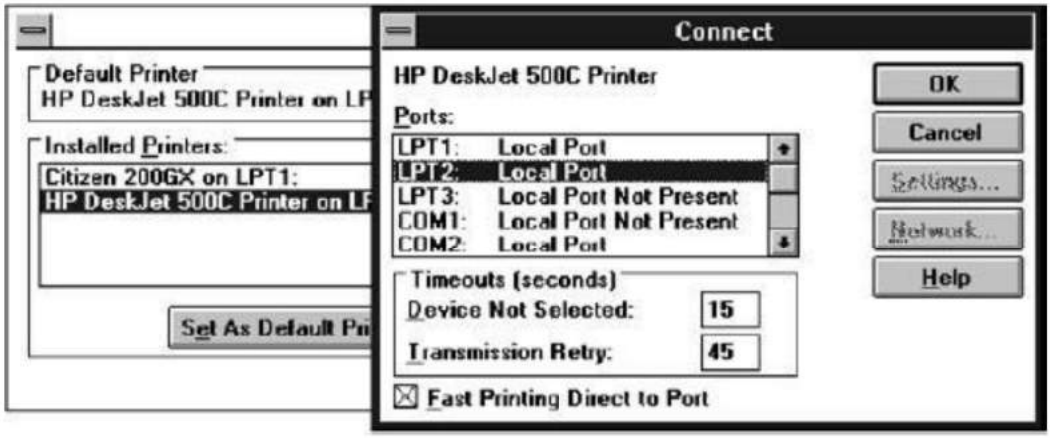
Modal and Modeless

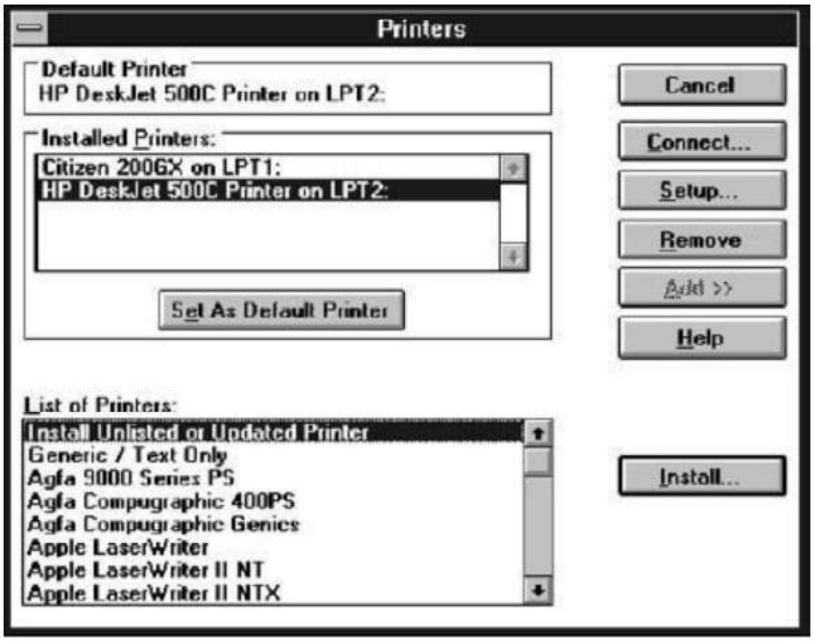
- Modal:
  — Use when interaction with any other window must not be permitted.
  — Use for:
    • Presenting information.
  — For example, messages (sometimes called a message box).
    • Receiving user input.
  — For example, data or information (sometimes called a prompt box).
    • Asking questions.
  — For example, data, information, or directions (sometimes called a question box).
  — Use carefully because it constrains what the user can do.
- Modeless:
  — Use when interaction with other windows must be permitted.
  — Use when interaction with other windows must be repeated.

Cascading and Unfolding

- Cascading:
  — Purpose:
    • To provide advanced options at a lower level in a complex dialog.
  — Guidelines:
    • Provide a command button leading to the next dialog box with a –To a Window‖ indicator, an ellipsis (. . .).
    • Present the additional dialog box in cascaded form.
    • Provide no more than two cascades in a given path.
    • Do not cover previous critical information.
  — Title Bar.
  — Relevant displayed information.
    • If independent, close the secondary window from which it was opened.
- Unfolding:
  — Purpose:
    • To provide advanced options at the same level in a complex dialog.
  — Guidelines:
    • Provide a command button with an expanding dialog symbol (>>).
    • Expand to right or downward.

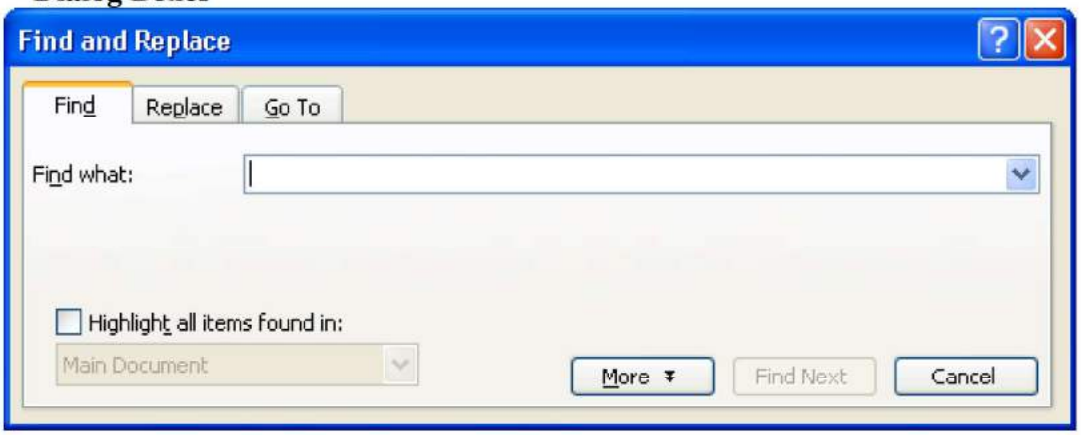


Cascaded Window



Unfolded Window

Dialog Boxes

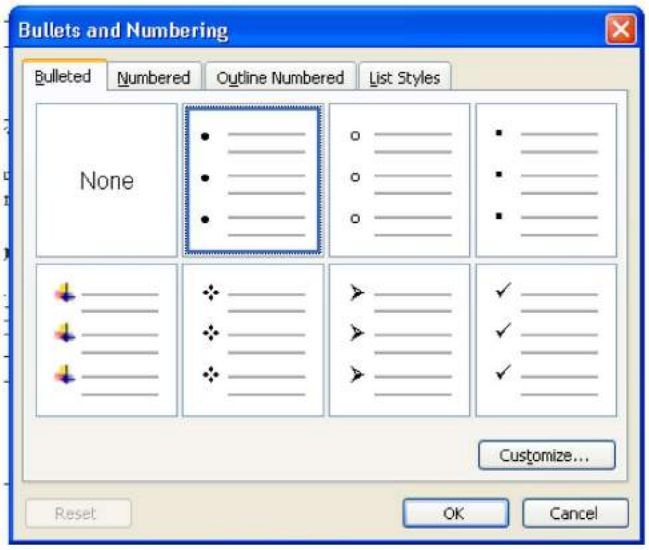


- Use for presenting brief messages.
- Use for requesting specific, transient actions.
- Use for performing actions that:
  — Take a short time to complete.
  — Are not frequently changed.
- Command buttons to include:
  — OK.
  — Cancel.
  — Others as necessary.

Property Sheets and Property Inspectors

Secondary windows provide two other techniques for displaying properties, *property sheets* and *property inspectors.*

Property Sheets



- Use for presenting the complete set of properties for an object.
- Categorize and group within property pages, as necessary.
  — Use tabbed property pages for grouping peer-related property sets.

— The recommended sizes for property sheets are:
  - 252 DLUs wide x 218 DLUs high
  - 227 DLUs wide x 215 DLUs high
  - 212 DLUs wide x 188 DLUs high

— Command buttons to include:
  - OK.
  - Cancel.
  - Apply.
  - Reset.
  - Others as necessary.

— For single property sheets, place the commands on the sheet.
— For tabbed property pages, place the commands outside the tabbed pages.

Property Inspectors

- Use for displaying only the most common or frequently accessed objects properties.
- Make changes dynamically.

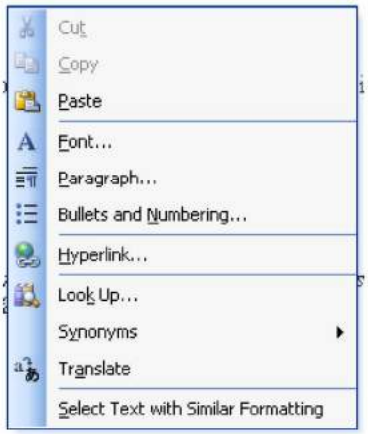
Message Boxes

- Use for displaying a message about a particular situation or condition.
- Command buttons to include:
  — OK.
  — Cancel.
  — Help.
  — Yes and No.
  — Stop.
  — Buttons to correct the action that caused the message box to be displayed.
- Enable the title bar close box only if the message includes a cancel button.
- Designate the most frequent or least destructive option as the default command

Palette Windows

- Use to present a set of controls.
- Design as resizable.
  — Alternately, design them as fixed in size.

Pop-up Windows



- Use pop-up windows to display:
  — Additional information when an abbreviated form of the information is the main presentation.
  — Textual labels for graphical controls.
  — Context-sensitive Help information

Window Management

Microsoft Windows also provides several window management schemes, a *single document interface, a multiple-document interface, workbooks*, and *projects*.

Single-Document Interface

- Description:
  — A single primary window with a set of secondary windows.
- Proper usage:
  — Where object and window have a simple, one-to-one relationship.
  — Where the object's primary presentation or use is as a single unit.
  — To support alternate views with a control that allows the view to be changed.
  — To support simultaneous views by splitting the window into panes.
- Advantages:

— Most common usage.
— Window manipulation is easier and less confusing.
— Data-centered approach.

- Disadvantage:
  — Information is displayed or edited in separate windows.

Multiple-Document Interface

- Description:
  — A technique for managing a set of windows where documents are opened into windows.
  — Contains:
    - A single primary window, called the parent.
    - A set of related document or child windows, each also essentially a primary window.

  — Each child window is constrained to appear only within the parent window.
  — The child windows share the parent window's operational elements.
  — The parent window's elements can be dynamically changed to reflect the requirements of the active child window.
- Proper usage:
  — To present multiple occurrences of an object.
  — To compare data within two or more windows.
  — To present multiple parts of an application.
  — Best suited for viewing homogeneous object types.
  — To clearly segregate the objects and their windows used in a task.
- Advantages:
  — The child windows share the parent window's interface components (menus, toolbars, and status bars), making it a very space-efficient interface.
  — Useful for managing a set of objects.
  — Provides a grouping and focus for a set of activities within the larger environment of the desktop.
- Disadvantages:
  — Reinforces an application as the primary focus.
  — Containment for secondary windows within child windows does not exist, obscuring window relationships and possibly creating confusion.
  — Because the parent window does not actually contain objects, context cannot always be maintained on closing and opening.
  — The relationship between files and their windows is abstract, making an MDI application more challenging for beginning users to learn.
  — Confining child windows to the parent window can be inconvenient or inappropriate for some tasks.
  — The nested nature of child windows may make it difficult for the user to distinguish a child window in a parent window from a primary window that is a peer with the parent window but is positioned on top.

Workbooks

- Description:

— A window or task management technique that consists of a set of views organized like a tabbed notebook.
— It is based upon the metaphor of a book or notebook.
— Views of objects are presented as sections within the workbook's primary windows; child windows do not exist.
— Each section represents a view of data.
— Tabs can be included and used to navigate between sections.
— Otherwise, its characteristics and behavior are similar to those of the multiple document interface with all child windows maximized.

- Proper usage:
  — To manage a set of views of an object.
  — To optimize quick navigation of multiple views.
  — For content where the order of the sections is significant.
- Advantages:
  — Provides a grouping and focus for a set of activities within the larger environment of the desktop.
  — Conserves screen real estate.
  — Provides the greater simplicity of the single-document window interface.
  — Provides greater simplicity by eliminating child window management.
  — Preserves some management capabilities of the multiple-document interface.
- Disadvantage:
  — Cannot present simultaneous views.

Projects

- Description:
  — A technique that consists of a container: a project window holding a set of objects.
  — The objects being held within the project window can be opened in primary windows that are peers with the project window.
  — Visual containment of the peer windows within the project window is not necessary.
  — Each opened peer window must possess its own menu bar and other interface elements.
  — Each opened peer window can have its own entry on the task bar.
  — When a project window is closed, all the peer windows of objects also close.
  — When the project window is opened, the peer windows of the contained objects are restored to their former positions.
  — Peer windows of a project may be restored without the project window itself being restored.
- Proper usage:
  — To manage a set of objects that do not necessarily need to be contained.
  — When child windows are not to be constrained.
- Advantages:
  — Provides a grouping and focus for a set of activities within the larger environment of the desktop.
  — Preserves some management capabilities of the multiple document interface.
  — Provides the greatest flexibility in the placement and arrangement of windows.

- Disadvantage:
  — Increased complexity due to difficulty in differentiating peer primary windows of the project from windows of other applications.

Organizing Window Functions

Window Organization

- Organize windows to support user tasks.
- Support the most common tasks in the most efficient sequence of steps.
- Use primary windows to:
  — Begin an interaction and provide a top-level context for dependent windows.
  — Perform a major interaction.
- Use secondary windows to:
  — Extend the interaction.
  — Obtain or display supplemental information related to the primary window.
- Use dialog boxes for:
  — Infrequently used or needed information.
  — -Nice-to-know- information.

Number of Windows

- Minimize the number of windows needed to accomplish an objective.
- The general rule:
  - Minimize the number of windows used to accomplish an objective.
  - Use a single window whenever possible. Consider, however, the user's task.
  - Don't clutter up a single window with rarely used information when it can be placed on a second, infrequently used, window.

Window Operations

Active Window

- A window should be made active with as few steps as possible.
- Visually differentiate the active window from other windows.

General Guidelines

- Design easy to use and learn windowing operations.
  — Direct manipulation seems to be a faster and more intuitive interaction style than indirect manipulation for many windowing operations.
- Minimize the number of window operations necessary to achieve a desired effect.
- Make navigating between windows particularly easy and efficient to do.
- Make the setting up of windows particularly easy to remember.
- In overlapping systems, provide powerful commands for arranging windows on the screen in user-tailorable configurations.

Opening a Window

- Provide an iconic representation or textual list of available windows.
  — If opening with an expansion of an icon, animate the icon expansion.
- When opening a window:
  — Position the opening window in the most forward plane of the screen.
  — Adapt the window to the size and shape of the monitor on which it will be presented.
  — Designate it as the active window.
  — Set it off against a neutral background.
  — Ensure that its title bar is visible.
- When a primary window is opened or restored, position it on top.
  — Restore all secondary windows to the states that existed when the primary window was closed.
- When a dependent secondary window is opened, position it on top of its associated primary window.
  — Position a secondary window with peer windows on top of its peers.
  — Present layered or cascaded windows with any related peer secondary windows.
- When a dependent secondary window is activated, its primary window and related peer windows should also be positioned at the top.
- If more than one object is selected and opened, display each object in a separate window.
- Designate the last window selected as the active window.
- Display a window in the same state as when it was last accessed.
  — If the task, however, requires a particular sequence of windows, use a fixed or consistent presentation sequence.
- With tiled windows, provide an easy way to resize and move newly opened windows.

Sizing Windows

- Provide large-enough windows to:
  — Present all relevant and expected information for the task.
  — Avoid hiding important information.
  — Avoid crowding or visual confusion.
  — Minimize the need for scrolling.
    - But use less than the full size of the entire screen.
- If a window is too large, determine:
  — Is all the information needed?
  — Is all the information related?
- Otherwise, make the window as small as possible.
  — Optimum window sizes:
    - For text, about 12 lines.
    - For alphanumeric information, about seven lines.
- Larger windows seem to have these advantages:
  - They permit displaying of more information.

- They facilitate learning: Data relationships and groupings are more obvious.
- Less window manipulation requirements exist.
- Breadth is preferred to depth (based on menu research).
- More efficient data validation and data correction can be performed.

- Disadvantages include:
  - Longer pointer movements are required.
  - Windows are more crowded.
  - More visual scanning is required.
  - Other windows more easily obscure parts of the window.
  - It is not as easy to hide inappropriate data.

Window Placement

- Considerations:
  — In placing a window on the display, consider:
    • The use of the window.
    • The overall display dimensions.
    • The reason for the window's appearance.
- General:
  — Position the window so it is entirely visible.
  — If the window is being restored, place the window where it last appeared.
  — If the window is new, and a location has not yet been established, place it:
    • At the point of the viewer's attention, usually the location of the pointer or cursor.
    • In a position convenient to navigate to.
    • So that it is not obscuring important or related underlying window information.
  — For multiple windows, give each additional window its own unique and discernible location.
    • A cascading presentation is recommended.
  — In a multiple-monitor configuration, display the secondary window on the same monitor as its primary window.
  — If none of the above location considerations apply, then:
    • Horizontally center a secondary window within its primary window just below the title bar, menu bar, and any docked toolbars.
  — If the user then moves the window, display it at this new location the next time the user opens the window.
    • Adjust it as necessary to the current display configuration.
  — Do not let the user move a window to a position where it cannot be easily repositioned.
- Dialog boxes:
  — If the dialog box relates to the entire system, center it on screen.
  — Keep key information on the underlying screen visible.
  — If one dialog box calls another, make the new one movable whenever possible.

Window Separation

- Crisply, clearly, and pleasingly demarcate a window from the background of the screen on which it appears.
  — Provide a surrounding solid line border for the window.
  — Provide a window background that sets the window off well against the overall screen background.
  — Consider incorporating a drop shadow beneath the window.

Moving a Window

- Permit the user to change the position of all windows.
- Change the pointer shape to indicate that the move selection is successful.
- Move the entire window as the pointer moves.
  — If it is impossible to move the entire window, move the window outline while leaving the window displayed in its original position.
- Permit the moving of a window without its being active.

Resizing a Window

- Permit the user to change the size of primary windows.
  — Unless the information displayed in the window is fixed or cannot be scaled to provide more information.
- Change the pointer shape to indicate that the resizing selection is successful.
- The simplest operation is to anchor the upper-left corner and resize from the lower right corner.
  — Also permit resizing from any point on the window.
- Show the changing window as the pointer moves.
  — If it is impossible to show the entire window being resized, show the window's outline while leaving the window displayed in its original position.
- When window size changes and content remains the same:
  — Change image size proportionally as window size changes.
- If resizing creates a window or image too small for easy use, do one of the following:
  — Clip (truncate) information arranged in some logical structure or layout when minimum size is attained, or
  — When no layout considerations exist, format (restructure) information as size is reduced, or
  — Remove less useful information (if it can be determined), or
  — When minimum size is attained, replace information with a message that indicates that the minimum size has been reached and that the window must be enlarged to continue working.
- Permit resizing a window without its being active.

Other Operations

Permit primary windows to be maximized, minimized, and restored.

Window Shuffling

Window shuffling must be easy to accomplish.

Keyboard Control/Mouse less Operation

- Window actions should be capable of being performed through the keyboard as well as with a mouse.
- Keyboard alternatives should be designated through use of mnemonic codes as much as possible.
- Keyboard designations should be capable of being modified by the user.

Closing a Window

- Close a window when:
  — The user requests that it be closed.
  — The user performs the action required in the window.
  — The window has no further relevance.
- If a primary window is closed, also close all of its secondary windows.
- When a window is closed, save its current state, including size and position, for use when the window is opened again.

Select the Proper Device-Based Controls

Device-based controls, often called input devices, are the mechanisms through which people communicate their desires to the system.

Characteristics of Device-Based Controls

Several specific tasks are performed using graphical systems.

- To point at an object on the screen.
- To select the object or identify it as the focus of attention.
- To drag an object across the screen.
- To draw something free form on the screen.
- To track or follow a moving object.
- To orient or position an object.
- To enter or manipulate data or information.

Direct and Indirect Devices

- *Direct* devices are operated on the screen itself. Examples include the light pen, the finger, and voice.
- *Indirect* devices are operated in a location other than the screen, most often on the desktop.

Trackball

- Description:
  — A spherical object (ball) that rotates freely in all directions in its socket.
  — Direction and speed is tracked and translated into cursor movement.
- Advantages:

— Direct relationship between hand and pointer movement in terms of direction and speed.
— Does not obscure vision of screen.
— Does not require additional desk space (if mounted on keyboard).

- Disadvantages:
  — Movement is indirect, in a plane different from the screen.
  — No direct relationship exists between hand and pointer movement in terms of distance.
  — Requires a degree of eye-hand coordination.
  — Requires hand to be removed from keyboard keys.
  — Requires different hand movements.
  — Requires hand to be removed from keyboard (if not mounted on keyboard).
  — Requires additional desk space (if not mounted on keyboard).
  — May be difficult to control.
  — May be fatiguing to use over extended time.

Joystick

- Description:
  — A stick or bat-shaped device anchored at the bottom.
  — Variable in size, smaller ones being operated by fingers, larger ones requiring the whole hand.
  — Variable in cursor direction movement method, force joysticks respond to pressure, movable ones respond to movement.
  — Variable in degree of movement allowed, from horizontal-vertical only to continuous.
- Advantages:
  — Direct relationship between hand and pointer movement in terms of direction.
  — Does not obscure vision of screen.
  — Does not require additional desk space (if mounted on keyboard).
- Disadvantages:
  — Movement indirect, in plane different from screen.
  — Indirect relationship between hand and pointer in terms of speed and distance.
  — Requires a degree of eye-hand coordination.
  — Requires hand to be removed from keyboard keys.
  — Requires different hand movements to use.
  — Requires hand to be removed from keyboard (if not mounted on keyboard).
  — Requires additional desk space (if not mounted on keyboard).
  — May be fatiguing to use over extended time.
  — May be slow and inaccurate.

Graphic Tablet

- Description:
  — Pressure-, heat-, light-, or light-blockage-sensitive horizontal surfaces that lie on the desktop or keyboard.
  — May be operated with fingers, light pen, or objects like a stylus or pencil.
  — Pointer imitates movements on tablet.

- Advantages:
  — Direct relationship between touch movements and pointer movements in terms of direction, distance, and speed.
  — More comfortable horizontal operating plane.
  — Does not obscure vision of screen.
- Disadvantages:
  — Movement is indirect, in a plane different from screen.
  — Requires hand to be removed from keyboard.
  — Requires hand to be removed from keyboard keys.
  — Requires different hand movements to use.
  — Requires additional desk space.

  — Finger may be too large for accuracy with small objects

Touch Screen

- Description:
  — A special surface on the screen sensitive to finger or stylus touch.
- Advantages:
  — Direct relationship between hand and pointer location in terms of direction, distance, and speed.
  — Movement is direct, in the same plane as screen.
  — Requires no additional desk space.
  — Stands up well in high-use environments.
- Disadvantages:
  — Finger may obscure part of screen.
  — Finger may be too large for accuracy with small objects.
  — Requires moving the hand far from the keyboard to use.
  — Very fatiguing to use for extended period of time.
  — May soil or damage the screen.
- Design Guidelines:
  — Screen objects should be at least 3/4″□□□3/4″□in size.
  — Object separation should be at least 1/8″.
  — Provide visual feedback in response to activation. Auditory feedback may also be appropriate.
  — When the consequences are destructive, require confirmation after selection to eliminate inadvertent selection.
  — Provide an instructional invitation to begin using.

Light Pen

- Description:
  — A special surface on a screen sensitive to the touch of a special stylus or pen.
- Advantages:
  — Direct relationship between hand and pointer movement in terms of direction, distance, and speed.
  — Movement is direct, in the same plane as screen.
  — Requires minimal additional desk space.

— Stands up well in high-use environments.
— More accurate than finger touching.

- Disadvantages:
  — Hand may obscure part of screen.
  — Requires picking it up to use.
  — Requires moving the hand far from the keyboard to use.
  — Very fatiguing to use for extended period of time.

Voice

- Description:
  — Automatic speech recognition by the computer.
- Advantages:
  — Simple and direct.
  — Useful for people who cannot use a keyboard.
  — Useful when the user's hands are occupied.
- Disadvantages:
  — High error rates due to difficulties in:
    • Recognizing boundaries between spoken words.
    • Blurred word boundaries due to normal speech patterns.
  — Slower throughput than with typing.
  — Difficult to use in noisy environments.
  — Impractical to use in quiet environments.

Mouse

- Description:
  — A rectangular or dome-shaped, movable, desktop control containing from one to three buttons used to manipulate objects and information on the screen.
  — Movement of screen pointer mimics the mouse movement.
- Advantages:
  — Direct relationship between hand and pointer movement in terms of direction, distance, and speed.
  — Permits a comfortable hand resting position
  — Selection mechanisms are included on mouse.
  — Does not obscure vision of the screen.
- Disadvantages:
  — Movement is indirect, in a plane different from screen.
  — Requires hand to be removed from keyboard.
  — Requires additional desk space.
  — May require long movement distances.
  — Requires a degree of eye-hand coordination.
- ***Mouse Usage Guidelines***
  - Provide a –hot zone‖ around small or thin objects that mouse positioning.
  - Never use double-clicks or double-drags as the only means operations.
  - Do not use mouse plus keystroke combinations.
  - Do not require a person to point at a moving target.

Keyboard

- Description:
  — Standard typewriter keyboard and cursor movement keys.
- Advantages:
  — Familiar.
  — Accurate.
  — Does not take up additional desk space.
  — Very useful for:
    • Entering text and alphanumeric data.
    • Inserting in text and alphanumeric data.
    • Keyed shortcuts—accelerators.
    • Keyboard mnemonics—equivalents.
  — Advantageous for:
    • Performing actions when less than three mouse buttons exist.
    • Use with very large screens.
    • Touch typists.
- Disadvantages:
  — Slow for non-touch-typists.
  — Slower than other devices in pointing.
  — Requires discrete actions to operate.
  — No direct relationship between finger or hand movement on the keys and cursor movement on screen in terms of speed and distance.
- **Keyboard Guidelines**
  - Provide keyboard accelerators.
  - Assign single keys for frequently performed, small-scale tasks.
  - Use standard platform accelerators.
  - Assign Shift-*key* combinations for actions that extend or are complementary to the actions of the key or key combination used without the Shift-*key*.
  - Assign Ctrl-*key* combinations for:
    • Infrequent actions.
    • Tasks that represent larger-scale versions of the task assigned to the unmodified key.
  - Provide keyboard equivalents.
  - Use standard platform equivalents.
  - Use the first letter of the item description.
  - If first letter conflicts exist, use:
    • Another distinctive consonant in the item description.
    • A vowel in the item description.
  - Provide window navigation through use of keyboard keys.

Selecting the Proper Device-Based Controls

- Consider the characteristics of the task.
  — Provide keyboards for tasks involving:
    • Heavy text entry and manipulation.
    • Movement through structured arrays consisting of a few discrete objects.

MODULE-V

Choose the Proper Screen-Based Controls

- Screen-based controls, often simply called *controls* and sometimes called *widgets*, are the elements of a screen that constitute its body.
- By definition, they are graphic objects that represent the properties or operations of other objects. A control may:
    - Permit the entry or selection of a particular value.
    - Permit the changing or editing of a particular value.
    - Display only a particular piece of text, value, or graphic.
    - Cause a command to be performed.
    - Possess a contextual pop-up window.
- Three extremely important principles regarding controls should be noted:
    - A control must:
      Look the way it works.

Work the way it looks.
    - A control must be used exactly as its design intended.
    - A control must be presented in a standard manner.
- The look of a control should make it obvious that it is a control. Its design characteristics should signal ―enterability‖ or ―clickability.‖ Microsoft Windows, for example, presents the following simple rules:
    - Raised elements can be pressed.
    - Recessed elements cannot be pressed.
    - Elements on a flat white background can be opened, edited, or moved.

Operable Controls

- Operable controls are those that permit the entry, selection, changing, or editing of a particular value, or cause a command to be performed.
- Classes include buttons, text entry/read-only, selection, combination entry/selection, and other specialized controls.

Buttons

- Description:
  — A square or rectangular-shaped control with a label inside that indicates action to be accomplished.
  — The label may consist of text, graphics, or both.
- Purpose:
  — To start actions.
  — To change properties.
  — To display a pop-up menu.
- Advantages:
  — Always visible, reminding one of the choices available.
  — Convenient.
  — Can be logically organized in the work area.
  — Can provide meaningful descriptions of the actions that will be performed.
  — Larger size generally provides faster selection target.
  — Can possess 3-D appearance:
    • Adds an aesthetically pleasing style to the screen.
    • Provides visual feedback through button movement when activated.
  — May permit use of keyboard equivalents and accelerators.
  — Faster than using a two-step menu bar/pull-down sequence.
- Disadvantages:
  — Consumes screen space.
  — Size limits the number that may be displayed.
  — Requires looking away from main working area to activate.
  — Requires moving the pointer to select.
- Proper usage:
  — Use for frequently used actions that are specific to a window.
    • To cause something to happen immediately.
    • To display another window.

- To display a menu of options.
- To set a mode or property value.

- A button comes in three styles.

Command buttons.

Toolbar buttons without labels.

Symbol button

Command Buttons

Command button guidelines include the following.

Usage

- For windows with a menu bar:
  — Use to provide fast access to frequently used or critical commands.
- For windows without a menu bar:
  — Use to provide access to all necessary commands.

Structure

- Provide a rectangular shape with the label inscribed within it.
- Give the button a raised appearance.
- Maintain consistency in style throughout an application.

Labels

- Use standard button labels when available.
- Provide meaningful descriptions of the actions that will be performed.
- Use single-word labels whenever possible.
  — Use two-three words for clarity, if necessary.
- Use mixed-case letters with the first letter of each significant label word capitalized.
- Display labels:
  — In the regular system font.
  — In the same size font.
- Do not number labels.

- Center the label within the button borders, leaving at least two pixels between the text and the button border.
- Provide consistency in button labeling across all screens.

Size

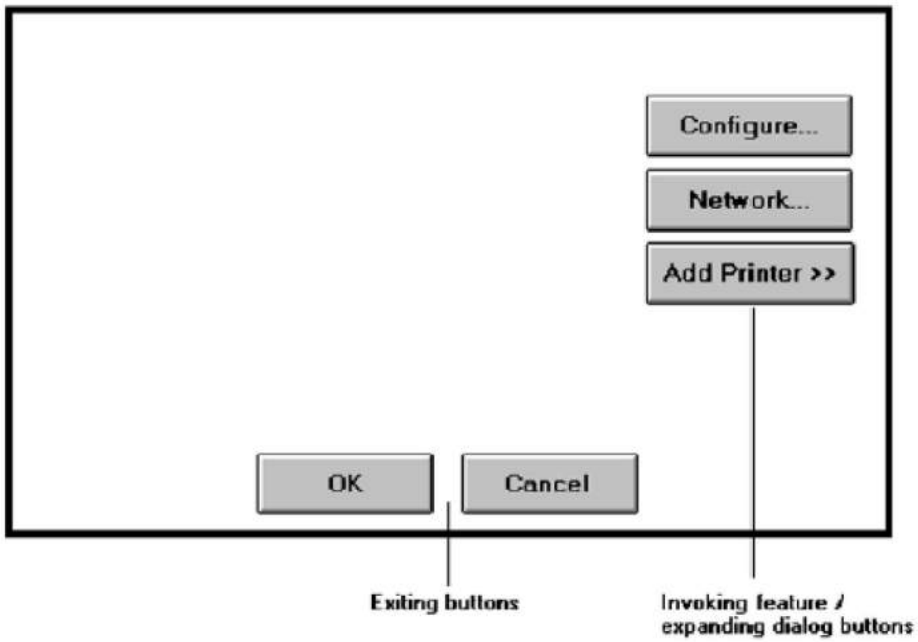
- Provide as large a button as feasible.
- Maintain consistent button heights and widths.
- Exception: Buttons containing excessively long labels may be wider.

Number

- Restrict the number of buttons on a window to six or fewer.

Location and Layout

- Maintain consistency in button location between windows.
- Never simply -fitl buttons in available space.
- If buttons are for exiting the dialog:
  — Position them centered and aligned horizontally at the bottom.
- If buttons are used for invoking a dialog feature or expanding the dialog:
  — Position them centered and aligned vertically on the right side.
- If a button has a contingent relationship to another control:
  — Position it adjacent to the related control.
- If a button has a contingent relationship to a group of controls:
  — Position it at the bottom or to right of related controls.
- If, due to space constraints, exiting and expanding/invoking feature buttons must be placed together:
  — If at the bottom, place exiting buttons to the right, separating the groupings by one button's width.
  — If along the right side, place exiting buttons at the bottom, separating the groupings by one button's height.
- For exiting and expanding/invoking feature buttons, do not:
  — Align with the other screen controls.
  — Present displayed within a line border.
- Provide equal and adequate spacing between adjacent buttons.
- Provide adequate spacing between buttons and the screen body controls.



Organization

- Organize standard buttons in the manner recommended by the platform being used.
- For other buttons, organize them in common and customary grouping schemes.
  - — For buttons ordered left to right, place those for most frequent actions to the left.
  - — For buttons ordered top to bottom, place those for most frequent actions at the top.
- Keep related buttons grouped together.
- Separate potentially destructive buttons from frequently chosen selections.
- Buttons found on more than one window should be consistently positioned.
- The order should never change.
- For mutually exclusive actions, use two buttons; do not dynamically change the text.
- Windows recommends the following:
  - An affirmative action to the left (or above).
  - The default first.
  - OK and Cancel next to each other.
  - Help last, if supported.

Intent Indicators

- When a button causes an action to be immediately performed, no intent indicator is necessary.

- When a button leads to a cascading dialog, include an ellipsis (...) after the label.

- When a button leads to a menu, include a triangle pointing in the direction the menu will appear after the label.

- When a button leads to an expanding dialog, include a double arrow (>>) with the label.

- When a button has a contingent relationship to another control that must be indicated, include a single arrow (->) pointing at the control.

Expansion Buttons

- Gray them out after expansion.
- Provide a contraction button, if necessary.
  - — Locate it beneath, or to right of, the expansion button.
  - — Gray it out when not applicable.

Defaults

- Intent:
  - — When a window is first displayed, provide a default action, if practical.
- Selection:
  - — A default should be the most likely action:
    - • A confirmation.
    - • An application of the activity being performed.
    - • A positive action such as OK, unless the result is catastrophic.
  - — If a destructive action is performed (such as a deletion), the default should be Cancel.
- Presentation:
  - — Indicate the default action by displaying the button with a bold or double border.
- Procedures:
  - — The default can be changed as the user interacts with the window.
  - — When the user navigates to a button, it can temporarily become the default.
  - — Use the Enter key to activate a default button.
  - — If another control requires use of the Enter key, temporarily disable the default while the focus is on the other control.
  - — Permit double-clicking on a single selection control in a window to also carry out the default command.

Unavailable Choices

- Temporarily unavailable choices should be dimmed or grayed out.

Keyboard Equivalents and Accelerators

- Equivalents:
  — Assign a keyboard equivalent mnemonic to each button to facilitate keyboard selection.
  — The mnemonic should be the first character of the button's label.
    • If duplication exists in first characters, for duplicate items, use another character in the label.
    • Preferably, choose the first succeeding consonant.
  — Designate the mnemonic character by underlining it.
  — Maintain the same mnemonic on all identical buttons on other screens.

- Accelerators:
  — Assign a keyboard accelerator to each button to facilitate keyboard selection.

Scrolling

- If a window can be scrolled, do not scroll the command buttons.
  — Exception: if the screen cannot scroll independently of the buttons.
- Use buttons to move between multipage forms, not scroll bars.
  — Label buttons Next and Previous.

Button Activation

Pointing:
— Highlight the button in some visually distinctive manner when the pointer is resting on it and the button is available for selection.
Activation:
— Call attention to the button in another visually distinctive manner when it has been activated or pressed.
— If a button can be pressed continuously, permit the user to hold the mouse button down and repeat the action.

Toolbars

- *Toolbars* are compilations of commands, actions, or functions, usually graphical in structure but sometimes textual, grouped together for speedy access.
- Toolbars may also be called *button bars*, *control bars,* or *access bars*. Specialized toolbars may also be referred to as *ribbons*, *toolboxes*, or *palettes*. Toolbars may also appear in palette windows.

Usage

- To provide easy and fast access to most frequently used commands or options across multiple screens.
- To invoke a sub application within an application.
- To use in place of certain menu items.

Structure

- Images:
  — Provide buttons of equal size.
  — Create a meaningful and unique icon.
    • Design them using icon design guidelines.
  — Center the image within the button.
  — Give the button a raised appearance.
  — Ensure that toolbar images are discernible from Web page graphical images.
- Text:
  — Create a meaningful label, adhering to label guidelines for command buttons.
  — Create toolbar buttons of equal size, following the size guidelines recently described.
- Consistency:
  — Use the same icon throughout an application and between applications.

Size

- Button:
  — 24 (w) by 22 (h) pixels, including border.
  — 32 (w) by 30 (h) pixels, including border.
  — Larger buttons can be used on high-resolution displays.
- Label:
  — 16 (w) by 16 (h) pixels.
  — 14 (w) by 24 (h) pixels.
- Default:
  — Provide the smaller size as the default size with a user option to change it.
- Image:
  — Center the image in the button.

Organization

- Order the buttons based on common and customary grouping schemes.
  — For buttons ordered left to right, place those for the most frequently used actions to the left.
  — For buttons ordered top to bottom, place those for the most frequently used actions at the top.
- Keep related buttons grouped together.
- Separate potentially destructive buttons from frequently chosen selections.
- Permit user reconfiguration of button organization.

Location

- Position main features and functions bar horizontally across top of window just below menu bar.
- Position subtask and sub features bars along sides of window.
- Permit the location of the bar to be changed by the user.
- Permit display of the bar to be turned on or off by the user.
  — Also provide access through standard menus.

Active Items

- Make only currently available toolbar items available.
- Temporarily not available items may be displayed grayed out.

Customization

- Permit toolbars to be turned off by the user.
- Allow the customizing of toolbars.
  — Provide a default, however.

Keyboard Equivalents and Accelerators

- Equivalents:
  — Assign keyboard equivalents to facilitate keyboard selection.
  — Maintain the same mnemonic on all identical buttons on all screens.
- Accelerators:
  — Assign a keyboard accelerator to facilitate keyboard selection.

Button Activation

- Pointing:
  — Highlight the button in some visually distinctive manner when the pointer is resting on it and the button is available for selection.
- Activation:
  — Call attention to the button in another visually distinctive manner when it has been activated or pressed.

Text Entry/Read-Only Controls

- A Text Entry/Read-Only control contains text that is exclusively entered or modified through the keyboard.
- It may also contain entered text being presented for reading or display purposes only.

Text Boxes

- Description:
  — A control, usually rectangular in shape, in which:

- Text may be entered or edited.
- Text may be displayed for read-only purposes.

— Usually possesses a caption describing the kind of information contained within it.

— An outline field border:
- Is included for enterable/editable text boxes.
- Is not included for read-only text boxes.

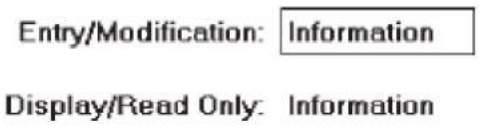
— Two types exist:
- Single line.
- Multiple line.

— When first displayed, the box may be blank or contain an initial value.

- Purpose:
  — To permit the display, entering, or editing of textual information.
  — To display read-only information.
- Advantages:
  — Very flexible.
  — Familiar.
  — Consumes little screen space.
- Disadvantages:
  — Requires use of typewriter keyboard.
  — Requires user to remember what must be keyed.
- Proper usage:
  — Most useful for data that is:
    - Unlimited in scope.
    - Difficult to categorize.
    - Of a variety of different lengths.
  — When using a selection list is not possible.

Types of text box

- Two types of *text boxes* exist. One consists of a rectangular box into which information is typed. It may also be referred to as an *edit* control.
- The second is also rectangular in shape but contains text displayed purely for read-only purposes. The former type has historically been referred to as an *entry field*, the latter as an *inquiry* or *display field*.



Two forms of Text Box
Single-Line and Multiple-Line Text Boxes

- Single line:
  — Description:
    - A control consisting of no more than one line of text.
  — Purpose:

• To make textual entries when the information can be contained within one line of the screen.

— Typical uses:

• Typing the name of a file to save.
• Typing the path of a file to copy.
• Typing variable data on a form.
• Typing a command.

- Multiple line:

— Description:

• A control consisting of a multiline rectangular box for multiple lines of text.

— Purpose:

• To type, edit, and read passages of text.

— Typical uses:

• Creating or reading an electronic mail message.
• Displaying and editing text files.

Captions

- Structure and size:

— Provide a descriptive caption to identify the kind of information to be typed, or contained within, the text box.
— Use a mixed-case font.
— Display the caption in normal intensity or in a color of moderate brightness.

- Formatting:

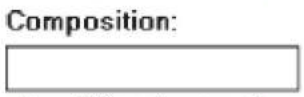
— Single fields:

• Position the field caption to the left of the text box.

— Place a colon (:) immediately following the caption.
— Separate the colon from the text box by one space.

• Alternately, the caption may be placed above the text box.

— Place a colon (:) immediately following the caption.
— Position above the upper-left corner of the box, flush with the left edge.
— Multiple occurrence fields:

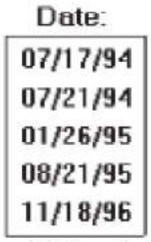


• For entry/modification text boxes:

— Position the caption left-justified one line above the column of entry fields.

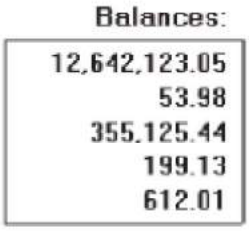
• For display/read-only boxes:

— If the data field is long and fixed-length, or the displayed data is about the same length, center the caption above the displayed text box data.



— If the data displayed is alphanumeric, short, or quite variable in length, left-justify the caption above the displayed text box data.

— If the data field is numeric and variable in length, right-justify the caption above the displayed text box data.



Fields

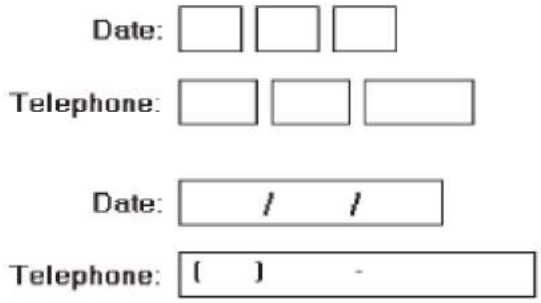
- Structure:

  — Identify entry/modification text boxes with a line border or reverse polarity rectangular box.

    • To visually indicate that it is an enterable field, present the box in a recessed manner.

    • Present display/read-only text boxes on the window background.

  — Break up long text boxes through incorporation of slashes (/), dashes (-), spaces, or other common delimiters.



- Size:
  — Size to indicate the approximate length of the field.
  — Text boxes for fixed-length data must be large enough to contain the entire entry.
  — Text boxes for variable-length data must be large enough to contain the majority of the entries.
  — Where entries may be larger than the entry field, scrolling must be provided to permit keying into, or viewing, the entire field.
  — Employ word wrapping for continuous text in multiple-line text boxes.
- Highlighting:
  — Call attention to text box data through a highlighting technique.
    - Higher intensity.
    - If color is used, choose one that both complements the screen background and contrasts well with it.
- Unavailable fields:
  — Gray-out temporarily unavailable text boxes.
- Fonts:
  — To support multiple fonts, use a *Rich-Text Box*.

Selection Controls

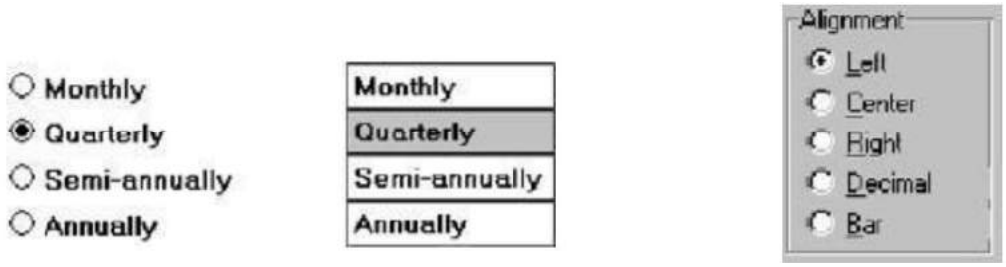
- A selection control presents on the screen all the possible alternatives, conditions, or choices that may exist for an entity, property, or value.
- The relevant item or items are selected from those displayed.
- Selection controls include radio buttons, check boxes, list boxes, drop-down/pop-up list boxes, and palettes.

Radio Buttons

- Description:
  — A two-part control consisting of the following:
    - Small circles, diamonds, or rectangles.
- Choice descriptions.
  — When a choice is selected:
    - The option is highlighted.
    - Any existing choice is automatically unhighlighted and deselected.
- Purpose:
  — To set one item from a small set of mutually exclusive options (2 to 8).
- Advantages:

— Easy-to-access choices.
— Easy-to-compare choices.
— Preferred by users.

- Disadvantages:
  — Consume screen space.
  — Limited number of choices.
- Proper usage:
  — For setting attributes, properties, or values.
  — For mutually exclusive choices (that is, only one can be selected).
  — Where adequate screen space is available.
  — Most useful for data and choices that are:
    • Discrete.
    • Small and fixed in number.
    • Not easily remembered.
    • In need of a textual description to meaningfully describe the alternatives.
    • Most easily understood when the alternatives can be seen together and compared to one another.
    • Never changed in content.
  — Do not use:
    • For commands.
    • Singly to indicate the presence or absence of a state.



Choice Descriptions

- Provide meaningful, fully spelled-out choice descriptions clearly describing the values or effects set by the radio buttons.
- Display in a single line of text.
- Display using mixed-case letters, using the sentence style.
- Position descriptions to the right of the button. Separate them by at least one space from the button.
- When a choice is conditionally unavailable for selection, display the choice description grayed out or dimmed.
- Include a none choice if it adds clarity.

Size

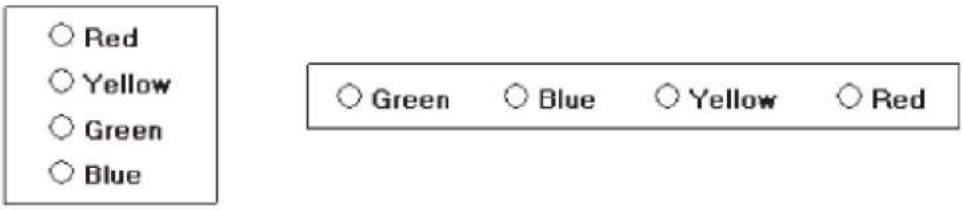
- Show a minimum of two choices, a maximum of eight.

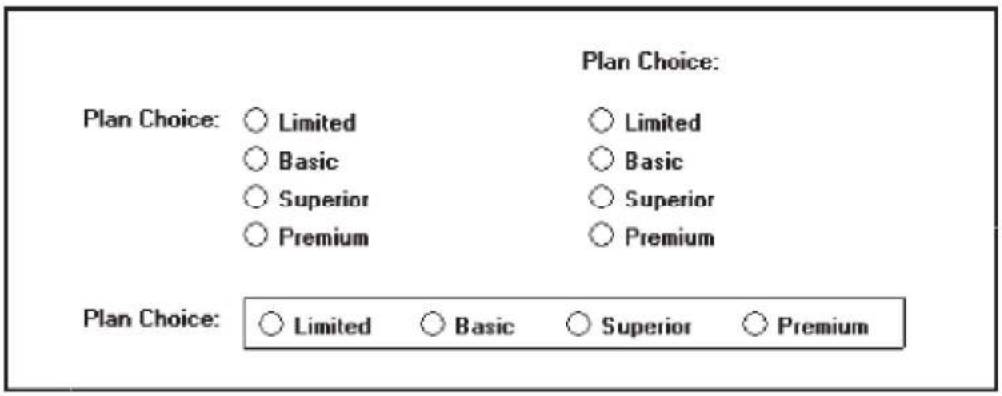
Defaults

- When the control possesses a state or affect that has been predetermined to have a higher probability of selection than the others, designate it as the default and display its button filled in.
- When the control includes choices whose states cannot be predetermined, display all the buttons without setting a dot, or in the *indeterminate* state.
- When a multiple selection includes choices whose states vary, display the buttons in another unique manner, or in the *mixed value* state.

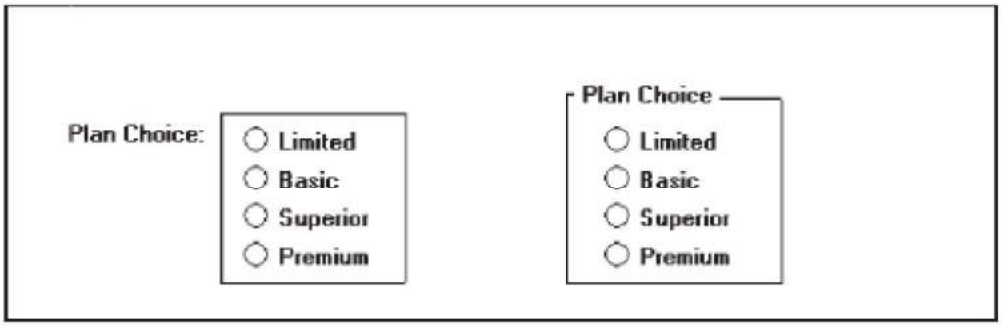
Structure

- A columnar orientation is the preferred manner of presentation.
- Left-align the buttons and choice descriptions.
- If vertical space on the screen is limited, orient the buttons horizontally.
- Provide adequate separation between choices so that the buttons are associated with the proper description.
  — A distance equal to three spaces is usually sufficient.
- Enclose the buttons in a border to visually strengthen the relationship they possess.





Still Better



Best

Organization

- Arrange selections in expected order or follow other patterns such as frequency of occurrence, sequence of use, or importance.
  — For selections arrayed top to bottom, begin ordering at the top.
  — For selections arrayed left to right, begin ordering at the left.
- If, under certain conditions, a choice is not available, display it subdued or less brightly than the available choices.

Related Control

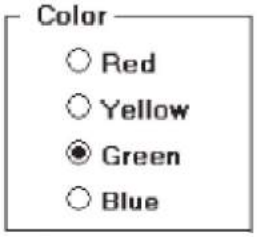
- Position any control related to a radio button immediately to the right of the choice description.
- If the radio button choice description also acts as the label for the control that follows it, end the label with an arrow (>).

Captions

- Structure:
  — Provide a caption for each radio button control.

• Exception: In screens containing only one radio button control, the screen title may serve as the caption.

- Display:
  — Fully spelled out.
  — In mixed-case letters, capitalizing the first letter of all significant words.
- Columnar orientation:
  — With a control border, position the caption:
    • Upper-left-justified within the border.

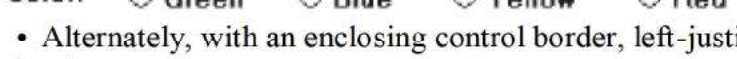


    • Alternately, the caption may be located to the left of the topmost choice description.

  — Without an enclosing control border, position the caption:
    • Left-justified above the choice descriptions, separated by one space line.

    • Alternately, the caption may be located to the left of the topmost choice description.

- Horizontal orientation:
  — Position the caption to the left of the choice descriptions.



    • Alternately, with an enclosing control border, left-justified within the border.

  — Be consistent in caption style and orientation within a screen.

Keyboard Equivalents

- Assign a keyboard mnemonic to each choice description.

- Designate the mnemonic by underlining the applicable letter in the choice description.

Selection Method and Indication

- Pointing:
  — The selection target area should be as large as possible.
    • Include the button and the choice description text.
  — Highlight the selection choice in some visually distinctive way when the cursor's resting on it and the choice is available for selection.
    • This cursor should be as long as the longest choice description plus one space at each end. Do not place the cursor over the small button.

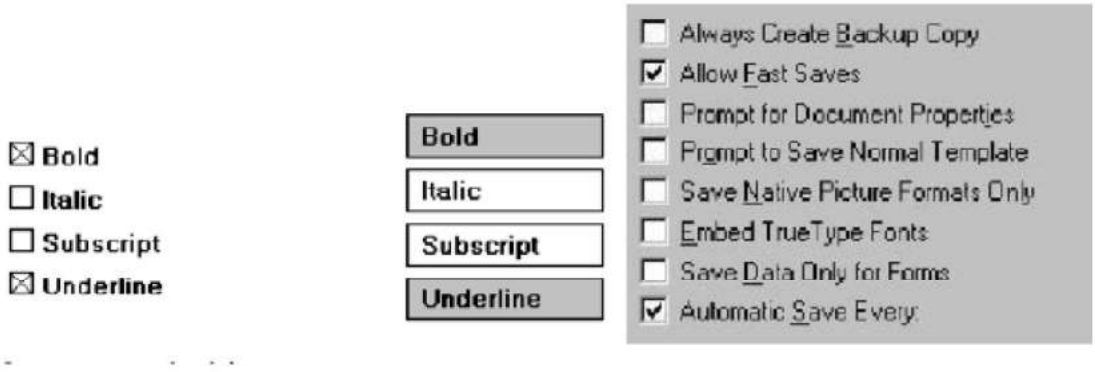
- Activation:
  — When a choice is selected, distinguish it visually from the unselected choices.
    • A radio button should be filled in with a solid dark dot or made to look depressed or higher through use of a shadow.
  — When a choice is selected, any other selected choice must be deselected.
- Defaults:
  — If a radio button control is displayed that contains a choice previously selected or a default choice, display the selected choice as set in the control

Check Boxes

- Description:
  — A two-part control consisting of a square box and choice description.
  — Each option acts as a switch and can be either ‑on‖ or ‑off.‖
    • When an option is selected (on), a mark such as an ‑X‖ or ‑check‖ appears within the square box, or the box is highlighted in some other manner.
    • Otherwise the square box is unselected or empty (off).
  — Each box can be:
    • Switched on or off independently.
    • Used alone or grouped in sets.
- Purpose:
  — To set one or more options as either on or off.
- Advantages
  — Easy-to-access choices.
  — Easy-to-compare choices.
  — Preferred by users.
- Disadvantages:

— Consume screen space.
— Limited number of choices.
— Single check boxes difficult to align with other screen controls.

- Proper usage:
  — For setting attributes, properties, or values.
  — For nonexclusive choices (that is, more than one can be selected).
  — Where adequate screen space is available.
  — Most useful for data and choices that are:
    • Discrete.
    • Small and fixed in number.
    • Not easily remembered.
    • In need of a textual description to describe meaningfully.
    • Most easily understood when the alternatives can be seen together and compared to one another.
    • Never changed in content.
  — Can be used to affect other controls.
  — Use only when both states of a choice are clearly opposite and unambiguous.



Choice Descriptions

- Provide meaningful, fully spelled-out choice descriptions clearly describing the values or effects set by the check boxes.
- Display them in a single line of text.
- Display them using mixed-case letters in sentence style.
- Position descriptions to the right of the check box. Separate by at least one space from the box.
- When a choice is unavailable for selection under a certain condition, display the choice description visually dimmed.

Size

- Show a minimum of one choice, a maximum of eight.

Defaults

- When the control possesses a state or affect that has been preset, designate it as the default and display its check box marked.
- When a multiple selection includes choices whose states vary, display the buttons in another unique manner, or the *mixed value* state.

Structure

- Provide groupings of related check boxes.
- A columnar orientation is the preferred manner of presentation for multiple related check boxes.
- Left-align the check boxes and choice descriptions.
- If vertical space on the screen is limited, orient the boxes horizontally.
- Provide adequate separation between boxes so that the buttons are associated with the proper description.
  — A distance equal to three spaces is usually sufficient.
- Enclose the boxes in a border to visually strengthen the relationship they possess.

Organization

- Arrange selections in logical order or follow other patterns such as frequency of occurrence, sequence of use, or importance.
  — For selections arrayed top to bottom, begin ordering at the top.
  — For selections arrayed left to right, begin ordering at the left.
- If, under certain conditions, a choice is not available, display it subdued or less brightly than the available choices.

Related Control

- Position any control related to a check box immediately to the right of the choice description.
  — If a the check box choice description also acts as the label for the control that follows it , end the label with an arrow (>).

Captions and Keyboard Equivalents

Same as Radio Button

Selection Method and Indication

- Pointing:
  — The selection target area should be as large as possible.
    • Include the check box and the choice description text.
  — Highlight the selection choice in some visually distinctive way when the cursor's resting on it and the choice is available for selection.
    • This cursor should be as long as the longest choice description plus one space at each end. Do not place the cursor over the check box.

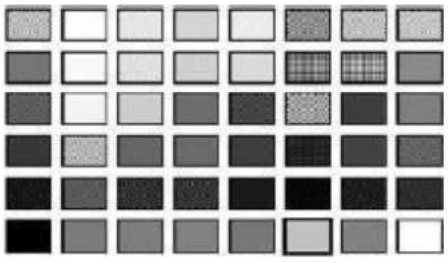
- Activation:
  — When a choice is selected, distinguish it visually from the non-selected choices.
    - A check box should be filled in or made to look depressed or higher through use of a shadow.
- Defaults:
  — If a check box is displayed that contains a choice previously selected or default choice, display the selected choice as set in the control.
- Select/deselect all:
  — Do not use *Select All* and *Deselect All* check boxes.
- Mixed-value state:
  — When a check box represents a value, and a multiple selection encompasses multiple value occurrences set in both the on and off state, display the check box in a *mixed value* state.

    - Fill the check box with another easily differentiable symbol or pattern.

  — Toggle the check box as follows:
    - Selection 1: Set the associated value for all elements. Fill the check box with an -X‖ or -check.‖
    - Selection 2: Unset the value for all associated elements. Blank-out the check box.
    - Selection 3: Return all elements to their original state. Fill the check box with the mixed value symbol or pattern.

Palettes

- Description:
  — A control consisting of a series of graphical alternatives. The choices themselves are descriptive, being composed of colors, patterns, or images.
  — In addition to being a standard screen control, a palette may also be presented on a pull-down or pop-up menu or a toolbar.



- Purpose:
  — To set one of a series of mutually exclusive options presented graphically or pictorially.
- Advantages:
  — Pictures aid comprehension.

— Easy-to-compare choices.
— Usually consume less screen space than textual equivalents.

- Disadvantages:
  — A limited number of choices can be displayed.
  — Difficult to organize for scanning efficiency.
  — Requires skill and time to design meaningful and attractive graphical representations.
- Proper usage:
  — For setting attributes, properties, or values.
  — For mutually exclusive choices (that is, only one can be selected).
  — Where adequate screen space is available.
  — Most useful for data and choices that are:
    • Discrete.
    • Frequently selected.
    • Limited in number.
    • Variable in number.
    • Not easily remembered.
    • Most easily understood when the alternatives may be seen together and compared to one another.
    • Most meaningfully represented pictorially or by example.
    • Can be clearly represented pictorially.
    • Rarely changed in content.
  — Do not use:
    • Where the alternatives cannot be meaningfully and clearly represented pictorially.
    • Where words are clearer than images.
    • Where the choices are going to change.

Graphical Representations

- Provide meaningful, accurate, and clear illustrations or representations of choices.
- Create images large enough to:
  — Clearly illustrate the available alternatives.
  — Permit ease in pointing and selecting.
- Create images of equal size.
- Always test illustrations before implementing them.

Size

- Present all available alternatives within the limits imposed by:
  — The size of the graphical representations.
  — The screen display's capabilities.

Layout

- Create boxes large enough to:
  — Effectively illustrate the available alternatives.
  — Permit ease in pointing and selecting.

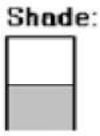
- Create boxes of equal size.
- Position the boxes adjacent to, or butted up against, one another.
- A columnar orientation is the preferred manner.
- If vertical space on the screen is limited, orient the choices horizontally.

Organization

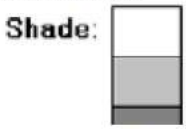
- Arrange palettes in expected or normal order.
  — For palettes arrayed top to bottom, begin ordering at the top.
  — For palettes arrayed left to right, begin ordering at the left.
- If an expected or normal order does not exist, arrange choices by frequency of occurrence, sequence of use, importance, or alphabetically (if textual).
- If, under certain conditions, a choice is not available, display it subdued or less brightly than the other choices.

Captions

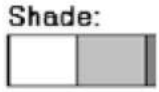
- Provide a caption for each palette.
  — On screens containing only one palette, the screen title may serve as the caption.
- Display the caption fully spelled out using mixed-case letters.
- Columnar orientation:
  — The field caption may be positioned left-aligned above the palette.



  — Alternately, the caption may be positioned to the left of the topmost alternative.



- Horizontal orientation:
  — The field caption may be positioned above the palette.



  — Alternately, the caption may be positioned to the left of the alternatives.

Selection Method and Indication

- Pointing:
  — Highlight the choice in some visually distinctive way when the pointer or cursor is resting on it and the choice is available for selection.
- Activation:

— When a choice is selected, distinguish it visually from the unselected choices by highlighting it in a manner different from when it is pointed at, or by placing a bold border around it.

- Defaults:
  — If a palette is displayed with a choice previously selected or a default choice, display the currently active choice in the manner used when it was selected.

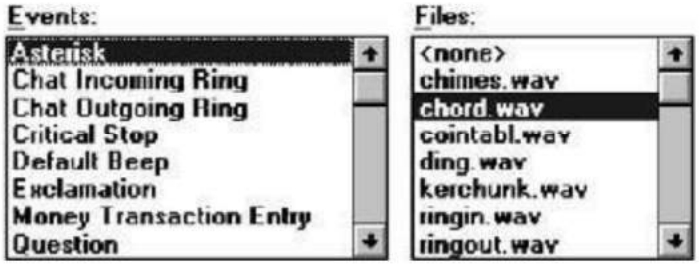
List Boxes

- Description:
  — A permanently displayed box-shaped control containing a list of attributes or objects from which:
    - A single selection is made (mutually exclusive), or
    - Multiple selections are made (non-mutually-exclusive).

  — The choice may be text, pictorial representations, or graphics.
  — Selections are made by using a mouse to point and click.
  — Capable of being scrolled to view large lists of choices.
  — No text entry field exists in which to type text.
  — A list box may be may be associated with a *summary list box* control, which allows the selected choice to be displayed or an item added to the list.
- Purpose:
  — To display a collection of items containing:
    - Mutually exclusive options.
    - Non-mutually-exclusive options.
- Advantages:
  — Unlimited number of choices.
  — Reminds users of available options.
  — Box always visible.
- Disadvantages:
  — Consumes screen space.
  — Often requires an action (scrolling) to see all list choices.
  — The list content may change, making it hard to find items.
  — The list may be ordered in an unpredictable way, making it hard to find items.
- Proper usage:
  — For selecting values or setting attributes.
  — For choices that are:
    - Mutually exclusive (only one can be selected).
    - Non-mutually-exclusive (one or more may be selected).

  — Where screen space is available.
  — For data and choices that are:
    - Best represented textually.
    - Not frequently selected.
    - Not well known, easily learned, or remembered.
    - Ordered in an unpredictable fashion.
    - Frequently changed.
    - Large in number.
    - Fixed or variable in list length.

— When screen space or layout considerations make radio buttons or check boxes impractical.



List Box General Guidelines

Selection Descriptions

- Clearly and meaningfully describe the choices available. Spell them out as fully as possible.
  — Graphical representations must clearly represent the options.
- Present in mixed case, using the sentence style structure.
- Left-align into columns.

List Size

- Not actual limit in size.
- Present all available alternatives.
- Require no more than 40 page-downs to search a list.
  — If more are required, provide a method for using search criteria or scoping the options.

Box Size

- Must be long enough to display 6 to 8 choices without requiring scrolling.
  — Exceptions:
  • If screen space constraints exist, the box may be reduced in size to display at least three items.
  • If it is the major control within a window, the box may be larger.
  — If more items are available than are visible in the box, provide vertical scrolling to display all items.
- Must be wide enough to display the longest possible choice.

— When box cannot be made wide enough to display the longest entry:
• Make it wide enough to permit entries to be distinguishable, or,
• Break the long entries with an ellipsis (...) in the middle, or,
• Provide horizontal scrolling.

Organization

- Order in a logical and meaningful way to permit easy browsing.
  — Consider using separate controls to enable the user to change the sort order or filter items displayed in the list.
- If a particular choice is not available in the current context, omit it from the list.
  — Exception: If it is important that the existence and unavailability of a particular list item be communicated, display the choice dimmed or grayed out instead of deleting it.

Layout and Separation

- Enclose the choices in a box with a solid border.
  — The border should be the same color as the choice descriptions.
- Leave one blank character position between the choice descriptions and the left border.
- Leave one blank character position between the longest choice description in the list and the right border, if possible.

Captions

- Use mixed-case letters.
- The preferred position of the control caption is above the upper-left corner of the list box.

- Alternately, the caption may be located to the left of the topmost choice description.

- Be consistent in caption style and orientation within a screen, and related screens.

Disabling

- When a list box is disabled, display its caption and show its entries as grayed out or dimmed.

Selection Method and Indication

- Pointing:
  — Highlight the selection choice in some visually distinctive way when the pointer or cursor is resting on it and the choice is available for selection.
- Selection:
  — Use a reverse video or reverse color bar to surround the choice description when it is selected.
  — The cursor should be as wide as the box itself.

  — Mark the selected choice in a distinguishing way.
- Activation:
  — Require the pressing of a command button when an item, or items, is selected.

Single-Selection List Boxes

- Purpose:
  — To permit selection of only one item from a large listing.
- Design guidelines:
  — Related text box
  • If presented with an associated text box control:
  — Position the list box below and as close as possible to the text box.
  — The list box caption should be worded similarly to the text box caption.

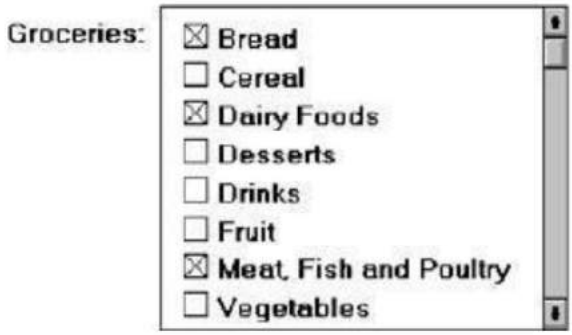
— If the related text box and the list box are very close in proximity, the caption may be omitted from the list box.

— Use the same background color for the text box as is used in the list box.

- Defaults:
  • When the list box is first displayed:
  — Present the currently active choice highlighted or marked with a circle or d diamond to the left of the entry.
  — If a choice has not been previously selected, provide a default choice and display it in the same manner that is used in selecting it.
  — If the list represents mixed values for a multiple selection, do not highlight an entry.
- Other:
  —Follow other relevant list box guidelines.

Extended and Multiple-Selection List Boxes

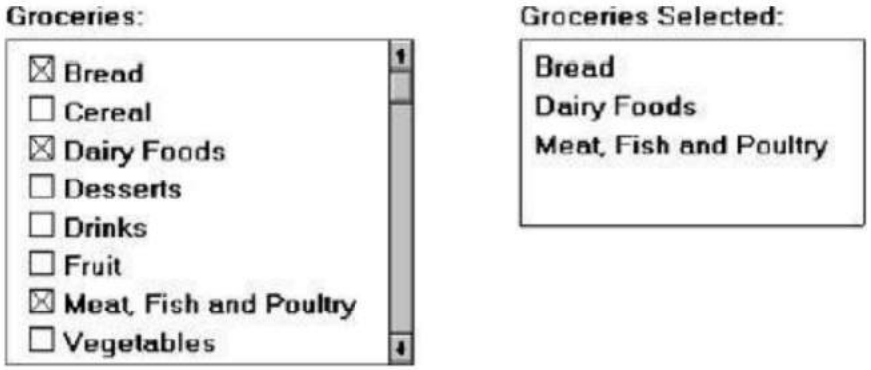
- Purpose:
  — To permit selection of more than one item in a long listing.
    • Extended list box: Optimized for individual item or range selection.
    • Multiple-selection list box: Optimized for independent item selection.
- Design guidelines:
  — Selection indication:
    • Mark the selected choice with an X or check mark to the left of the entry.



• Consider providing a *summary list box*.

— Position it to the right of the list box.

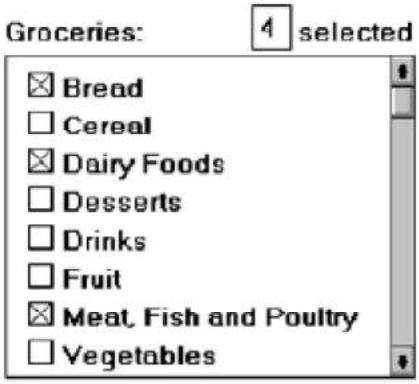
— Use the same colors for the summary list box as are used in the list box.



— Provide command buttons to *Add* (one item) or *Add All* (items) to the summary list box, and *Remove* (one item) or *Remove All* (items) from the summary list box.

- Consider providing a display-only text control indicating how many choices have been selected.

— Position it justified upper-right above the list box.



— Select all and Deselect All buttons

— Provide command buttons to accomplish fast *Select All* and *Deselect All* actions, when these actions must be frequently or quickly performed.

— Defaults:

• When the list box is first displayed:

— Display the currently active choices highlighted.

— Mark with an X or check mark to the left of the entry.

— If the list represents mixed values for a multiple selection, do not highlight an entry.

— Other:

• Follow other relevant list box guidelines.

List View Controls

- Description:
  — A special extended-selection list box that displays a collection of items, consisting of an icon and a label.
  — The contents can be displayed in four different views:
  • Large Icon: Items appear as a full-sized icon with a label below.
  • Small Icon: Items appear as a small icon with label to the right.
  • List: Items appear as a small icon with label to the right.
  — Arrayed in a columnar, sorted layout.
  • Report: Items appear as a line in a multicolumn format.
  — Leftmost column includes icon and its label.
  — Subsequent columns include application-specific information.
- Purpose and usage:
  — Where the representation of objects as icons is appropriate.
  — To represent items with multiple columns of information.

Drop-down/Pop-up List Boxes

- Description
  — A single rectangular control that shows one item with a small button to the right side.
  • The button provides a visual cue that an associated selection box is available but hidden.
  — When the button is selected, a larger associated box appears, containing a list of choices from which one may be selected.
  — Selections are made by using the mouse to point and click.
  — Text may not be typed into the control.
- Purpose:
  — To select one item from a large list of mutually exclusive options when screen space is limited.
- Advantages:
  — Unlimited number of choices.
  — Reminds users of available options.
  — Conserves screen space.
- Disadvantages:
  — Requires an extra action to display the list of choices.
  — When displayed, all choices may not always be visible, requiring scrolling.
  — The list may be ordered in an unpredictable way, making it hard to find items.
- Proper usage:
  — For selecting values or setting attributes.
  — For choices that are mutually exclusive (only one can be selected).
  — Where screen space is limited.

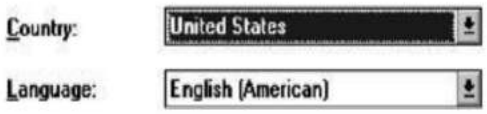
— For data and choices that are:
  - Best represented textually.
  - Infrequently selected.
  - Not well known, easily learned, or remembered.
  - Ordered in a unpredictable fashion.
  - Large in number.
  - Variable or fixed in list length.

— Use drop-down/pop-up lists when:
  - Screen space or layout considerations make radio buttons or single-selection list boxes impractical.
  - The first, or displayed, item will be selected most of the time.

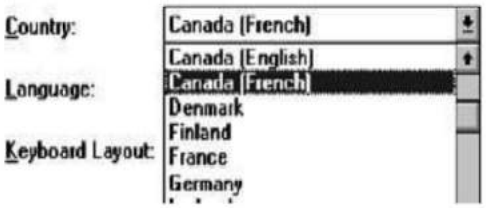
— Do not use a drop-down list if it important that all options be seen together.

Drop Down List Box

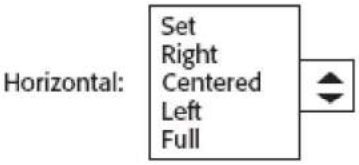
Before selection



After Selection



Pop Up List Box



Prompt Button

- Provide a visual cue that a box is hidden by including a downward pointing arrow, or other meaningful image, to the right side of the selection field.
  — Position the button directly against, or within, the selection field.

Selection Descriptions

- Clearly and meaningfully describe the choices available. Spell them out as fully as possible.
  — Graphical representations must clearly represent the options.
  — Left-align them in columns.
  — Display the descriptions using mixed-case letters.

List Size

- Not limited in size.

- Present all available alternatives.

Box Size

- Long enough to display 6 to 8 choices without scrolling.
  — If more than eight choices are available, provide vertical scrolling to display all items.
- Wide enough to display the longest possible choice.
- When a box cannot be made wide enough to display the longest entry:
  — Make it wide enough to permit entries to be distinguishable, or,
  — Break long entries with ellipses (...) in the middle, or,
  — Provide horizontal scrolling.

Organization

- Order in a logical and meaningful way to permit easy browsing.
- If a particular choice is not available in the current context, omit it from the list.
  — Exception: If it is important that the existence and unavailability of a particular list item be communicated, display the choice dimmed or grayed out instead of deleting it.

Layout and Separation

- Enclose the choices in a box composed of a solid line border.
  — The border should be the same color as the choice descriptions.
  — Leave one blank character position between the choices and the left border.
  — Leave one blank character position between the longest choice description in the list and the right border, if possible.

Captions

- Display using mixed-case letters.
- Position the caption to the left of the box.
  — Alternately, it may be positioned left-justified above the box.

Defaults

- When the drop-down/pop-up listing is first presented, display the currently set value.
- If a choice has not been previously selected, provide a default choice.

Disabling

- When a drop-down/pop-up list box is disabled, display its caption and entries as disabled or dimmed.

Selection Method and Indication

- Pointing:
  — Highlight the selection choice in some visually distinctive way when the pointer or cursor is resting on it and the choice is available for selection.
- Activation:
  — Close the drop-down/pop-up list box when an item is selected.////

Custom Controls

- Implement custom controls with caution.
- If used, make the look and behavior of custom controls different from that of standard controls.

Presentation Controls

- Common presentation controls are *static text fields, group boxes column headings, ToolTips, balloon tips,* and *progress indicators*.

Static Text Fields

- Description:
  — Read-only textual information.
- Purpose:
  — To identify a control by displaying a control caption.
  — To clarify a screen by providing instructional or prompting information.
  — To present descriptive information.
- Proper usage:
  — To display a control caption.
  — To display instructional or prompting information.
  — To display descriptive information.

Static Text Field Guidelines

- Captions:
  — Include a colon (:) as part of the caption.
  — Include a mnemonic for keyboard access.
  — When the associated control is disabled, display it dimmed.
  — Follow all other presented guidelines for caption presentation and layout.

- Instructional or prompting information:
  — Display it in a unique and consistent font style for easy recognition and differentiation.
  — Follow all other presented guidelines for prompting and instructional information.
- Descriptive information:
  — Follow all other guidelines for required screen or control descriptive information.

Group Boxes

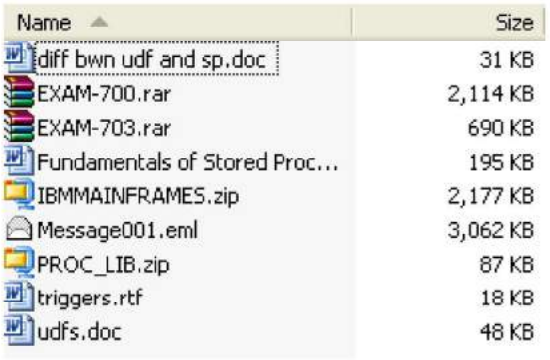
- Description:
  — A rectangular frame that surrounds a control or group of controls.
  — An optional caption may be included in the frame's upper-left corner.
- Purpose:
  — To visually relate the elements of a control.
  — To visually relate a group of related controls.
- Proper usage:
  — To provide a border around radio button or check box controls.
  — To provide a border around two or more functionally related controls.
- Guidelines:
  — Label or heading:
    - Typically, use a noun or noun phrase for the label or heading.
    - Provide a brief label or heading, preferably one or two words.
    - Relate label or heading's content to the group box's content.
    - Capitalize the first letter of each significant word.
    - Do not include and ending colon (:).
  — Follow all other guidelines presented for control and section borders.

Column Headings

- Description:
  — Read-only textual information that serves as a heading above columns of text or numbers.
  — Can be divided into two or more parts.
- Purpose:
  — To identify a column of information contained in a table.
- Proper usage:
  — To display a heading above a column of information contained in a table.
- Guidelines:
  — Heading:
    - Provide a brief heading.
    - Can include text and a graphic image.

- Capitalize the first letter of each significant word.
- Do not include an ending colon (:).

— The width of the column should fit the average size of the column entries.
— Does not support keyboard access.



ToolTips

- Description:
  — A small pop-up window containing descriptive text that appears when a pointer is moved over a control or element either:
    - Not possessing a label.
    - In need of additional descriptive or status information.
- Purpose:
  — To provide descriptive information about a control or screen element.
- Advantages:
  — Identifies an otherwise unidentified control.
  — Reduces possible screen clutter caused by control captions and descriptive information.
  — Enables control size to be reduced.
- Disadvantages:
  — Not obvious, must be discovered.
  — Inadvertent appearance can be distracting.
- Proper usage:
  — To identify a control that has no caption.
  — To provide additional descriptive or status information about a screen element.

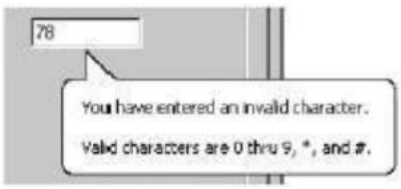
ToolTip Guidclincs

- Display after a short time-out.
- For toolbars, provide a brief word as a label.
  — Use mixed case in the headline style of presentation with no ending punctuation.
- For other elements, provide a brief phrase presenting descriptive or status information.
  — Use mixed case in the sentence style of presentation.
- Present ToolTips at the lower-right edge of the pointer.
  — Display them fully on the screen.

— For text boxes, display ToolTips centered under the control.
- Display them in the standard system ToolTip colors.
- Remove the ToolTip when the control is activated or the pointer is moved away.
- Don't substitute ToolTips for good design.

Balloon Tips

- Description:
  — A small pop-up window that contains information in a word balloon.
  — Components can include:
    • Title.
    • Body text.
    • Message Icons.
  — Appear adjacent to the item to which they apply, generally above or to left.
  — Only one tip, the last posted, is visible at any time.
  — Tips are removed after a specified time period.
- Purpose:
  — To provide additional descriptive or status information about a screen element.
- Advantages:
  — Provides useful reminder and status information.
- Disadvantages:
  — If overused they lose their attention-getting value.
  — If overused in situations the user considers not very important, their continual appearance can be aggravating.
- Proper usage:
  — To display noncritical:
    • Reminder information.
    • Notification information.
  — Do not use tips to display critical information.



Balloon Tip Guidelines

- General:
  — Use a notification tip to inform the user about state changes.
  — Use a reminder tip for state changes that the user might not usually notice.
  — Point the tip of the balloon to the item it references.

— Do not use them to replace ToolTips.
— Do not overuse balloon tips.

- Content:
  — Restrict them to a length of 100 characters, including title and body text.
  — Title text should:
    • If the tip refers to an icon or other image representing a specific object, include:
      — The object's name, using its normal capitalization.
      — The object's status, using sentence-style presentation without ending punctuation.
    • Be presented in bold.
  — Body text should:
    • Include a description of the situation in one or two brief sentences.
    • Include a brief suggestion for correcting the situation.
    • Be presented using mixed-case in the sentence style.

Progress Indicators

- Description:
  — A rectangular bar that fills as a process is being performed, indicating the percentage of the process that has been completed.
- Purpose:
  — To provide feedback concerning the completion of a lengthy operation.
- Proper usage:
  — To provide an indication of the proportion of a process completed.

Progress Indicator Guidelines

- When filling the indicator:
  — If horizontally arrayed, fill it from left to right.
  — If vertically arrayed, fill it from bottom to top.
- Fill it with a color or a shade of gray.
- Include descriptive text for the process, as necessary.
- Place text outside of the control.

Sample Box

- Description:

— A box illustrating what will show up on the screen based upon the parameteror parameters selected.
— May include text, graphics, or both.

- Purpose:
  — To provide a representation of actual screen content based upon the parameteror parameters selected.
- Guidelines:
  — Include a brief label.
  — Use mixed case in the headline style.
  — Locate it adjacent to the controls upon which it is dependent.

Scrolling Tickers

- Description:
  — Text that scrolls horizontally through a container window.
- Advantages:
  — Consume less screen space than full text.
  - Disadvantages:
    — Hard to read.
  — Time-consuming to interpret.
  — Distracting.
- Guideline:
  — Do not use.///

Test, Test, and Retest

Testing steps to be reviewed are:

- Identifying the purpose and scope of testing.
- Understanding the importance of testing.

- Developing a prototype.
- Developing the right kind of test plan.
- Designing a test to yield relevant data.
- Soliciting, selecting, and scheduling users to participate.
- Providing the proper test facility.
- Conducting tests and collecting data.
- Analyzing the data and generating design recommendations.
- Modifying the prototype as necessary.
- Testing the system again.
- Evaluating the working system.

The Purpose of Usability Testing

- First, it establishes a communication bridge between developers and users. Through testing, the developer learns about the user's goals, perceptions, questions, and problems.
- Second, testing is used to evaluate a product. It validates design decisions. It also can identify potential problems in design at a point in the development process where they can be more easily addressed.

The Importance of Usability Testing

A thorough usability testing process is important for many reasons,

- Developers and users possess different models.
- Developer's intuitions are not always correct.
- There is no average user.
- It's impossible to predict usability from appearance.
- Design standards and guidelines are not sufficient.
- Informal feedback is inadequate.
- Problems found late are more difficult and expensive to fix.
- Advantages over a competitive product can be achieved.

Scope of Testing

- Testing should begin in the earliest stages of product development and continue throughout the development process.
- It should include as many of the user's tasks, and as many of the product's components, as reasonably possible.

Prototypes

- A prototype is primarily a vehicle for exploration, communication, and evaluation. Its purpose is to obtain user input in design, and to provide feedback to designers.
- A prototype is a simulation of an actual system that can be quickly created.

- A prototype may be a rough approximation, such as a simple hand-drawn sketch, or it may be interactive, allowing the user to key or select data using controls, navigate through menus, retrieve displays of data, and perform basic system functions.
- A prototype may have great breadth, including as many features as possible to present concepts and overall organization, or it might have more depth, including more detail on a given feature or task to focus on individual design aspects.

Hand Sketches and Scenarios

- Description:
  — Screen sketches created by hand.
  — Focus is on the design, not the interface mechanics.
  — A low-fidelity prototype.
- Advantages:
  — Can be used very early in the development process.
  — Suited for use by entire design team.
  — No large investment of time and cost.
  — No programming skill needed.
  — Easily portable.
  — Fast to modify and iterate.
  — A rough approximation often yields more substantive critical comments.
  — Easier to comprehend than functional specifications.
  — Can be used to define requirements.
- Disadvantages:
  — Only a rough approximation.
  — Limited in providing an understanding of navigation and flow.
  — A demonstration, not an exercise.
  — Driven by a facilitator, not the user.
  — Limited usefulness for a usability test.
  — A poor detailed specification for writing the code.
  — Usually restricted to most common tasks.
- ***Sketch Creation Process***
  - Sketch (storyboard) the screens while determining:
    — The source of the screen's information.
    — The content and structure of individual screens.
    — The overall order of screens and windows.
  - Use an erasable medium.
  - Sketch the screens needed to complete each workflow task.
  - Try out selected metaphors and change them as necessary.
  - First, storyboard common/critical/frequent scenarios.
    — Follow them from beginning to end.
    — Then, go back and build in exceptions.
  - Don't get too detailed; exact control positioning is not important, just overall order and flow.
  - Storyboard as a team, including at least one user.
  - Only develop online prototypes when everyone agrees that a complete set of screens has been satisfactorily sketched.

Interactive Paper Prototypes

- Description:
  — Interface components (menus, windows, and screens) constructed of common paper technologies (Post-It notes, transparencies, and so on).
  — The components are manually manipulated to reflect the dynamics of the software.
  — A low-fidelity prototype.
- Advantages:
  — More illustrative of program dynamics than sketches.
  — Can be used to demonstrate the interaction.
  — Otherwise, generally the same as for hand-drawn sketches and scenarios.
- Disadvantages:
  — Only a rough approximation.
  — A demonstration, not an exercise.
  — Driven by a facilitator, not the user.
  — Limited usefulness for usability testing.

Programmed Facades

- Description:
  — Examples of finished dialogs and screens for some important aspects of the system.
  — Created by prototyping tools.
  — Medium-fidelity to high-fidelity prototypes.
- Advantages:
  — Provide a good detailed specification for writing code.
  — A vehicle for data collection.
- Disadvantages:
  — May solidify the design too soon.
  — May create the false expectation that the ‑real thing‖ is only a short time away.
  — More expensive to develop.
  — More time-consuming to create.
  — Not effective for requirements gathering.
  — Not all of the functions demonstrated may be used because of cost, schedule limitations, or lack of user interest.
  — Not practical for investigating more than two or three approaches.

Prototype-Oriented Languages

- Description:
  — An example of finished dialogs and screens for some important aspects of the system.
  — Created through programming languages that support the actual programming process.
  — A high-fidelity prototype.
- Advantages:

— May include the final code.
— Otherwise, generally the same as those of programmed facades.

- Disadvantages:
  — Generally the same as for programmed facades.

Kinds of Tests

A test is a tool that is used to measure something. The ‑something‖ may be:

- Conformance with a requirement.
- Conformance with guidelines for good design.
- Identification of design problems.
- Ease of system learning.
- Retention of learning over time.
- Speed of task completion.
- Speed of need fulfillment.
- Error rates.
- Subjective user satisfaction.

Guidelines Review

- Description:
  — A review of the interface in terms of an organization's standards and design guidelines.
- Advantages:
  — Can be performed by developers.
  — Low cost.
  — Can identify general and recurring problems
  — Particularly useful for identifying screen design and layout problems.
- Disadvantages:
  — May miss severe conceptual, navigation, and operational problems.

Heuristic Evaluation

- Description:
  — A detailed evaluation of a system by interface design specialists to identify problems.
- Advantages:
  — Easy to do.
  — Relatively low cost.
  — Does not waste user's time.
  — Can identify many problems.
- Disadvantages:
  — Evaluators must possess interface design expertise.
  — Evaluators may not possess an adequate understanding of the tasks and user communities.
  — Difficult to identify system wide structural problems.
  — Difficult to uncover missing exits and interface elements.

— Difficult to identify the most important problems among all problems uncovered.
— Does not provide any systematic way to generate solutions to the problems uncovered.

- Guidelines:
  — Use 3 to 5 expert evaluators.
  — Choose knowledgeable people:
    - Familiar with the project situation.
    - Possessing a long-term relationship with the organization.

- ***Heuristic Evaluation Process***

  - Preparing the session:
    — Select evaluators.
    — Prepare or assemble:
      - A project overview.
      - A checklist of heuristics.
    — Provide briefing to evaluators to:
      - Review the purpose of the evaluation session.
      - Preview the evaluation process.
      - Present the project overview and heuristics.
      - Answer any evaluator questions.
      - Provide any special evaluator training that may be necessary.
  - Conducting the session:
    — Have each evaluator review the system alone.
    — The evaluator should:
      - Establish own process or method of reviewing the system.
    — provide usage scenarios, if necessary.
      - Compare his or her findings with the list of usability principles.
      - Identify any other relevant problems or issues.
      - Make at least two passes through the system.
    — Detected problems should be related to the specific heuristics they violate.
    — Comments are recorded either:
      - By the evaluator.
      - By an observer.
    — The observer may answer questions and provide hints.
    — Restrict the length of the session to no more than 2 hours.
  - After the session:
    — Hold a debriefing session including observers and design team members where:
      - Each evaluator presents problems detected and the heuristic it violated.
      - A composite problem listing is assembled.
      - Design suggestions for improving the problematic aspects of the system are discussed.
    — After the debriefing session:
      - Generate a composite list of violations as a ratings form.
      - Request evaluators to assign severity ratings to each violation.
      - Analyze results and establish a program to correct violations and deficiencies.

- ***Heuristic Evaluation Effectiveness***

- One of the earliest papers addressing the effectiveness of heuristic evaluations was by Nielsen (1992). He reported that the probability of finding a major usability problem averaged 42 percent for single evaluators in six case studies. The corresponding probability for uncovering a minor problem was only 32 percent.
- Heuristic evaluations are useful in identifying many usability problems and should be part of the testing arsenal. Performing this kind of evaluation before beginning actual testing with users will eliminate a number of design problems, and is but one step along the path toward a very usable system.

- Research based set of heuristics

1. Automate unwanted workload. • Free cognitive resources for high-level tasks. • Eliminate mental calculations, estimations, comparisons, and unnecessary thinking.
2. Reduce uncertainty. • Display data in a manner that is clear and obvious.
3. Fuse data. • Reduce cognitive load by bringing together lower-level data into a higher-level summation.
4. Present new information with meaningful aids to interpretation. • Use a familiar framework, making it easier to absorb. • Use everyday terms, metaphors, and so on.
5. Use names that are conceptually related to functions. • Context-dependent. • Attempt to improve recall and recognition.
6. Group data in consistently meaningful ways to decrease search time.
7. Limit data-driven tasks. • Reduce the time needed to assimilate raw data. • Make appropriate use of color and graphics.
8. Include in the displays only that information needed by a user at a given time. • Allow users to remain focused on critical data. • Exclude extraneous information that is not relevant to current tasks.
9. Provide multiple coding of data where appropriate.
10. Practice judicious redundancy. • To resolve the conflict between heuristics 6 and 8.

From Gerhardt-Powals (1996).

Cognitive Walkthroughs

- Description:
  — Reviews of the interface in the context of tasks users perform.
- Advantages:
  — Allow a clear evaluation of the task flow early in the design process.
  — Do not require a functioning prototype.
  — Low cost.
  — Can be used to evaluate alternate solutions.
  — Can be performed by developers.
  — More structured than a heuristic evaluation.
  — Useful for assessing –exploratory learning.‖
- Disadvantages:

— Tedious to perform.
— May miss inconsistencies and general and recurring problems.

- Guidelines:
  — Needed to conduct the walkthrough are:
    • A general description of proposed system users and what relevant knowledge they possess.
    • A specific description of one or more core or representative tasks to be performed.
    • A list of the correct actions required to complete each of the tasks.
  — Review:
    • Several core or representative tasks across a range of functions.
    • Proposed tasks of particular concern.
  — Developers must be assigned roles of:
    • Scribe to record results of the action.
    • Facilitator to keep the evaluation moving.
  — Start with simple tasks.
  — Don't get bogged down demanding solutions.
  — Limit session to 60 to 90 minutes.

Think-Aloud Evaluations

- Description:
  — Users perform specific tasks while thinking out load.
  — Comments are recorded and analyzed.
- Advantages:
  — Utilizes actual representative tasks.
  — Provides insights into the user's reasoning.
- Disadvantages:
  — May be difficult to get users to think out loud.
- Guidelines:
  — Develop:
    • Several core or representative tasks.
- Tasks of particular concern.
  — Limit session to 60 to 90 minutes.

Usability Test

- Description:
  — An interface evaluation under real-world or controlled conditions.
  — Measures of performance are derived for specific tasks.
  — Problems are identified.
- Advantages:
  — Utilizes an actual work environment.
  — Identifies serious or recurring problems.
- Disadvantages:
  — High cost for establishing facility.
  — Requires a test conductor with user interface expertise.
  — Emphasizes first-time system usage.

— Poorly suited for detecting inconsistency problems.

Classic Experiments

- Description:
  — An objective comparison of two or more prototypes identical in all aspects except for one design issue.
- Advantages:
  — Objective measures of performance are obtained.
  — Subjective measures of user satisfaction may be obtained.
- Disadvantages:
  — Requires a rigorously controlled experiment to conduct the evaluation.
  — The experiment conductor must have expertise in setting up, running, and analyzing the data collected.
  — Requires creation of multiple prototypes.
- Guidelines:
  — State a clear and testable hypothesis.
  — Specify a small number of independent variables to be manipulated.
  — Carefully choose the measurements.
  — Judiciously select study participants and carefully or randomly assign them to groups.
  — Control for biasing factors.
  — Collect the data in a controlled environment.
  — Apply statistical methods to data analysis.
  — Resolve the problem that led to conducting the experiment.

Focus Groups

- Description:
  — A discussion with users about interface design prototypes or tasks.
- Advantages:
  — Useful for:
    - Obtaining initial user thoughts.
- Trying out ideas.
—
Easy to set up and run.
  — Low cost.
- Disadvantages:
  — Requires experienced moderator.
  — Not useful for establishing:
    - How people really work.
    - What kinds of usability problems people have.
- Guidelines:
  — Restrict group size to 8 to 12.
  — Limit to 90 to 120 minutes in length.
  — Record session for later detailed analysis.

Choosing a Testing Method

- Beer, Anodenko, and Sears (1997) suggest a good pairing is cognitive walkthroughs followed by think-aloud evaluations.
- Using cognitive walkthroughs early in the development process permits the identification and correction of the most serious problems. Later, when a functioning prototype is available, the remaining problems can be identified usinga think-aloud evaluation.
- A substantial leap forward in the testing process would be the creation of a software tool simulating the behavior of people. This will allow usability tests to be performed without requiring real users to perform the necessary tasks.
- In conclusion, each testing method has strengths and weaknesses. A well-rounded testing program will use a combination of some, or all, of these methods to guarantee the usability of its created product.
- It is very important that testing start as early as possible in the design process and,continue through all developmental stages.