

An action choice that resets the values displayed in a dialog or property window to the values they had when the window became displayed or when the values were last saved as defaults, whichever is most recent.

Reset To Default

An action choice that resets the values displayed in a dialog or property window to the values they had when default values were last saved.

Resume

An action choice that resumes a task that the user paused.

Retry

An action choice that attempts to complete an interrupted task.

Save As Default

An action choice that saves the values displayed in a dialog or property window as defaults to be used when the same (or similar) window is subsequently displayed to the same user by the application.

Stop

An action choice that ends a task and removes the message window.

Undo

An action choice that undoes the effect of the last Apply.

Yes

An action choice that indicates a positive response to a question presented in a message by an application or the system.

Guidelines

Required

If you provide the Cancel and Help push buttons, place them to the right of all other push buttons. For information on bidirectional and vertical language support, see Chapter 11.

Required

Do not use both a Close push button and a Cancel push button in the same window.

Required

Provide a Resume push button whenever you provide a Pause push button.

Required

When the user makes changes in a dialog or property window and then activates Reset before committing them, return the application to the state that it was in before the user made the changes. Do not reset changes that have been previously committed, for example when the user has selected Apply or OK.

Required

When the user selects Reset, restore the settings of values only in the window where Reset was activated.

Required

Provide a Reset push button whenever you provide an Apply or OK push button.

Recommended

When Apply and Undo are both provided as push buttons in a window, unless the Undo label is augmented to indicate otherwise, make

Undo undo the effects of the last Apply. If you support only a single−level Undo, then after the user selects Undo, restore the changes undone if the user selects Undo again.

Recommended

Provide an Undo push button whenever you provide an Apply push button.

Recommended

Provide a Save As Default push button if you expect the user to save the values just set.

Recommended

When the user selects the Help push button, provide access to help information that directly relates to the user's current situation. For example, when the user encounters an error message, provide easy access to information that will assist the user in solving the problem that caused the error message.

Recommended

When the user selects the Help push button, display help information in a separate primary window.

Essential Related Topics

For more information, see the **Push Button (Control)** reference page.

Supplemental Related Topics

For more information, see the Action Message , Close (Choice) , Default Action , Help Menu , and Primary Window reference pages.

Quick Transfer

NAME

Quick Transfer –Reference

Description

Quick transfer is a data transfer technique used to copy, cut, link, or move data to the insertion point of the control that has interacted emphasis (the editable control the user was most recently interacting with). There are three quick transfer operations: quick copy, quick cut (or quick move), and quick link.

When to Use

Required

Allow users to use quick transfer when they want to copy, link, or move a temporary (or secondary) selection.

Required

Design text controls to support quick transfer as both a source and target.

Recommended

If a control can be used as the target for a data transfer operation, allow it to be used as the target for a quick transfer operation.

Recommended

Allow any selection scope to be used as the source of a quick transfer.

Optional

Allow quick transfer to transfer text and graphics that ordinarily cannot be selected. For example, the user cannot ordinarily select a portion of a text label of a push button, since clicking on the button with the SELECT button or pressing the `Select` key or `Space` activates the push button. You can allow quick transfer, however, to transfer a region of a label or graphic.

Guidelines

Required

Make the target of a quick transfer the control that has interacted emphasis.

Required

Make the source of a quick transfer the elements identified between pressing and releasing the TRANSFER (or SELECT) button.

Required

The default operation for a quick transfer should be copy, though it may be overridden using `Ctrl` and/or `Shift` modifiers when the user releases the TRANSFER button.

TRANSFER Button–Based Initiation

Required

Allow the user to press `Alt` TRANSFER and move the pointer (optionally augmented with `Shift` and/or `Ctrl`) for quick transfer only when TRANSFER is bound to MB2.

Behavior

Required

Do not select the data transferred to a target as the result of a quick cut operation.

Required

If the user moves the pointer out of a control that allows scrolling while `Alt` TRANSFER is pressed, use autoscrolling to scroll the control in the direction of the pointer. If TRANSFER is released with the pointer outside of the control, or if `Cancel` (or `Esc`) is pressed while TRANSFER is pressed, then remove the emphasis, do not perform the transfer, and restore the view to the state it was in before scrolling started.

Recommended

Show emphasis on source elements of a quick transfer as they are being identified. The type of emphasis should be different from that used for a selection. For example, in text, use underlining as the emphasis.

Quick Copy

Required

Allow either of the following methods to copy the secondary selection to the insertion point:
Pressing `Alt` TRANSFER and moving the mouse
Pressing `CtrlAlt` TRANSFER and moving the mouse

Quick Cut

Required

Move the contents of the secondary selection to the insertion point in the target control when the user presses `Shift Alt` TRANSFER, moves the mouse, then releases the TRANSFER button.

Quick Link

Required

Place a link to the secondary selection at the insertion point in the target control when the user presses `CtrlAlt Shift` TRANSFER, moves the mouse to select elements, and then releases the TRANSFER button.

Essential Related Topics

For more information, see the **Data Transfer** and Primary Transfer reference pages.

Supplemental Related Topics

For more information, see the Internal Navigation reference page.

Radio Button (Control)

NAME

Radio Button (Control) -Reference

Description

A radio button and radio button box are defined as follows:

Radio button

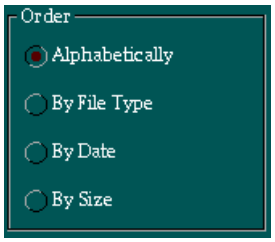
Represents a value choice that can either be on or off and that is part of a mutually exclusive group of radio buttons.

Radio box

A group of mutually exclusive radio buttons that are grouped together.

Radio Box and Buttons shows an example of a radio box and radio buttons.

Figure 1 Radio Box and Buttons



When to Use

Recommended

Use a radio box to display mutually exclusive value choices that have textual labels.

Guidelines

Required

The radio buttons in a radio box are mutually exclusive. When the user sets one radio button, unset any other radio button in the radio box.

Recommended

If a value choice can only be set or unset, and is not part of a larger set of choices that are not mutually exclusive, use a check box instead of two radio buttons.

Default Radio Buttons

Required

If the values represented by the radio buttons in a radio box cover all possibilities, do not allow the user to unset the radio button that is set. Attempting to do so should have no effect.

Required

If the values represented by the radio buttons in a radio box do not cover all possibilities, then allow the user to unset the radio button that is set.

Recommended

Ensure that the radio buttons in a radio box cover all possibilities. If necessary, add a radio button labeled None (or the equivalent) that covers any remaining possibilities.

Selection Properties

Required

When a radio box represents a value shared by multiple selected elements:
Display the radio buttons as selected if that value applies to all of the selected elements.
Do not display any of the radio buttons as selected if not all of the selected elements have a matching value.

Required

When a radio box represents a property shared by more than one selected element, ensure that the property of all selected elements corresponds to the radio button that the user sets.

Recommended

When a radio box represents a property shared by more than one selected element, and the user must be able to restore the properties of each of the selected elements after setting one of the radio buttons, then add another radio button that represents the initial state (such as No Changes). Display this button as set when no other value in the radio box applies to all the selected elements. When the user sets this radio button, restore the properties for each selected element to the value it had when this radio button was last set.

Layout

Required

Design a radio button with a graphic that indicates the state of the button and a text label that describes the state it controls.

Required

Use at least two radio buttons in a radio box.

Recommended

Arrange radio buttons in a group of rows, columns, or both.

Visuals

Recommended

Use a filled circle, diamond, or square as the graphic for the radio button.

Recommended

If you support color, use a color different from the other application colors to create a visually distinct on state. If you do not support color, represent the on state as either white or black, depending on the background being used.

Essential Related Topics

For more information, see the Control reference page.

Supplemental Related Topics

For more information, see the Check Box (Control), Choice, Label, and Value Set (Control) reference pages.

Range Adjust Click Technique

NAME

Range Adjust Click Technique -Reference

Description

The range adjust click technique is an adjustment selection technique in which a range of elements already identified as the current selection region can be adjusted by identifying an adjustment point. The following describes the steps for mouse−based and keyboard−based techniques:

Mouse−based range adjust click technique

Click the ADJUST button at the adjustment point.

Keyboard−based range adjust click technique

With the cursor at the adjustment point, press `Shift Select`, `Shift Space` (unless in text), or `Ctrl Shift Space`.

Guidelines

Required

When the user is using the range adjust click technique, and the adjustment policy is to reselect:
Determine the new selection region to be the range identified by the new anchor point and the adjustment point.
Enlarge the new selection region to include the anchor region, if indicated by the anchor inclusion policy.

Required

When the user is using the range adjust click technique, and the adjustment policy is to enlarge, make the new selection region the larger of the following:
The current selection region
The region determined by the reselect adjustment policy

Required

When the user is using the range adjust click technique, and the adjustment policy is to balance:
Move (if necessary) the anchor point so that it is at the end of the current selection region farthest from the adjustment point.
Determine the new selection region to be the range identified by the new anchor point and the adjustment point.
Enlarge the new selection region to include the anchor region, if the anchor point is unchanged and if indicated by the anchor inclusion policy.

Required

If the current selection region did not include an anchor element or if the anchor element is no longer in the selection region, identify the anchor element to be the element, if any, at the end of the range nearest the anchor point.

Required

After the user uses a mouse−based range adjust click technique, place the active cursor at one of the following:
If it is a text cursor, at the point at which the ADJUST button is released
If it is an element cursor, on the element at that end of the range

Essential Related Topics

For more information, see the Adjustment Techniques , Selection Policies , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Area Adjust Click Technique , Range Adjust Swipe Technique , Range Click Technique , Range Swipe Technique , Selection Models , Selection Modes , and Touch Adjust Click Technique reference pages.

Range Adjust Swipe Technique

NAME

Range Adjust Swipe Technique –Reference

Description

The range adjust swipe technique is an adjustment selection technique in which a range of elements already identified as the current selection region can be adjusted by identifying initial and final adjustment points in a single continuous action. The following describes the steps for mouse −based and keyboard−based techniques:

Mouse−based range adjust swipe technique

Press the ADJUST button at the initial adjustment point.
Move the mouse to the final adjustment point.
Release the ADJUST button at that point.

Keyboard−based range adjust click technique

With the cursor at the initial adjustment point, hold `Shift` down while pressing a navigation key to move the cursor to the final adjustment point.

When to Use

Required

If you are supporting discontiguous selections, support only the keyboard−based range adjust swipe technique when the initial adjustment point is at one end of the current selection region. Otherwise, support the range swipe technique.

Required

When discontiguous selections are not supported, and normal mode is in use, support only the keyboard−based range adjust swipe technique when the initial adjustment point is within or at one end of the current selection region. Otherwise, support the range swipe technique.

Guidelines

Required

When the user is using the range adjust swipe technique, and the adjustment policy is to reselect:

Determine the new selection region to be the range identified by the anchor point and the final adjustment point.
Enlarge the new selection region to include the anchor region, if indicated by the anchor inclusion policy.

Required

When the user is using the range adjust swipe technique, and the adjustment policy is to enlarge, make the new selection region the larger of the following:

The current selection region
The region determined using the reselect adjustment policy

Required

When the user is using the keyboard−based area adjust swipe technique, and the adjustment policy is to balance:

If the final adjustment point is not within the current selection region, move (if necessary) the anchor point so that it is at the end of the current selection region farthest from either the initial or the final adjustment point. The use of the final adjustment point is recommended.
If the final adjustment point is within the current selection region, or when using the mouse−based technique, move (if necessary) the anchor point so that it is at the end of the current selection region farthest from the initial adjustment point.

Required

When the user is using the area adjust swipe technique, and the adjustment policy is to balance:
Determine the new selection region to be the range identified by the new anchor point and the final adjustment point.
Enlarge the new selection region to include the anchor region if the anchor point is unchanged and if indicated by the anchor inclusion policy.

Required

If the current selection region does not include an anchor element or if the anchor element is no longer in the selection region, identify the anchor element to be the element, if any, at the end of the range nearest the anchor point.

Required

After the user uses a mouse−based range adjust swipe technique, place the active cursor at one of the following:
If it is a text cursor, at the point at which the ADJUST button is released
If it is an element cursor, on the element at that end of the range

Essential Related Topics

For more information, see the **Adjustment Techniques** , Selection Policies , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Area Adjust Swipe Technique , Range Adjust Click Technique , Range Click Technique , Range Swipe Technique , Selection Models , Selection Modes , and Touch Adjust Swipe Technique reference pages.

Range Click Technique

NAME

Range Click Technique –Reference

Description

The range click technique is a group selection technique in which a contiguous range of elements can be selected or deselected by indicating each end of the range in discrete actions. The following describes the steps for mouse–based and keyboard–based techniques:

Mouse–based range click technique
: Click the SELECT button at one end of a range of (generally text) elements.
Move the mouse to the other end of the range.
Click the ADJUST button at the other end of the range.

Mouse–based range click technique, forcing toggle mode
: Click `Ctrl` SELECT at one end of a range of elements.
Move the mouse to the other end of the range.
Click the ADJUST button at the other end of the range.

Keyboard–based range click technique
: Press `Select`, `Space` (unless in text), or `Ctrl Space` with the cursor at one end of a range of elements.
Use keyboard navigation to move the cursor to the other end of the range.
Press `Shift Select`, `Shift Space` (unless in text), or `Ctrl Shift Space`.

When to Use

Required

Support use of the range click technique only when the elements in the scope form a natural sequential order, and when the most likely set of elements to be selected is a subsequence of those elements. This is most common in text and sometimes true in lists.

Required

Support the mouse–based range swipe technique in toggle mode only when discontiguous selections are supported.

Required

Support use of the keyboard–based range click technique in normal mode only when a text cursor is used for all navigation between the two ends of the range. The result of attempted use in other cases is undefined.

Guidelines

Mouse–Based Technique

Required

After the user uses a mouse–based range click technique:
: Define the current selection region to consist of the identified range.
Identify the anchor point to be the point at which the SELECT button was clicked.
Identify the anchor element to be the element, if any, within the selection region nearest to the anchor point.

Required

After the user uses a mouse–based range click technique, place the active cursor at one of the following:
: The point at which ADJUST is clicked, if it can be placed there

If it is an element cursor, on the element at the end of the range at which the ADJUST button is clicked
If there are no elements in the range, where the cursor previously was in the scope

Required

When the user uses the range click technique in select mode:
Select all the elements in the indicated range.
Deselect all other elements in the scope.

Required

If currently in select mode and the user uses the range click technique by clicking `Ctrl` SELECT in place of SELECT, activate toggle mode.

Required

When the user is using the range click technique in toggle mode:
Toggle all elements in the indicated range, based on the toggling policy.
Do not change the selection state of the other elements in the scope.

Keyboard–Based Technique

Required

After the user uses a keyboard–based range click technique:
Define the current selection region to consist of the identified range.
Identify the anchor point to be the point at which the user pressed `Select`, `Space` (unless in text), or `Ctrl Space`.
Identify the anchor element to be the element, if any, within the selection region nearest to the anchor point.

Required

When the user is using the keyboard–based range click technique in normal mode:
Select all the elements in the indicated range.
Deselect all other elements in the scope.

Required

When the user is using the keyboard–based range click technique in add mode, and contiguous selections are required:
Select all the elements in the indicated range.
Deselect all other elements in the scope.

Required

When the user is using the keyboard–based range click technique in add mode, and discontiguous selections are allowed:
Toggle all elements in the indicated range, based on the toggling policy.
Do not change the selection state of the other elements.

Essential Related Topics

For more information, see the **Selection Modes**, Selection Policies, and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Area Click Technique, Range Adjust Click Technique, Range Adjust Swipe Technique, Range Swipe Technique, and Selection Models reference pages.

Range Swipe Technique

NAME

Range Swipe Technique –Reference

Description

The range swipe technique is a group selection technique in which a contiguous range of elements can be selected or deselected by indicating the ends of the range in a single continuous action. The following describes the steps for mouse–based and keyboard–based techniques:

Mouse–based range swipe technique
: Press the SELECT button at one end of a range of elements.
Move the mouse to the other end of the range.
Release the SELECT button at the other end of the range.

Mouse–based range swipe technique, forcing toggle mode
: Press `Ctrl` SELECT at one end of a range of elements.
Move the mouse to the other end of the range.
Release SELECT at the other end of the range.

Keyboard–based range swipe technique

: Press and hold `Shift` down while pressing a navigation key. The range is determined by the initial and final cursor position.

When to Use

Required

: Support use of the range swipe technique only when the elements in the scope form a natural sequential order and when the most likely set of elements to be selected is a subsequence of those elements. This is most common in text and sometimes true in lists.

Required

: Support the mouse–based range swipe technique in toggle mode only when discontiguous selections are supported.

Required

: When discontiguous selections are not supported and add mode is in use, do not support the keyboard–based range swipe technique. Use the range adjust swipe technique instead.

Required

: When discontiguous selections are not supported, and normal mode is in use, support the keyboard–based range swipe technique only when the initial adjustment point is disjoint from the current selection region. Otherwise use the range adjust swipe technique.

Required

: When discontiguous selections are supported, support the keyboard–based range swipe technique only when the initial adjustment point is not at one end of the current selection region. Otherwise use the range adjust swipe technique.

Guidelines

Mouse–Based Technique

Required

: After the user uses a mouse–based range swipe technique:

Define the current selection region to consist of the identified range.
Identify the anchor point to be the point at which the SELECT button was pressed.
Identify the anchor element to be the element, if any, within the selection region nearest to the anchor point.

Required

After the user finishes a mouse−based range swipe technique, place the active cursor:
At the point at which the user released the SELECT button, if it can be placed there
If an element cursor, on the element at the end of the range at which the user released the SELECT button
If there are no elements in the range, where the cursor previously was in the scope

Required

When the user is using the range swipe technique in select mode:
Select all the elements in the indicated range.
Deselect all other elements in the scope.

Required

If the user is currently operating the range swipe technique in select mode and presses `Ctrl` SELECT in place of SELECT, change to toggle mode.

Required

When the user is using the range swipe technique in toggle mode:
Toggle all elements in the indicated range based on the toggling policy.
Do not change the selection state of the other elements.

Keyboard−Based Technique

Required

After the user uses a keyboard−based range swipe technique:
Define the current selection region to consist of the identified range.
Identify the anchor point to be the initial point at which the user pressed the `Shift` navigation key.
Identify the anchor element to be the element, if any, within the selection region nearest to the anchor point.

Required

When the user is using the keyboard−based range swipe technique in normal mode:
Select all the elements in the indicated range.
Deselect all other elements in the scope.

Required

When the user is using the keyboard−based range swipe technique in add mode:
Toggle all elements in the indicated range, based on the toggling policy.
Do not change the selection state of the other elements in the scope.

Essential Related Topics

For more information, see the **Selection Modes** , Selection Policies , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Area Swipe Technique , Range Adjust Click Technique , Range Adjust Swipe Technique , Range Click Technique , Selection Models , and Touch Swipe Technique reference pages.

Restore (Choice)

NAME

Restore (Choice) –Reference

Description

Restore is an action choice that returns a window to the size it was and the position it was in before the user minimized or maximized the window. If appropriate, the Restore choice may also be represented by a restore button in the title bar of a window.

When to Use

Required

If a window can be minimized, provide a Restore choice in the window icon's window menu to allow the user to return the window from the minimized state to its previous size and position.

Required

If a window can be maximized, provide a Restore choice to allow the user to return the window from the maximized state to its previous size and position.

Guidelines

Required

If a window is maximized and the user chooses Restore from the window menu, restore the window to the size and position it had before it was maximized.

Required

If a window is minimized (iconified) and the user chooses Restore from the window icon's window menu, restore the window to the size and position it had before it was minimized.

Required

If a primary window is minimized and the user invokes Restore, restore all of the secondary windows that the application has not closed.

Required

Do not make the Restore choice available when the window is neither minimized nor maximized.

Visuals

Required

Provide a restore button that implements the Restore choice on the title bar when the window is currently maximized.

Required

Replace the restore button with a maximize button in the title bar after a window has been restored to a state other than maximized.

Required

Replace the maximize button with a restore button in the title bar when the window has been maximized.

Default Action

Required

Make the Restore choice the default action associated with the window icon. If the window was maximized before it was minimized, restore it to its maximized state.

Required

Allow the user to invoke the default action of the window icon by double–clicking the SELECT button on it, or by pressing `Enter` or `keypadEnter` when it has focus.

Essential Related Topics

For more information, see the **Window Menu** reference page.

Supplemental Related Topics

For more information, see the Maximize (Choice) , Minimize (Choice) , and Size (Choice) reference pages.

Sash (Control)

NAME

Sash (Control) –Reference

Description

A sash is a box on a split bar through which users can directly manipulate the split bar to change the sizes of associated panes.

Figure 1 Sash

When to Use

Recommended

Use a sash to adjust the boundary between the panes of a paned box. Use the split bar solely to demarcate the boundary between the panes of a paned box.

Guidelines

Required

Design a sash to allow the user to adjust the position of the split bar and the size of the panes next to it.

Required

If you provide a sash, allow the user to navigate to it by pressing `tab`.

Mouse Behavior

Required

Allow the user to adjust the size of the panes in the split window by adjusting the position of the split bar with the sash (if the `Shift` key is not pressed).

Required

When a user moves the split bar with the sash and does not press the `Shift` key, shrink the size of the pane in the direction of movement. Enlarge the size of the pane on the opposite edge of the split bar by an equal amount.

Required

Allow the user to adjust the position of the split bar by both pointer operations and keyboard methods.

Required

When the user drags a sash while pressing the SELECT button or TRANSFER button, make the sash track the movement of the pointer. In a vertically oriented paned window, make the sash track the vertical position of the pointer. In a horizontally oriented paned window, make the sash track the horizontal position of the pointer.

Recommended

In a split window with multiple panes, when a user adjusts a pane to its minimum size with the pointer and continues to move the pointer in the same direction:

If there is another pane next to the minimized pane in the direction of movement, decrease the size of that pane while leaving the pointer on the original split bar.

Increase the size of the pane on the trailing edge of the original split bar as the size of the adjacent panes decrease. When the pane reaches its predetermined maximum size, do not move the split bar any further.

If no additional panes can be reduced by moving the leading edge of the split bar, do not move the split bar.

Recommended

When the user drags a sash while pressing `Shift` SELECT or `Shift` TRANSFER, have the pane in the initial direction of movement track the movement of the pointer while remaining the same size. That is, treat the current size of that pane as its minimum size. In a vertically oriented paned box, track the vertical position of the pointer. In a horizontally oriented paned box, track the horizontal position of the pointer.

Keyboard Behavior

Required

When the user presses the following keys with focus on the sash, move the sash as follows:

`[darr ]` and `[uarr ]`

If the sash can move vertically, move the sash one increment in the specified direction.

`[larr ]` and `[rarr ]`

If the sash can move horizontally, move the sash one increment in the specified direction.

`Ctrl[uarr ]` and `Ctrl[darr ]`

If the sash can move vertically, move the sash one large increment in the specified direction.

`Ctrl[rarr ]` and `Ctrl[larr ]`

If the sash can move horizontally, move the sash one large increment in the specified direction.

Recommended

When the user presses the following keys when focus is on the sash, move the sash as follows:

`Shift[uarr ]` and `Shift[darr ]`

If the sash can move vertically, move the pane in the initial direction one increment in the specified direction.

`Shift[rarr ]` and `Shift[larr ]`

If the sash can move horizontally, move the pane in the initial direction one increment in the specified direction.

`CtrlShift[uarr ]` and `CtrlShift[darr ]`

If the sash can move vertically, move the pane in the initial direction one large increment in the specified direction.

`CtrlShift[rarr ]` and `CtrlShift[larr ]`

If the sash can move horizontally, move the pane in the initial direction one large increment in the specified direction.

Visuals

Required

Make the sash look like a handle on the split bar separating two panes of a paned box.

Required

If the split bar has a sash, place the sash near the right end of the separators in a vertically split paned box or near the bottom end of the separators in a horizontally split paned box. Center the sash on the split bar in the other dimension.

Supplemental Related Topics

For more information, see the **Paned Box (Control)** and Window Navigation reference pages.

Save/Save As (Action Choice)

NAME

Save/Save As (Action Choice) -Reference

Description

The Save and Save As action choices are defined as follows:

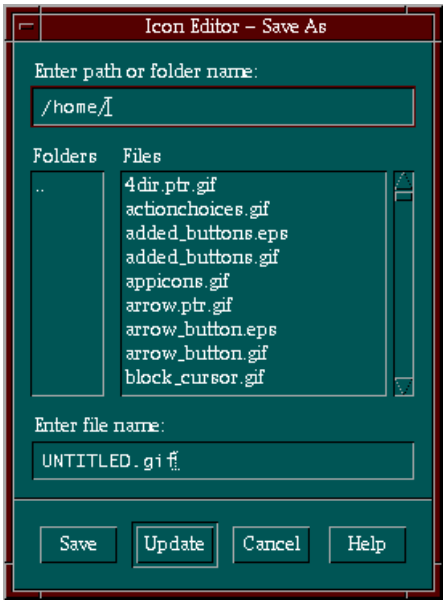
Save

An action choice in the File pull–down menu that stores the application's data into a previously determined file or object.

Save As

A dialog choice in the File pull–down menu that leads to a file selection dialog in which users can specify the file or object in which application data is to be stored.

Figure 1 Save As Dialog



Guidelines

Required

Place the Save and Save As choices in the File menu.

Required

Provide a Save choice for each file or object with a specified name that is not automatically saved when changed.

Required

Provide a Save As choice for each file or object that the user can save with a new name.

Required

Do not close the window or change the appearance of the data when the user chooses Save or Save As.

Required

The file or object to be saved should be the last file the user interacted with using Save As or Move To. Otherwise, the file or object to be saved should be the one that the user opened in the current window.

Required

If the user attempts to save a new file or object under an existing name, display a warning message stating that a loss of data will occur. For example:

```
myfile already exists.
Choose OK to overwrite.
```

Required

If the file or object being saved does not have a user–assigned name, either make Save unavailable or display the Save As dialog.

Required

If the user cannot save the data in the current file or object (for example, a file is read–only), display a warning message and allow the user to either cancel the operation or provide a new destination via a Save As dialog.

Save As Dialog

Required

Display the Save As dialog in a secondary window that is dependent on the window from which the user chooses Save As.

Required

Provide a Save (or OK) push button to allow the user to save the current file or object with the specified name.

Required

Provide a Cancel push button to allow the user to close the Save As dialog without saving the file or object.

Recommended

Provide a Help push button that displays help information for the Save As window.

Required

Provide a text–entry field to allow the user to type the name of the new file or object.

Required

Provide a drop–down list that allows the user to display a list of file types that can be saved.

Recommended

Provide a drop–down list that allows a user to specify the storage device (for example, /dev/tape).

Required

For objects, provide a list of appropriate containers. Highlight the current container and show the containment hierarchy by indenting containers that are within other containers.

Required

For objects, provide a list box that lists the names of all objects in the specified container.

Mnemonics

Required

Assign "S" as the mnemonic for the Save menu item.

Required

Assign "A" as the mnemonic for the Save As menu item.

Essential Related Topics

For more information, see the **File Menu** reference page.

Supplemental Related Topics

For more information, see the Close (Choice) , Dialog (Window) , Open (Choice) , and Text–Entry Field (Control) reference pages.

Scroll Bar (Control)

NAME

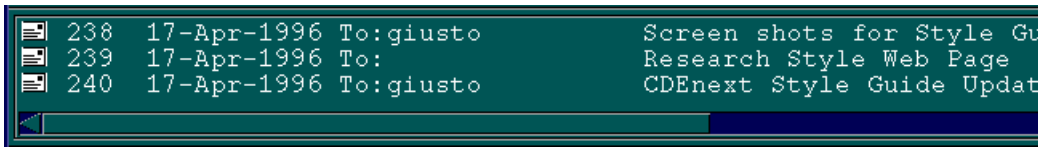
Scroll Bar (Control) –Reference

Description

A scroll bar is an element associated with an area that can be scrolled. The scroll bar indicates to a user that more information is available (or can be added in a horizontal or vertical direction) and can be scrolled into view.

A scroll bar contains a scroll box, which is the part of the scroll bar that indicates the position of the scrollable information in the view. The user can scroll the view by moving the scroll box in the scroll bar.

Figure 1 Scroll Bars



When to Use

Recommended

Provide a scroll bar if nonnumeric information cannot be fully shown within one window.

Guidelines

Required

Do not use a scroll bar to represent numeric values. Use a slider instead.

Required

Adjust the size of a scroll box so that it is proportional to the amount of information that can be visible in the area that can be scrolled in relation to the total amount of information.

Required

Adjust the position of a scroll box so that it indicates the position of the information visible in the area that can be scrolled in relation to the total amount of information.

Required

Change the size of a scroll box only when one of the following is true:
Information is added to or removed from data represented in the area that can be scrolled
The window's size changes

Recommended

Provide a scroll bar along each dimension in which more information is available in a scrolling area. For example, if the area can be scrolled vertically to display additional information, provide a vertical scroll bar.

Recommended

When using an explicit focus policy, do not place the active cursor on or within the scroll bar.

Recommended

If the information in a scrolled area is completely visible, the scroll bar can be removed.

Mouse Behavior

Required

When the user presses the SELECT button on the scroll bar's arrow button, scroll the associated scrolling area one unit in the direction of the arrow and repeat until the button is released.

Required

When the user presses the SELECT button in the scroll bar's scroll track, scroll the associated scrolling area one page in the direction of the mouse position relative to the scroll box and autorepeat (in the same direction) until the button is released.

Required

When the user presses the SELECT button in the scroll box, track the appropriate mouse coordinate with the scroll box, correspondingly scrolling the associated scrolling area, until the button is released. If the user moves the mouse perpendicularly away from the scroll bar (beyond a range specified by the operating environment), snap the scroll box back to its initial position at the time the user presses SELECT. When the user moves the mouse back into the scrolling range, begin tracking the mouse again.

Required

When the user presses the TRANSFER button in the scroll box or track, move the scroll box to the pointer position and scroll the corresponding area. While TRANSFER remains pressed, track the mouse with the scroll box in the exact same manner as when the user presses SELECT in the scroll box.

Recommended

When the user clicks `Ctrl` SELECT on one of the scroll bar's arrow buttons, scroll the scrolling area to the far end of the underlying information in the direction indicated by the arrow.

Recommended

When the user clicks `Ctrl` SELECT in the scroll bar's track, scroll the scrolling area to the far end of the underlying data in the direction of the mouse relative to the scroll box.

Recommended

When the user presses `Shift` SELECT in the scroll bar, scroll the associated scrolling area so that the scroll box is aligned with the position of the mouse. The alignment may be adjusted as the mouse is moved until the user releases the button.

Keyboard Behavior

Required

When the user presses a directional key while focus is in the scroll bar, scroll the associated scrolling area one unit in the indicated direction.

Required

When the user presses `Ctrl` plus a directional key while focus is in the scroll bar, scroll the associated scrolling area one large scrolling increment in the indicated direction.

Required

When the user presses `Begin` (or `Home`) or `End` in a horizontally aligned scroll bar, scroll the associated scrolling area to the leftmost or rightmost end, respectively.

Required

When the user presses `Ctrl Begin` (or `Ctrl Home`) or `Ctrl End` in a vertically aligned scroll bar, scroll the associated scrolling area to the topmost or bottommost end, respectively.

Required

When the user presses `PageUp` or `PageDown` in a horizontally aligned scroll bar, scroll the associated scrolling area by one page in the indicated direction.

Required

When the user presses `PageLeft` (or `Ctrl PageUp`) or `PageDown` (or `Ctrl PageRight`) in a vertically aligned scroll bar, scroll the associated scrolling area by one page in the indicated direction.

Essential Related Topics

For more information, see the **Internal Navigation** , Scrolling , and Slider (Control) reference pages.

Supplemental Related Topics

For more information, see the Mouse (Device) , Size (Choice) , and Emphasis (Cue) reference pages.

Scrolling

NAME

Scrolling –Reference

Description

Scrolling is a technique that allows the user to interact with data that is too large to be viewed or edited in the provided window.

When to Use

Recommended

When a control does not have a scrolling element associated with it, such as a scroll bar, do not allow it to be scrolled.

Optional

If the contents of a control indicates that additional elements exist beyond the bounds of the visible area (such as in text), allow the user to scroll the control through keyboard operations or autoscrolling even if the area does not have an associated scrolling element.

Guidelines

Scrolling Increment

Required

Make the unit scrolling increments (for example, the smallest amount a scrolled area can be scrolled when using an associated scroll bar) equal to those shown in Scrolling Increments .

Table 1 Scrolling Increments

Type of Element	Unit Scrolling Increment	
	Horizontal	Vertical
Text	Width of em (M) character	I c l b c l
Icons or controls	Width of smallest icon or control	I c s i c
Graphic	Five percent of the width of the view, or by units of a scale specified by the user	I p c h t v b c s s b u

Recommended

Make the page scrolling increment (for example, the amount by which an area is scrolled using a paging operation) equal to the width or height

of the scrolled area minus the unit scrolling increment.

Recommended

Define a medium scrolling increment that is larger than the unit scrolling increment and smaller than a paging increment.

Scrolling and Internal Navigation

Recommended

If a horizontally scrollable area does not support horizontal navigation internally, then:
Allow `[larr ]` and `[rarr ]` to scroll left and right one scrolling increment, respectively.
Allow `Ctrl [larr ]` and `Ctrl [rarr ]` to scroll left and right one medium scrolling increment, respectively.
Allow `Home` (or `Begin`) and `End` to scroll to the left and right ends of the data.

Recommended

If a vertically scrollable area does not support vertical navigation internally, then:
Allow `[uarr ]` and `[darr ]` to scroll up and down one scrolling increment, respectively.
Allow `Ctrl [uarr ]` and `Ctrl [darr ]` to scroll up and down one medium scrolling increment, respectively.
Allow `Ctrl Home` (or `Ctrl Begin`) and `Ctrl End` to scroll to the top and bottom ends of the data, respectively.

Paging

Required

When the user presses `PageUp` or `PageDown` and focus is on a scrollable area, scroll up or down by one page.

Required

When the user presses `PageLeft` (or `Ctrl PageUp`) or `PageRight` (or `Ctrl PageDown`) and focus is on a scrollable area, scroll left or right by one page.

Required

When the user is using the paging keys to scroll (and `ScrollLock` is not used), move the active cursor within the scrollable area so that it is visible, unless the cursor cannot be placed in the visible area after scrolling.

Required

Paging keys should apply to the most deeply nested appropriate scrollable area that contains the active cursor.

Navigation and Scrolling

Required

If a scrolled control (for example, a text field) maintains a cursor location even when it does not have focus, then when keyboard navigation is used to move focus to that control, scroll it, if necessary, so that the cursor location is visible.

Required

When the user navigates to an element within a scrollable area, scroll the area so that the element is visible.

Required

When any keyboard operation moves the cursor within a scrolled control, or inserts, deletes, or modifies elements at the cursor position, scroll the control so that the cursor is visible when the operation is complete.

Required

Design navigation operations to traverse through an entire scrollable area, not just the visible portion of it.

Scrolling with a Stationary Cursor

If the user can change the size of a scrollable control or area, and the cursor is visible in that area, scroll it so that the cursor remains visible.

Required

When the user scrolls with a mouse, do not move the cursor in the underlying data; instead, allow it to be scrolled outside of the visible area.

Required

If the active cursor is within a scrolled area, but is not visible, indicate focus emphasis by placing an element cursor on the entire scrolled area.

Recommended

If focus is within a scrolled area and the user presses a navigation or paging key, augmented with `ScrollLock`, scroll the area as if the key were interpreted by a scroll bar (even if there is none). The active cursor should remain where it is, even if it is scrolled outside of the visible area.

Mouse−Based Scrolling

Required

When the user presses `Cancel` (or `Esc`), cancel the mouse−based scrolling action and return the scrolling area to its position prior to the start of the scrolling operation.

Recommended

When the user scrolls with the mouse, continually update the information in the scrolled area as the mouse is moved.

Recommended

If it is not possible to continually update information as it is being scrolled, update the information in the scrolling area when the user pauses scrolling.

Recommended

When the user scrolls with the mouse, make the scrolling area appear to move smoothly by scrolling it in small increments.

Autoscrolling

Required

When a mouse−based selection operation is in progress within a control that supports scrolling and the user moves the pointer outside of the control, scroll towards the pointer. This is called "selection autoscrolling."

Recommended

When the user pauses (for a time that the user can set) during a drag−and−drop operation, at the edge of a scrolled area, scroll towards that edge. This is called "drag autoscrolling."

Recommended

When a drag−and−drop operation is initiated in a scrolled area and the elements being dragged are not valid outside that area, perform drag autoscrolling when the user moves the pointer outside of the area.

Essential Related Topics

For more information, see the Control Navigation and Scroll Bar (Control) reference pages.

Supplemental Related Topics

For more information, see the Control , Keyboard (Device) , and Mouse (Device) reference pages.

Secondary Window

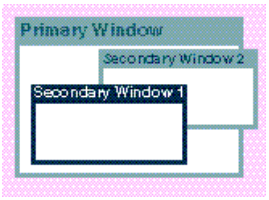
NAME

Secondary Window –Reference

Description

A secondary window is a window dependent on another window, either primary or another secondary, and is used to supplement the interaction in that window.

Figure 1 Secondary Windows



When to Use

Required

Display a secondary window when the user activates a dialog choice or, more generally, to allow the user to further specify information that is needed to complete a request.

Required

Use a secondary window to display a message.

Required

Use a secondary window to show properties associated with an object or other element.

Required

Use a secondary window to hold a tear–off control.

Required

Use a secondary window when a More choice provides access to additional controls.

Guidelines

Recommended

Design a secondary window so that resizing or maximizing is not necessary.

Recommended

Allow a window to have any number of dependent secondary windows.

Recommended

If appropriate, allow the user to resize, maximize, or minimize a secondary window that displays a view of an element.

Required

Do not show unavailable emphasis on a dialog choice that caused the display of a secondary window just because the window is still displayed.

Modality

Required

Support the following four modes of interaction with secondary windows:

Modeless

Allows interaction with the secondary window and all other windows, unless the secondary window was activated by a dialog choice in a modal window. In that case, limit interaction to windows displayed subsequent to the activation of the dialog choice.

Primary modal

Does not allow interaction with any ancestor of the window.

Application modal

Does not allow interaction with any window created by the same application except for this window and its secondary descendants, even if the application has multiple primary windows.

System modal

Does not allow interaction with any window on the workspace. This includes windows from all other applications and any window icon view. To indicate a system modal secondary window, the pointer should change shape to a caution pointer whenever it leaves the system modal secondary window.

Required

Do not display a secondary window as a system modal window unless your application is a manager application that requires that the user be restrained from performing any other desktop operation.

Required

Do not display a secondary window as an application modal window unless the task in progress in your application requires that the user be restrained from changing the state of the application in any way.

Required

Do not allow a modal secondary window to be resized or maximized.

Visuals

Required

If your application provides an Apply push button with a secondary window, provide an OK push button that, when activated, performs the default action for the secondary window and then closes the secondary window.

Required

Place the controls that the user can activate and that affect the contents or layout of the secondary window at the bottom of the secondary window.

Recommended

Do not provide a sizing border, minimize or maximize buttons, or a menu bar for a secondary window unless explicitly specified to by the user.

Recommended

Arrange the controls (that the user can activate) in a secondary window in the order of use or of frequency of use, with positive choices

first, followed by negative choices, canceling choices, and finally help choices.

Recommended

If the secondary window contains basic controls and the application supports a few advanced controls, hide the controls for these features and provide a More choice to expand the window and display them.

Recommended

If the secondary window contains basic controls and the application supports many advanced controls, provide these controls in a separate secondary window that can be displayed from a More choice on the secondary window with the basic controls.

Recommended

Provide menu cascade buttons within secondary windows only if any other alternative design would harm the layout of the secondary window.

Displaying Secondary Windows

Required

Remove a secondary window when the user closes or minimizes the primary window it depends on.

Required

Do not remove a secondary window just because the primary window it depends on loses input focus, except possibly those containing torn−off controls.

Required

If a secondary window is open when its associated primary window is closed or minimized, display that secondary window when its associated primary window is reopened or restored.

Required

When a secondary window is closed, close any secondary windows dependent on it. However, do not affect the associated primary window or any window that the secondary window is dependent on.

Required

If a secondary window can be stacked below its associated primary window, place it so that it will not be completely covered by the primary window.

Recommended

Display a secondary window on top of any window it depends on in the stacking order.

Recommended

Allow the user to move a secondary window.

Recommended

When you display a secondary window and the user does not need to see the information in the associated primary window, center the secondary window vertically and horizontally over the primary window. Be sure that the secondary window is below the title bar and menu bar of the primary window.

Recommended

When you display a secondary window where the user needs to see information or an element in the associated primary window, position the secondary window so that the necessary information or element is displayed.

Recommended

Allow the user to specify a default location, relative either to the screen or the window on which it depends, for commonly used secondary

windows.

Recommended

Remove a secondary window when it is no longer needed.

Recommended

If the dialog choice that caused the display of a secondary window is chosen while the window is still displayed, bring the secondary window and its dependent windows to the top of the stacking order without changing their positions.

Recommended

If an application supports multiple primary windows that show the same type of view, but for different information, allow it to share a secondary window between the different primary windows if the secondary window provides functions applicable to all the primary windows.

Essential Related Topics

For more information, see the **Primary Window** , Push Button (Predefined) , and Window Navigation reference pages.

Supplemental Related Topics

For more information, see the Cascading (Choice Type) , Dialog (Choice Type) , Selection Dialog (Secondary Window) , Size Border (Control) , and Stacking Order reference pages.

Selected Menu

NAME

Selected Menu –Reference

Description

A Selected menu is a cascading choice that appears as a menu–bar item. It provides access to choices that apply to the selected objects in the current view.

When to Use

Optional

Provide a Selected menu when the window contains objects that may be selected.

Guidelines

Required

In conjunction with the Menu Guidelines reference page, use Selected Menu Choices to decide which choices to include in the Selected menu and how to organize them.

Table 1 Selected Menu Choices

	Mnemonic	Menu Choice	Keyboard Function
Recommended	S	Properties	`Ctrl I`
Required	N	New	`Ctrl N`
Required	O	Open	`Ctrl O`
Recommended		Copy To ...	
Recommended		Move To ...	
Optional	E	Clear to Trash	`Delete` (if no Delete to Trash)
Required	D	Delete to Trash	`Delete`
Optional		Print ..	

Required

Use `CtrlN` and `Ctrl O` for the New and Open entries on the Selected menu only if your application does not have a state or contents that needs to be named and persistently stored.

Required

If your application manages and displays only one type of object that the user can select, replace the Selected label with one naming the type of object.

Required

Include type–specific choices at the bottom of the Selected menu that apply to the types of objects selected. Add and remove type–specific choices (rather than making those choices unavailable) as the selection changes.

Recommended

If there are a large number of type–specific choices, add type–specific cascaded menus as necessary.

Guidelines for Specific Menu Entries

Required

Print out the contents of each of the selected objects when the user chooses the Print choice on the Selected menu.

Recommended

Provide the Properties choice in the Edit menu if you do not provide a Selected menu.

Essential Related Topics

For more information, see the Edit Menu , Menu Bar (Menu Type) , Menu Guidelines , New (Action Choice) , Object , and Open (Choice) reference pages.

Selection

NAME

Selection –Reference

Description

Selection includes the following:

Pending delete

A mode of a selection scope in which an insertion within a selected region replaces the selected item.

Persistent selection

A selection whose state is unaffected by making a selection in another selection scope.

Primary selection

A selection that is automatically deselected when a new primary selection is made elsewhere.

Guidelines

Required

A selection scope must use, at any one time, either a primary selection approach or a persistent selection approach.

Required

When a scope uses a primary selection approach, deselect the primary selection when the user makes a nonempty primary selection in another scope.

Required

Selections in a scope that use a persistent selection approach should be unaffected by selections made in a different scope.

Required

If your application selection model allows at most one element at a time to be selected, use a persistent selection approach.

Recommended

When a scope uses a primary selection approach, deselect the primary selection when, during the use of a mouse–based selection technique in another scope that uses primary selection, emphasis indicates that one or more elements would be selected if the technique were completed at that point.

Recommended

If a selection model allows multiple elements to be selected, support either a primary or a persistent selection approach.

On a Selection

Required

Allow a selection if the pointer or cursor is on a selected element that cannot be activated or toggled (for example, a push button within a selected text region).

Required

Allow a selection if the pointer or cursor is between elements in a selected text region.

Optional

Allow a selection if the pointer or cursor is in the background in the midst of a dense region of selected elements.

Pending Delete

Required

If pending delete is enabled, allow the user to delete the contents of a selection before performing a transfer by inserting or pasting elements into the selection (except possibly on a link operation or, for a primary copy, at the edge of the primary selection).

Required

Support pending delete only when a text or graphics cursor is used.

Required

Disable pending delete behavior if it would lead to deletion of an object.

Enable pending delete behavior when a text cursor is used.

Disable pending delete behavior when an element cursor is used.

Recommended

Disable pending delete behavior when a graphics cursor is used. If it is enabled, allow the user to disable it.

Navigation and Interaction

Recommended

When the user is using keyboard navigation, do not skip over a selectable element simply because it is obscured; navigate to it and make it visible within its window. Raise it, if necessary, above any siblings or other controls in the window.

Recommended

If the user can manipulate the shape of elements (or a group of elements) through sizing borders or handles, display them when the elements (or group of elements) are selected, and remove them when they are deselected.

Essential Related Topics

For more information, see the **Selection Models** , Selection Modes , and Selection Techniques reference pages.

Supplemental Related Topics

For more information, see the Keyboard (Device) and Mouse (Device) reference pages.

Selection (CDE)

NAME

Selection (CDE) -Reference

Description

Selection is the way the user interacts with elements or objects in CDE.

Guidelines

Selection Models

Required

> There are five selection models: single selection, browse selection, multiple selection, range selection, and discontiguous selection. Each collection should have one or more appropriate selection models. The model limits the kinds of choices the user can make in the collection. Some collections enforce a selection model, while others allow the user or application to change it.

Mouse−Based Single Selection

Required

> In a collection that uses single selection, when the user clicks the SELECT button on a deselected element, move the location cursor to that element, select it, and deselect any other selection in the collection. Single selection is the simplest selection model for selecting a single element.

Mouse−Based Browse Selection

Required

> In a collection that uses browse selection, when the user releases the SELECT button on a selectable element, select that element, and deselect any other selection in the collection. As the pointer is dragged through selectable elements, select each element under the pointer and deselect the previously selected element. The selection should remain on the element where the SELECT button is released, and the location cursor should move there.
>
> Browse selection selects a single element. It also allows the user to browse through the collection by dragging the pointer.

Mouse−Based Multiple Selection

Required

> If your application contains collections that follow the multiple selection model, make the ADJUST button behave like the SELECT button when the TRANSFER button is configured to behave as the ADJUST button.

Required

> Allow the user to change an environment setting indicating that MB2 should be used for the adjust function instead. The ADJUST button can be used to toggle the selection state of elements under the multiple selection model.

Required

> In a collection that uses multiple selection, clicking the SELECT button or the ADJUST button on an unselected element adds that element to the current selection. Clicking the SELECT button or the ADJUST button on a selected element removes that element from the current selection. Clicking the SELECT button or the ADJUST button moves the location cursor to that element.

Mouse−Based Range Selection

Required

In a collection that uses range selection, pressing the SELECT button on an unselected element sets an anchor on the element, or at the position where the SELECT button was pressed, and deselects all elements in the collection. If the SELECT button is released before the drag threshold has been exceeded, then the element under the pointer should be selected. If mouse motion exceeds the drag threshold, then a new selection should begin. The anchor and the current position of the pointer determines the current range. As the pointer is dragged through the collection, highlight the current range. When the SELECT button is released, the anchor should not move and all the elements within the current range should be selected.

Required

In a collection that uses range selection, pressing the SELECT button on a currently selected element should not cause all other elements in the selection set to be deselected. If the SELECT button is released before the drag threshold is exceeded, then, at that point, all other elements should be deselected and the element under the pointer should remain selected. If mouse motion exceeds the drag threshold, then no element should be deselected and a drag operation should begin.

Required

In a text−like collection that uses range selection, make the anchor point the text pointer position when the SELECT button is pressed. The current range should consist of all elements between the anchor point and the current text pointer position.

Required

In text−like collections, order elements linearly, and always consider a text pointer to be between elements at a point near the actual pointer position.

Required

In a graphics−like or list−like collection that uses a marquee to indicate the range of a range selection, the current range should consist of those elements that fall completely within the marquee. If there is an anchor element, always make the marquee large enough to enclose it completely. Otherwise, use an anchor point to be the point at which the SELECT button was pressed; the anchor point determines one corner of the marquee. If the collection is not arranged as a list or matrix, extend the marquee to the pointer position. If the collection is arranged as a list or matrix, extend the marquee either to completely enclose the element under the pointer or to the pointer position. Clicking the SELECT button on a selectable element makes it an anchor element, selects it, and deselects all other elements.

Required

If your application contains collections that follow the range selection model, make the ADJUST button behave like `Shift` SELECT when the TRANSFER button is configured to behave as the ADJUST button.

Required

Allow the user to change an environment setting that indicates MB2 should be used for the ADJUST button function. The ADJUST button can then be used to extend the selection set in the same manner as `Shift` SELECT.

Required

In a collection that uses range selection, when the user presses `Shift` SELECT or the ADJUST button, the anchor should remain unchanged and an extended range for the selection is determined, based on one of the extension models.

Optional

Support the following mouse−based range selection models:

Reselect

The extended range is determined by the anchor and the current pointer position in exactly the same manner as when the selection was initially made.

Enlarge Only

The selection can only be enlarged. The extended range is determined by the anchor and the current pointer position, but then is enlarged to include the current selection.

Balance Beam

A balance point is defined at the midpoint of the current selection. When the user presses `Shift` SELECT or ADJUST on the opposite side of the balance point from the anchor, this model works exactly like the reselect model. When the user presses `Shift` SELECT, ADJUST, or starts a navigation action modified by `Shift` on the same side of the balance point as the anchor, this model moves the anchor to the opposite end of the selection and then works exactly like the reselect model.

When the user releases the SELECT button or ADJUST button, the anchor does not move, all the elements within the extended range are selected, and all the elements outside of it are deselected.

Mouse−Based Discontiguous Selection

Required

In a collection that uses discontiguous selection, make the behavior of the SELECT button exactly the same as in the range selection model. After the user sets the anchor with the SELECT button, `Shift` SELECT should work exactly as in the range selection model.

Required

In a collection that uses discontiguous selection, when the current selection is not empty and the user clicks `Ctrl` SELECT, move the anchor and location cursor to that point. If the current selection is not empty and the user clicks `Ctrl` SELECT on an element, toggle the selection state of that element and make it the anchor element.

Required

In a collection that uses discontiguous selection, pressing `Ctrl` SELECT and moving the pointer toggles the selection state of a range of elements. The range itself is determined exactly as for the pointer motion. Releasing `Ctrl` SELECT toggles the selection state of the elements in the range according to one of two models:

Anchor toggle

Toggling is based on an anchor element. If the range is anchored by a point and is not empty, the anchor element is set to the element within the range that is nearest to the anchor point. Toggling sets the selection state of all elements in the range to the inverse of the initial state of the anchor element.

Full toggle

The selection state of each element in the extended range is toggled.

Required

In a collection that uses discontiguous selection, after `Ctrl` SELECT toggles a selection, pressing `Shift` SELECT or `Ctrl Shift` SELECT extends the range of toggled elements. The extended range is determined in exactly the same way as when `Shift` SELECT is used to extend a range selection. When the user releases `Ctrl Shift` SELECT, the selection state of elements added to the range is determined by the toggle model in use (either anchor toggle or full toggle). If elements are removed from the range, they either revert to their state prior to the last use of `Ctrl` SELECT or change to the state opposite that of the elements remaining within the extended range.

Required

In a collection that uses discontiguous selection, allow the ADJUST button to be used to extend the range of a discontiguous selection. The extended range is determined in exactly the same way as when the ADJUST button is used to extend a range selection.

Required

Allow the user to change an environment setting that indicates MB2 should be used for the ADJUST button function. The ADJUST button can then be used to extend the selection set in the same manner as `Shift` SELECT.

Keyboard Selection

Required

The selection models should support keyboard selection modes according to the following rules:

Single selection supports only add mode.

Browse selection supports only normal mode.
Multiple selection supports only add mode.
Range selection supports normal mode. If it also supports add mode, normal mode is the default.
Discontiguous selection supports both normal mode and add mode. Normal mode is the default.

Required

Selection must be available from the keyboard. In normal mode, which is used for making simple contiguous selections from the keyboard, the location cursor is never disjoint from the current selection. In add mode, which is used for making more complex and possibly disjoint selections, the location cursor can move independently of the current selection.

Required

If a collection supports both normal mode and add mode, pressing `Shift F8` switches from one mode to the other. Mouse–based selection should not change when the keyboard selection mode changes. In editable components, add mode is a temporary mode that is exited when the user performs an operation on the selection or deselects the selection.

Keyboard–Based Single Selection

Required

In a collection that uses single selection, pressing the navigation keys moves the location cursor independently from the selected element. Pressing `Select` or `Spacebar` on an unselected element, selects the element with the location cursor and deselects any other selection in the collection.

Single selection supports only add mode. Pressing `Select` or `Spacebar` is similar to clicking the SELECT button.

Keyboard–Based Browse Selection

Required

In a collection that uses browse selection, pressing the navigation keys moves the location cursor and selects the cursored element, deselecting any other element. If the application has deselected all elements or if the cursor is left disjoint from the selection, pressing the `Select` key or the `Spacebar` selects the cursored element and deselects any other element.

Browse selection supports only normal mode. A navigation operation is similar to dragging the SELECT button.

Keyboard–Based Multiple Selection

Required

In a collection that uses multiple selection, pressing the navigation keys moves the location cursor independently from the current selection. Pressing `Select` or `Spacebar` on an unselected element adds the element to the current selection. Pressing `Select` or `Spacebar` on a selected element removes the element from the current selection.

Multiple selection supports only add mode. Pressing `Select` or `Spacebar` is similar to clicking the SELECT button.

Keyboard–Based Range Selection

Required

In a collection that uses range selection and that is in normal mode, pressing the navigation keys moves the location cursor and deselects the current selection. If the cursor is on an element, select it and move the anchor with the location cursor.

Required

Text–like collections can use a different model in which the navigation keys leave the anchor at its current location, except that if the current selection is not empty it is deselected and the anchor is moved to the location of the cursor prior to navigation. Range selection supports normal mode; if the collection also supports add mode, make normal mode the default.

Required

In a collection that uses range selection, whether in normal mode or add mode, pressing `Select` or `Spacebar` (except in a text

component) moves the anchor to the cursor, deselects the current selection, and, if the cursor is on an element, selects the element. Unless the anchor is on a deselected item, pressing `Shift Select` or `Shift Spacebar` (except in text) extends the selection from the anchor to the cursor, based on the extension model that `Shift` SELECT uses (Reselect, Enlarge Only, or Balance Beam).

Required

In a range selection, pressing `Select` or `Spacebar` is similar to clicking the SELECT button, and pressing `Shift Select` or `Shift Spacebar` extends the range as with `Shift` SELECT.

Required

In a collection that uses range selection and that is in normal mode, pressing `Shift` in conjunction with the navigation keys extends the selection, based on the extension model that `Shift` SELECT uses. If the current selection is empty, first move the anchor to the cursor. Then move the cursor according to the navigation keys and extend the selection based on the extension model that `Shift` SELECT uses.

Required

In a range selection, make shifted navigation extend the selection in a manner similar to dragging the pointer while holding `Shift` SELECT.

Required

In a collection that uses range selection and that is in add mode, pressing the navigation keys moves the location cursor but leaves the anchor unchanged.

Make shifted navigation move the location cursor according to the navigation keys and extend the selection, based on the extension model that `Shift` SELECT uses.

Required

Make shifted navigation in add mode similar to shifted navigation in normal mode, except that when the selection is empty the anchor should not move to the cursor prior to navigation.

Keyboard–Based Discontiguous Selection

Required

In a collection that uses discontiguous selection and is in normal mode, all keyboard operations should have the same effect as in the range selection model. Do not permit multiple discontiguous selections in normal mode.

Required

In a collection that uses discontiguous selection and is in add mode, pressing `Select` or `Spacebar` moves the anchor to the location cursor and initiates toggling. If the cursor is on an element, toggle the selection state of that element, but keep the selection state of all other elements unchanged. Pressing `Shift Select` or `Shift Spacebar` and shifted navigation operations extend the selection between the anchor and the location cursor, based on the toggle mechanism that `Ctrl` SELECT uses (anchor toggle or full toggle). In add mode, permit the use of the keyboard to make multiple discontiguous selections.

Canceling a Selection

Required

The `Cancel` key cancels or undoes any incomplete motion operation used for selection. Once the user presses `Cancel` to cancel a motion operation, ignore subsequent key and button releases until after all buttons and keys are released. Pressing `Cancel` while extending a selection or toggling leaves the selection state of all elements as they were prior to the button press.

Autoscrolling and Selection

Required

If the user drags the pointer out of a scrollable collection during a motion–based selection operation, use autoscrolling to scroll the collection in the direction of the pointer. If the user presses `Cancel` with the SELECT button pressed, cancel the selection operation.

Selecting and Deselecting All Elements

Required

In a collection that uses multiple, range, or discontiguous selection, pressing `Ctrl /` selects all the elements in the collection, places the anchor at the beginning of the collection, and leaves the location cursor at its previous position.

Required

In a collection that is in add mode, pressing `Ctrl \` deselects all the elements in the collection. In a collection that is in normal mode, pressing `Ctrl \` deselects all the elements in the collection, except the element with the location cursor if the location cursor is being displayed. In either mode, pressing `Ctrl \` leaves the location cursor at its current position and moves the anchor to the location cursor.

Using Mnemonics for Elements

Required

If your application supports mnemonics associated with selectable elements, typing a mnemonic while the collection has keyboard focus should be equivalent to moving the location cursor to the element and pressing `Select` or `Spacebar`.

Selection Actions

Required

When the keyboard focus policy is explicit, the destination component should be the editable component that last had keyboard focus. When the keyboard focus policy is implicit, the destination component should be the editable component that last received mouse button or keyboard input.

The destination component is used to identify the component on which certain operations, primarily data transfer operations, act. There can be only one destination component at a time.

Required

If the keyboard focus is in a component (or a pop–up menu of a component) that supports selections, operations that act on a selection should act on the selection in that component.

A selection operation should act on the component that has focus, if that component supports selections.

Required

If the keyboard focus is in a component (or a pop–up menu of a component) that supports some operation that does not act on a selection, invoking the operation should act on that component.

An operation that does not act on a selection should act on the component that has focus, if that component supports the operation.

Required

Inserting or pasting elements into a selection, except for a primary transfer operation at the bounds of the primary selection, should first delete the selection if pending delete is enabled. (Pending delete controls the conditions under which the selection is deleted. It is enabled by default.)

Required

In normal mode, inserting or pasting elements disjoint from the selection should also deselect the selection, except for primary transfer operations whose source and destination are in the same collection. In add mode, the selection should not be deselected.

In add mode, a transfer operation that is disjoint from the selection should not affect the selection.

Required

In editable list–like and graphics–like collections, pressing `Delete` deletes the selected elements.

Required

In editable text−like collections, pressing `Delete` or `Backspace` behaves as follows:

If the selection is not empty and the control is in normal mode, the selection is deleted.

If the selection is not empty, the control is in add mode, and the cursor is not disjoint from the selection, the selection is deleted.

If the selection is not empty and the control is in add mode, but the cursor is disjoint from the selection, pressing `Delete` deletes one character forward and pressing `Backspace` deletes one character backward.

If the selection is empty, pressing `Delete` deletes one character forward and pressing `Backspace` deletes one character backward.

Transfer Models

Required

If the move, copy, or link operation the user requests is not available, the transfer operation should fail.

Three transfer operations are generally available: copy, move, and link. The user requests one of these operations by pressing the buttons or keys appropriate for the type of transfer. In general, for mouse−based operations, pressing `Ctrl` forces a copy, `Shift` forces a move, and `Ctrl Shift` forces a link. However, any requested transfer operation must fail if that operation is not available.

Required

If a collection does not have a fixed insertion point or keeps elements ordered in a specific way, determine the insertion position for transferred data as follows:

For the TRANSFER button−based (or the SELECT button) primary and drag transfer operations, except as noted for text collections, the insertion position is the position at which the user releases the TRANSFER button (or the SELECT button).

In a text−like collection, when the user drops selected text, the insertion position is the position at which the user releases the TRANSFER button (or the SELECT button). When the user drops an icon, the insertion position is the text cursor and the data is pasted before it.

In a list−like collection, the insertion position for other transfer operations is the element with the location cursor and the data is pasted before it.

The insertion position is the position in the destination where transferred data is placed. Some mouse−based transfer operations place data at the pointer position, if possible. Other operations, including keyboard−based transfer, generally place the data at the location cursor.

Required

Support the use of MB1 to perform drag−and−drop operations.

Required

Pressing MB1 (the SELECT button) performs drag−and−drop operations. A drag can be initiated with either MB1 or MB2. This usage is compatible with other graphical user interface (GUI) environments.

Required

When MB2 of a 3−button mouse is configured to operate as the ADJUST button, do not perform any TRANSFER button operations when the user clicks MB2.

Required

Allow the user to change an environment setting that indicates that MB2 should be used for the ADJUST button function instead of the TRANSFER button. Clicking the ADJUST button should not result in the transfer of any data.

Required

Pressing the SELECT button should always initiate a drag if the drag is started on a selected item. The drag starts once the drag threshold has been reached. This is true for text regions, scrolling lists, and other similar elements.

Clipboard Transfer

Required

Make keyboard−based clipboard selection actions available in every editable collection in your application. Clipboard selection actions must be available from the keyboard.

Required

Pressing `Cut` (or `Shift Delete`) or selecting the Cut entry on the Edit menu cuts the selected elements from an editable component to the clipboard.

Required

Pressing `Copy` (or `Ctrl Insert`) or selecting the Copy entry on the Edit menu copies the selected elements to the clipboard.

Required

Pressing `Paste` (or `Shift Insert`) or selecting the Paste entry on the Edit menu pastes the contents of the clipboard into an editable component.

Required

If Paste or Paste Link is invoked using a component's pop–up menu, paste the data at the insertion position of the component. However, if the pop–up menu is popped up over a selection, first delete the selection, even if pending delete is disabled, and replace the selection with the pasted data if possible.

Popping up a pop–up menu over a selection indicates that a Paste or Paste Link operation should replace the selection.

Required

If Paste or Paste Link is invoked from the Edit menu or by a keyboard operation, and the insertion position in the target component is not disjoint from a selection, the pasted data should replace the selection contents if pending delete is enabled.

Pending delete determines whether the selection is deleted when the insertion position is not disjoint from the selection and Paste or Paste Link is invoked from the Edit menu or by a keyboard operation.

Primary Transfer

Required

In an editable collection, clicking the TRANSFER button, `Ctrl` TRANSFER, `Alt Copy`, or `Ctrl Alt Insert` copies the primary selection to the insertion position. (Note that the insertion position is usually different for mouse and keyboard operations.)

Required

In an editable collection, clicking `Ctrl Shift` TRANSFER places a link to the primary selection at the insertion position.

Required

A primary move should move the primary selection as well as the elements selected; that is, the element moved to the destination becomes selected as the primary selection. Primary copy and primary link should not select transferred data at the destination. This is the expected treatment of the selection in a move, copy, and link operation.

Quick Transfer

Required

All text components should support quick transfer.

Quick transfer is used to make a temporary selection and then immediately move, copy, or link that selection to the insertion position of the destination component. In text, quick transfer provides a convenient way to move, copy, or link text without disturbing the primary selection.

Required

If a component supports quick transfer, pressing `Alt` TRANSFER (or `Ctrl Alt` TRANSFER) and moving the pointer temporarily selects elements in the specified range and, on release, copies them to the insertion position of the destination component.

Required

If a component supports quick transfer, pressing `Alt Shift` TRANSFER and moving the pointer temporarily selects elements in the specified range and, on release, moves them to the insertion position of the destination component.

Required

If a component supports quick transfer, pressing `Ctrl Alt Shift` TRANSFER and moving the pointer temporarily selects elements in the specified range and, on release, places a link to them at the insertion position of the destination component.

Required

Quick transfer should not disturb the primary selection or affect the clipboard, except when the destination of the transfer is within or on the boundaries of the primary selection and pending delete is enabled. In this case, quick transfer should delete the contents of the primary selection, leaving an empty primary selection, before pasting the transferred elements.

Quick transfer is a secondary selection mechanism so it cannot disrupt the primary selection. When the destination of the transfer is in the primary selection, quick transfer replaces the primary selection with the secondary selection.

Required

With quick transfer, determine the range of the temporary selection from when the user presses the SELECT button, moves the pointer, and releases the SELECT button.

Required

If the user drags the pointer out of a scrollable collection while making the temporary selection, use autoscrolling to scroll the collection in the direction of the pointer. If the user releases the TRANSFER button with the pointer outside of the collection, or if the user presses the `Cancel` key with the TRANSFER button pressed, remove the highlighting and do not perform a transfer.

Drag Transfer

Required

In a collection that supports selection, releasing `Shift` TRANSFER or `Shift` SELECT forces a drag move operation. If a move is not possible, the operation should fail.

Required

In a collection that supports selection, releasing `Ctrl` TRANSFER or `Shift` SELECT forces a drag copy operation. If a copy is not possible, the operation should fail.

Required

In a collection that supports selection, releasing `Ctrl Shift` TRANSFER or `Shift` SELECT forces a drag link operation. If a link is not possible, the operation should fail.

Required

When a drag move operation moves a selection within the same component, move the selection along with the elements selected. In other words, when selected elements are moved with a drag operation, they should stay selected after the move.

Required

In text−like collections, when the user initiates a drag within a selected region, drag the entire text selection. Drag−and−drop actions need to operate on the entire selection.

Required

In list−like and graphics−like collections, when the user initiates a drag with either the SELECT button or the TRANSFER button on a selected element, drag the entire selection. Drag−and−drop actions need to operate on the entire selection.

Required

In list−like and graphics−like collections, when the user initiates a drag with the TRANSFER button or the SELECT button on an unselected element, drag just that element and leave the selection unaffected.

Unselected elements can be dragged without affecting the selection.

Required

When the user initiates a drag in an unselected region and the pointer is over two possible draggable elements, use the draggable element highest in the stacking order.

Required

At the start of a drag operation, replace the pointer with a drag icon.

Required

All drag icons should include a source indicator.

Required

Pressing the `Cancel` key ends a drag–and–drop operation by canceling the drag in progress.

Required

Releasing the TRANSFER button ends a drag–and–drop operation.

Required

When the TRANSFER button (or the SELECT button) is released, the drop operation should ordinarily occur at the location of the hot spot of the drag icon pointer and in the highest drop zone in the stacking order. However, if a drop occurs within a selection and pending delete is enabled, the transferred data should replace the contents of the entire selection.

Required

After a successful transfer, place the data in the drop zone and remove any transfer icon that your application used.

You can use a transfer icon to represent the type of data being transferred during a drop operation. A successful drop operation transfers data.

Required

After a failed transfer, keep the data at the drag source. Do not place it in the drop zone. Remove any transfer icon that your application used.

A failed drop operation does not transfer data.

Recommended

In a collection that supports selection, if pressing the TRANSFER button (or SELECT button) and moving the pointer results in the start of a drag operation, present feedback to the user that indicates that a copy, move, or link operation is in progress. Whether the operation is a copy, move, or link depends on the type of object created at the drop zone and whether the source object is removed.

Although, typically, an unmodified drag results in a move operation, depending on the location of the source object and the target drop zone, the drag may in fact result in a copy or link operation. For example, dragging an icon that represents an attachment to a mail message typically results in a copy of the attachment being created as opposed to the original being removed from the mail message. Any feedback presented should incorporate use of a drag icon that portrays the source object being manipulated.

Recommended

In a collection that supports selection, if pressing `Ctrl` TRANSFER (or `Ctrl` SELECT) and moving the pointer results in the start of a drag operation, present feedback to the user that indicates that a copy operation is in progress.

The feedback presented should incorporate use of a drag icon that portrays the source object being copied.

Recommended

In a collection that supports selection, if pressing `Ctrl Shift` TRANSFER (or `Ctrl Shift` SELECT) and moving the pointer results in the start of a drag operation, present feedback to the user that indicates that a link operation is in progress.

The feedback presented should incorporate use of a drag icon that portrays the source object being linked.

Recommended

In a collection that supports copy, move, or link operations that can be performed by dragging, the feedback presented to the user during the drag operation should indicate whether a single object or multiple objects are being manipulated.

Feedback provided during the drag operation should ensure that the user feels confident that the desired set of objects is being dragged. The drag icon used for multiobject drag operations should integrate the feedback used to indicate whether the operation is a move, copy, or link.

Optional

If your application allows the user to paste data into its data pane, allow the user to drag and drop files from the File Manager into the data pane.

The user should be able to drag and drop files into application data panes. The result should be the inclusion of some element of the file, or the display of an error message that indicates that the file selected cannot be incorporated into the application's data. Drag transfers that are accepted can result in a number of different responses from your application. For example:

The icon image for the file might be inserted at the drop point.
The application might perform some activity using the data contained within the file as its input.
The data contained within the file might be inserted at the drop point.
The name of the file might be inserted at the drop point.

Essential Related Topics

For more information, see the **Selection** , Selection Models , Selection Modes , and Selection Policies reference pages.

Supplemental Related Topics

For more information, see the Adjustment Techniques and Selection Techniques reference pages.

Selection Box (Control)

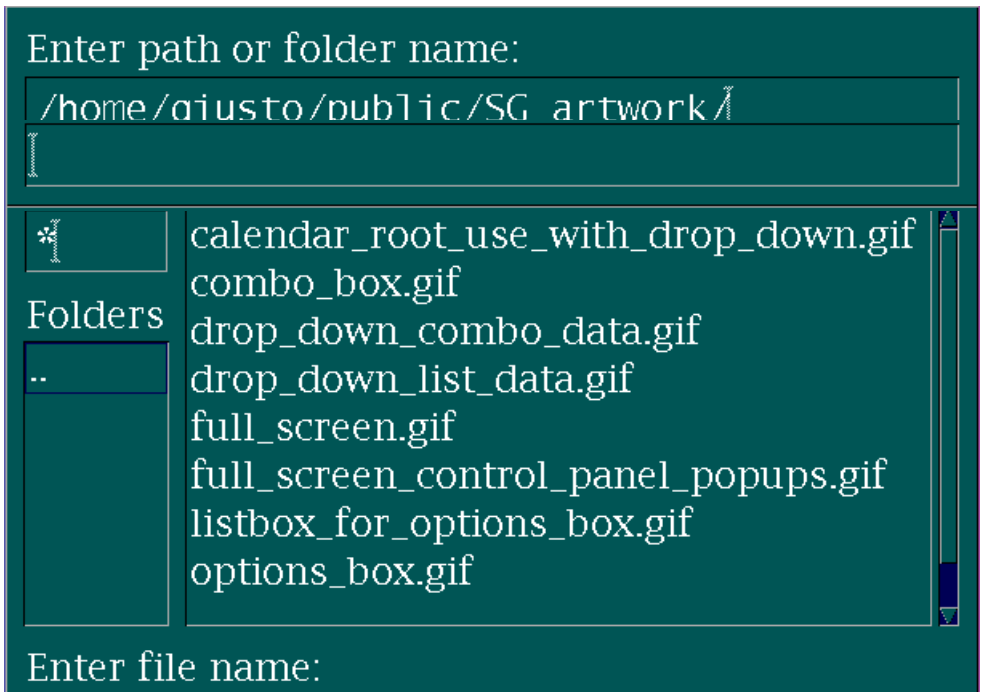
NAME

Selection Box (Control) -Reference

Description

A selection box is a combination text–list control in which both the text field and the list box are visible at all times. A selection box also contains labels associated with the text field and the list box.

Figure 1 Selection Box



When to Use

Recommended

Provide a selection box to allow users to make selections from a list of items.

Guidelines

Behavior

Required

When the user selects an item from the list, place the item in the text−entry field.

Recommended

Place a label above both the text−entry field and the list.

Recommended

When the user types an item into the text−entry field, display that item in the list box. If the item is not in the list, scroll the list box to the item that most closely matches the entry in the text−entry field. Use the best prefix match unless there is a better match. For example, if the user types in "Accounting" and there is no match in the list, scroll the list to "Accounts Payable" rather than "Accumulators."

Navigation

Required

Allow the user to navigate within the list with `Page Up` and `Page Down` while the cursor is in the text−entry field.

Required

Provide an OK push button to allow the user to accept any changes that have been made in the window and to remove the window.

Required

Provide a Cancel push button to allow the user to remove the window without applying any changes that were not previously applied in that window.

Required

Provide a Help push button to allow the user to display a window that contains context−sensitive help information.

Optional

Provide an Apply push button to allow the user to apply changes made by properties choices without removing the selection box.

Required

Use either the single or browse selection model in the list box.

Essential Related Topics

For more information, see the **Active Window** , Combination Text−List Control (Control Type) , and Secondary Window reference pages.

Supplemental Related Topics

For more information, see the Push Button (Predefined) and Text−Entry Field (Control) reference pages.

Selection Dialog (Secondary Window)

NAME

Selection Dialog (Secondary Window) –Reference

Description

A selection dialog is a secondary window that contains a selection box and associated push buttons that allow the user to make selections from a list of items for further processing by the application.

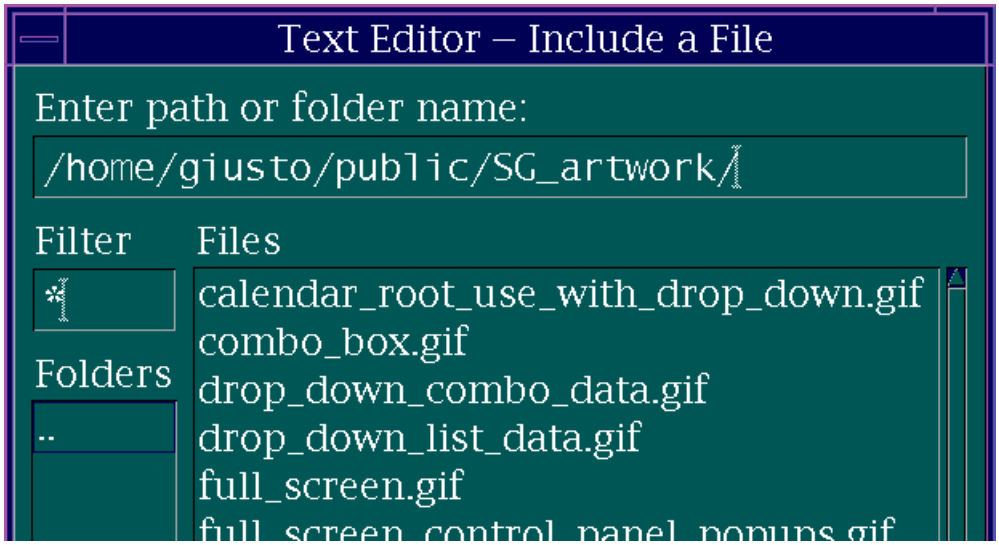



Figure 1 Selection Dialog