Topics

1. Operable Control.
2. Text Control.
3. Selection Control.
4. Custom Control.
5. Presentation Control.
6. Windows Test Prototypes.
7. Kinds of Tests.

SCREEN BASED CONTROLS

- Screen-based controls, often simply called controls and sometimes called widgets, are the elements of a screen that constitute its body.
- 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.

5.1 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.

5.1.1 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 to use and 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:

- The first resembles the control commonly found on electrical or mechanical devices and is sometimes called a pushbutton. They are often referred to as **command buttons**.
- The second style is square or rectangular in shape with an icon or graphic inside. These buttons may appear singly or be placed in groupings commonly called **button bars or toolbars.**
- The third style is square or rectangular in shape with a symbol inscribed. These buttons, specific to a platform and provided by it, are located in the borders of windows and are used to do such things as obtaining a system menu or resizing a window. They are commonly called a **symbol button.**

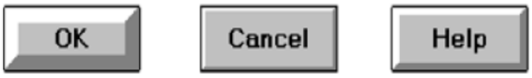


Figure 7.1 Command buttons.

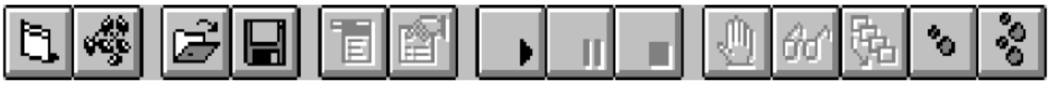


Figure 7.2 Toolbar buttons without labels.

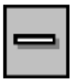


Figure 7.3 A symbol button.

5.1.1.1 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 and 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 and 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.
- Common button functions should have standard names and uses. Microsoft windows, for example, provides these standard names and definitions:
    - **OK** - Any changed information in the window is accepted and the window is closed.
    - **Cancel** - Closes window without implementing unsubmitted changes.
    - **Reset** - Resets defaults and cancels any changed information that has not been submitted.
    - **Apply** - Any changed information in the window is accepted and again displayed in the window that remains open.
    - **Close** - Closes the window.
    - **Help** - Opens online Help.

Size

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

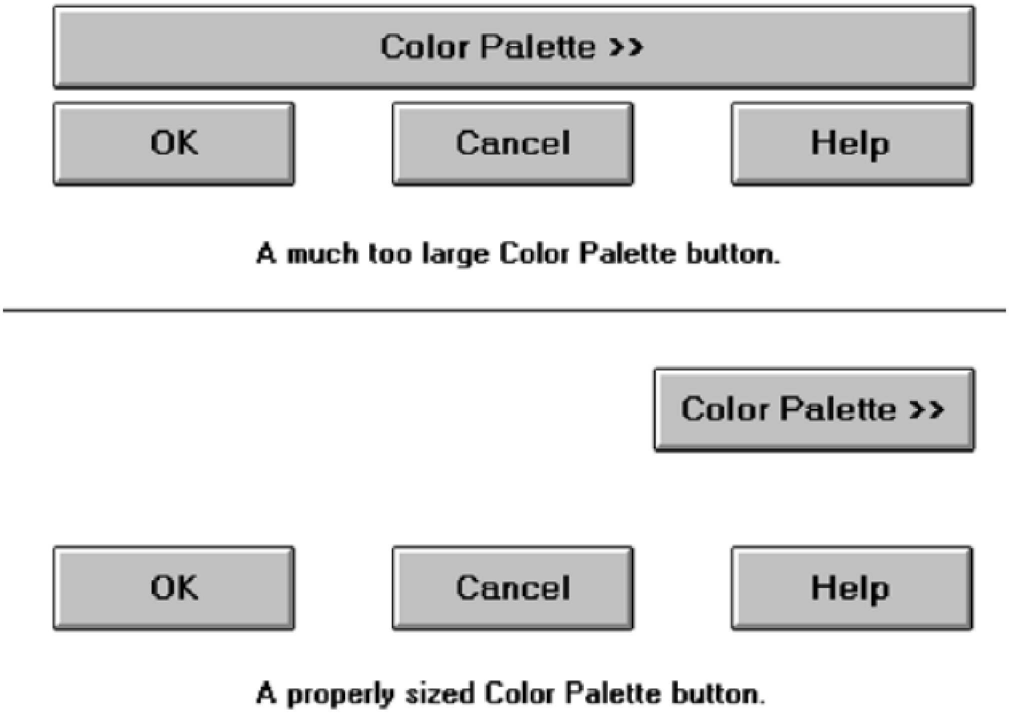

Color Palette >>
OK
Cancel
Help
A much too large Color Palette button.
Color Palette >>
OK
Cancel
Help
A properly sized Color Palette button.

Figure 7.4 Improper and proper button sizes.

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 “fit” buttons in available space.
- If buttons are for exiting the dialog then position them centered and aligned horizontally at the bottom.
- If buttons are used for invoking a dialog feature or expanding the dialog then position them centered and aligned vertically on the right side.

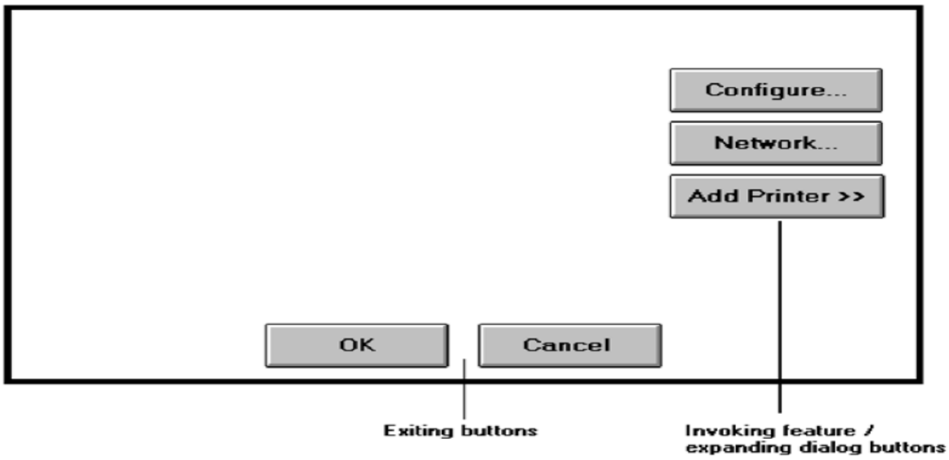
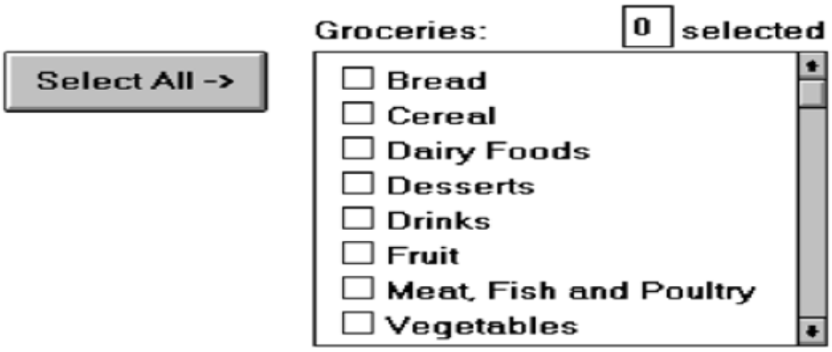


Figure 7.5 Exiting and invoking feature/expanding dialog buttons.

- If a button has a contingent relationship to another control then position it adjacent to the related control.

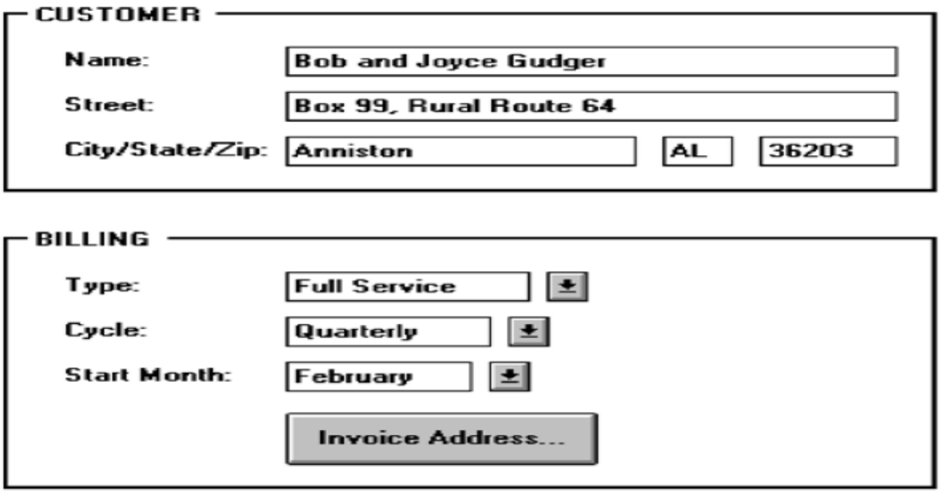


Groceries: 0 selected

□ Bread
□ Cereal
□ Dairy Foods
□ Desserts
□ Drinks
□ Fruit
□ Meat, Fish and Poultry
□ Vegetables

Figure 7.6 Button with contingent relationship to a control.

- If a button has a contingent relationship to a group of controls then position it at the bottom or to right of related controls.



CUSTOMER

Name: Bob and Joyce Gudger
Street: Box 99, Rural Route 64
City/State/Zip: Anniston AL 36203

BILLING

Type: Full Service
Cycle: Quarterly
Start Month: February

Figure 7.7 Button with contingent relationship to a grouping.

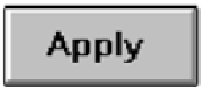
- If, due to space constraints, exiting and expanding/invoking feature buttons must be placed together and if at the bottom, place exiting buttons to the right, separating the groupings by one button's width.
- If, due to space constraints, exiting and expanding/invoking feature buttons must be placed together and 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 or 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

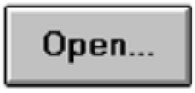
- Organize standard buttons in the manner recommended by the platform being used.
- For other buttons, organize them in common and customary grouping schemes.
  - For buttons ordered left to right, place those for most frequent actions to the left.
  - For buttons ordered top to bottom, place those for 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.
- Follow the standard, consistent ordering schemes recommended by the platform being used. 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

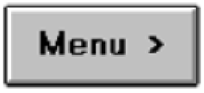
- When a button causes an action to be immediately performed, no intent indicator is necessary.



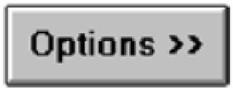
- When a button leads to a cascading dialog, include an ellipsis (...) after the label.



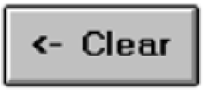
- When a button leads to a menu, include a triangle pointing in the direction the menu will appear after the label.



- 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

- When a button that expands a dialog is activated, and the dialog is expanded, display the button dimmed or grayed out.
- 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.

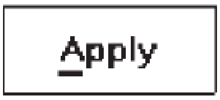
- Procedures:
    - The default can be changed as the user interacts with the window.
    - When the user navigates to a button, it can temporarily become the default.
    - Use the Enter key to activate a default button.
    - If another control requires use of the Enter key, temporarily disable the 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

- A button should visually indicate whether it is available for activation. Dim or grayout buttons for actions that are not available.

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.

Advantages of command buttons

- It is always visible, providing a reminder of its existence.
- Command buttons are conveniently and logically located in the work area and can be inscribed with meaningful descriptions of what they do.
- Their ability to assume a fairly large size speeds selection, and their three-dimensional appearance is aesthetically pleasing.
- Buttons can also provide meaningful visual feedback through the movement of the button when activated. Their activation is much easier and faster than using a two-step menu bar/pull-down sequence.

Disadvantage of command buttons

- Their larger size, which consumes considerable screen space and limits the number that can be displayed.

5.1.1.2 Toolbars

- Toolbars are compilations of commands, actions, or functions, usually graphical in structure but sometimes textual, grouped together for speedy access.
- Microsoft Windows defines a toolbar as a panel that contains a set of controls.
- 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.

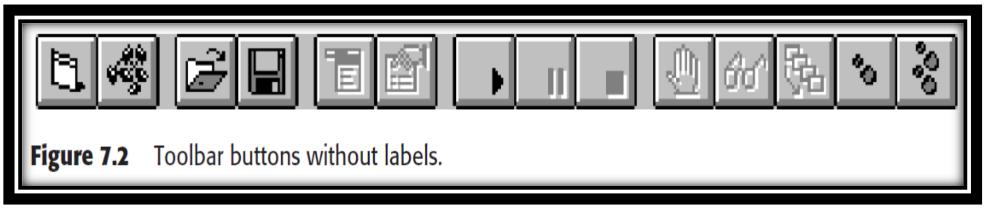


Figure 7.2 Toolbar buttons without labels.

Usage

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

Structure

- Images:
  - Provide buttons of equal size.
  - Create a meaningful and unique icon.
  - 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 subfeatures 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 and 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. Many users do not customize them, however, so a default set should always be provided.

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.

Toolbar advantages

- Toolbar buttons have continuous visibility and ease and speed of use.
- They also, individually, consume a relatively small amount of space.

Toolbar disadvantages

- Toolbar location being away from the main work area and their small size, which slows down selection.
- When a large number of buttons are grouped in a bar, they consume a great deal of screen space, and they can easily create screen clutter.
- In circumstances where they do not possess a label, the necessity of learning and remembering what they are used for can also cause problems.

5.1.1.3 Symbol Button

- The symbol when learned, identifies the button and the action to be performed when the button is selected.
- These buttons are specific to a platform and provided by it, are located in the borders of windows and are used to do such things as obtaining a system menu or resizing a window.

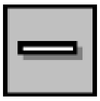


Figure 7.3 A symbol button.

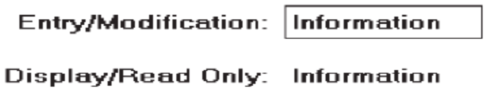
5.2 TEXT ENTRY/READ-ONLY CONTROLS

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. Historically been referred to as an entry field, It may also be referred to as an edit control.
  - Text may be displayed for read-only purposes. It is also referred to as an inquiry or display field.
- Usually possesses a caption describing the kind of information contained within it.
- An outline field border is included for enterable/editable text boxes and is not included for read-only text boxes.
- Two types of text boxes exist, Single-Line and Multiple-Line Text Boxes.
- When first displayed, the box may be blank or contain an initial value.



Entry/Modification: Information

Display/Read Only: Information

Figure 7.14 Text boxes.

Purpose

- To permit the display, entering, or editing of textual information.
- To display read-only information.

Advantages

- Text boxes are very flexible.
- Textboxes are much familiar.
- Textboxes consumes little screen space.

Disadvantages

- Textboxes require the use of typewriter keyboard.
- Textboxes require 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.
- It is preferred when using a selection list is not possible.

Single-Line and Multiple-Line Text Boxes

- Text boxes exist in two forms: single line and multiple line.
- A single-line box is used when the information contained within it can be confined to one screen line.
- Multipleline boxes are used when the information exceeds a single line. When first displayed, a text box may be blank or contain an initial value.
- **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 and 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 and 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.

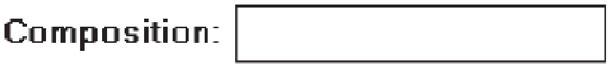


Figure 7.15

-
    -
        - 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.

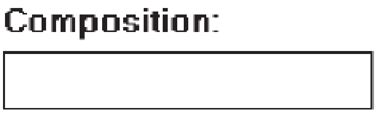


Figure 7.16

-
    - Multiple occurrence fields:
        - For entry/modification text boxes:

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

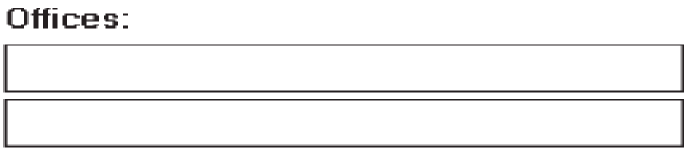
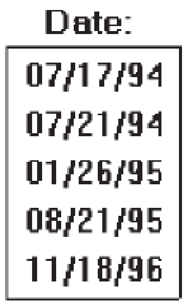


Figure 7.17

- 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.

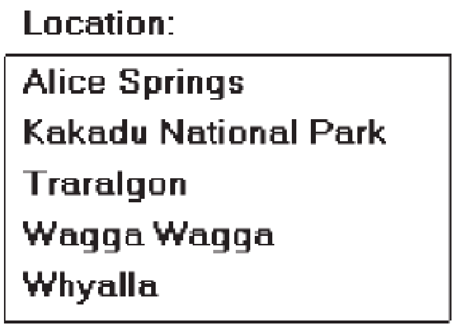


Date:

07/17/94
07/21/94
01/26/95
08/21/95
11/18/96

Figure 7.18

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

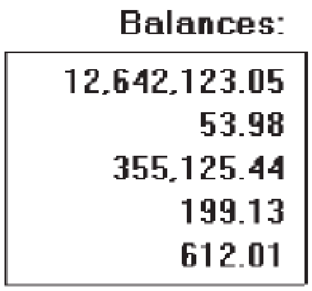


Location:

Alice Springs
Kakadu National Park
Traralgon
Wagga Wagga
Whyalla

Figure 7.19

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



Balances:

12,642,123.05
53.98
355,125.44
199.13
612.01

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.

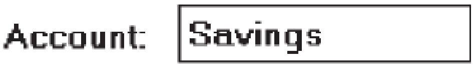

Account: Savings

Figure 7.21

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

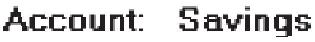

Account: Savings

Figure 7.22

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

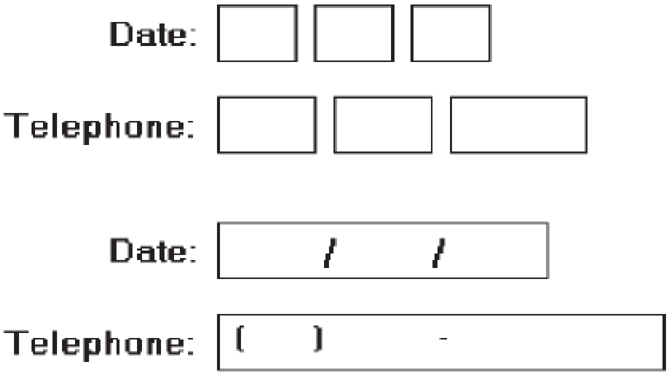


Figure 7.23

- 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.

5.3. 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.

5.3.1. Radio Buttons

- They are most often called radio buttons because of their resemblance to similar controls on radios. Microsoft Windows, however, refers to these controls as option buttons. One common display method consists of a circle associated with each choice description
- Microsoft Windows uses a small depressed circle that contains a small dot when selected. Other presentation methods include small circular buttons that look recessed when not selected and are raised when selected, and small diamond-shaped buttons that look raised when not selected and depressed when selected. Examples of radio buttons are illustrated in Figures 7.24 and 7.25.

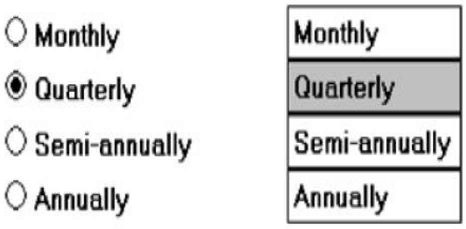


Figure 7.24 Radio buttons.

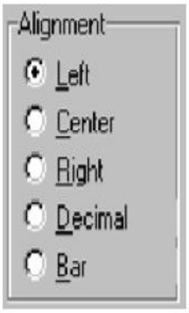


Figure 7.25 Radio buttons.

- **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 and Preferred by users.
- **Disadvantages**: Consume screen space and 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 and 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 or 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. Displaying more than eight is usually not efficient, wasting screen space. If the number of choices exceeds this maximum, consider using a single selection list box or a dropdown list box.

- **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.

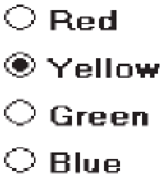


Figure 7.26

  - 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.

● Green
○ Blue
○ Yellow
○ Red

Figure 7.27

  - Enclose the buttons in a border to visually strengthen the relationship they possess.

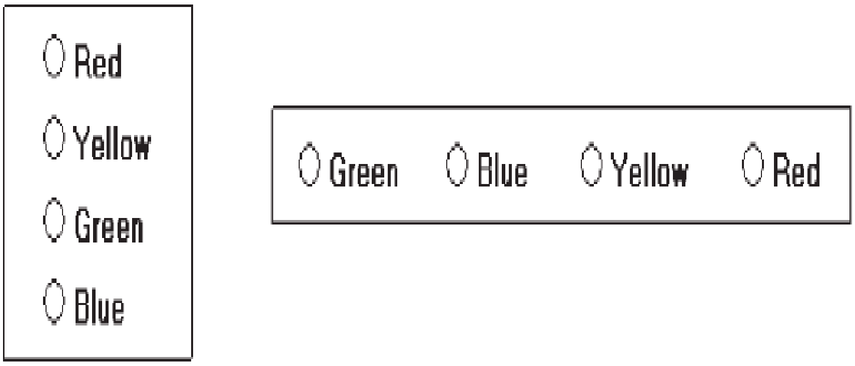
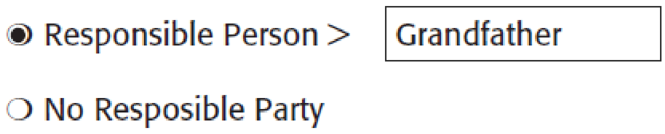


Figure 7.28

- **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 (>).

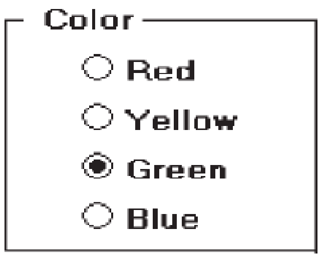


● Responsible Person > Grandfather

○ No Resposible Party

Figure 7.30

- **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.



Color

○ Red

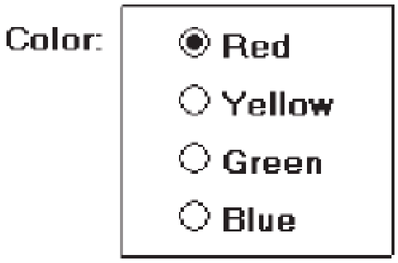
○ Yellow

● Green

○ Blue

Figure 7.31

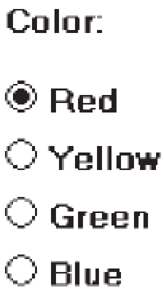
- Alternately, the caption may be located to the left of the topmost choice description.



Color: ● Red
○ Yellow
○ Green
○ Blue

Figure 7.32

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

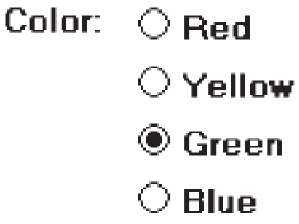


Color:

● Red
○ Yellow
○ Green
○ Blue

Figure 7.33

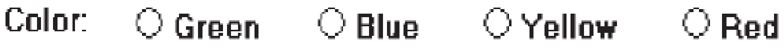
  - Alternately, the caption may be located to the left of the topmost choice description.



Color: ○ Red
○ Yellow
● Green
○ Blue

Figure 7.34

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



Color: ○ Green ○ Blue ○ Yellow ○ Red

Figure 7.35

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

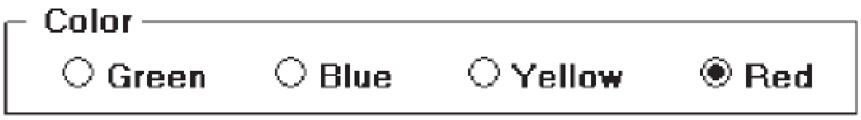


Figure 7.36

    - 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.

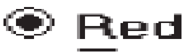


Figure 7.37

  - Assign unique keyboard mnemonics for each alternative in the standard way, choosing the first letter (or another) and designating it by character underlining.

- **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.

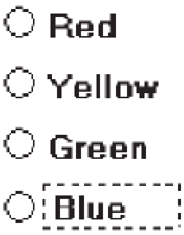
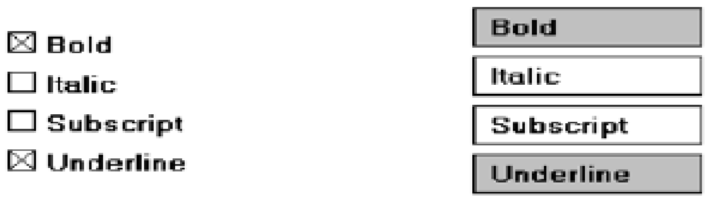


Figure 7.38

- 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

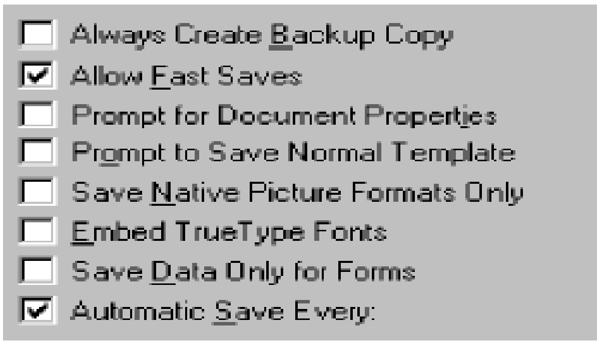
5.3.2 Check Boxes

- **Description**
  - Check box controls differ from radio buttons in that they permit selection of more than one alternative.
  - 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.



☒ Bold
☐ Italic
☐ Subscript
☒ Underline

Figure 7.39 Check boxes.

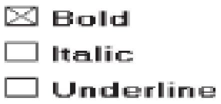


☐ Always Create Backup Copy
☑ Allow Fast Saves
☐ Prompt for Document Properties
☐ Prompt to Save Normal Template
☐ Save Native Picture Formats Only
☐ Embed TrueType Fonts
☐ Save Data Only for Forms
☑ Automatic Save Every:

Figure 7.40 Check boxes.

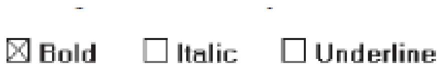
- **Purpose:** To set one or more options as either on or off.
- **Advantages**
  - Check Boxes provide Easy-to-access choices and Easy-to-compare choices.
  - Check Boxes are preferred by users.
- **Disadvantages**
  - Check Boxes Consume screen space and provide limited number of choices.
  - Single check boxes are difficult to align with other screen controls.
- **Proper usage**
  - Check Boxes are used for setting attributes, properties, or values and for nonexclusive choices.
  - Check Boxes can be used 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 and Never changed in content.
  - Check Boxes 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 and 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. Displaying more than eight is usually not efficient because it wastes screen space. If the number of choices exceeds this maximum, consider using a multiple selection list box.
- **Defaults**
  - When the control possesses a state or affect that has been preset, designate it as the default and display its check box marked.

- **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.



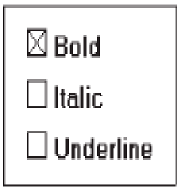
☒ Bold
☐ Italic
☐ Underline

  - 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.

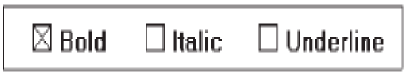


☒ Bold ☐ Italic ☐ Underline

  - Enclose the boxes in a border to visually strengthen the relationship they possess.

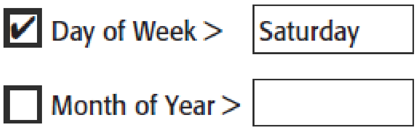


☒ Bold
☐ Italic
☐ Underline



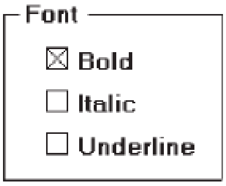
☒ Bold ☐ Italic ☐ Underline

- **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 (>).



☑ Day of Week > Saturday
☐ Month of Year >

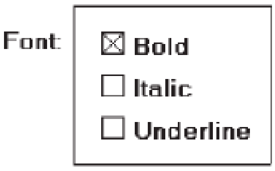
- **Captions**
    - Structure
        - Provide a caption for each grouping of related check boxes.
        - Exception: In screens containing only one check box grouping, the screen title may serve as the caption.
    - Display:
        - Fully spelled out and In mixed-case letters capitalizing the first letter of all significant words.
    - Columnar orientation:
        - With a control border, position the caption:
            - ❖ Upper-left-justified within the border.



Font

☒ Bold
□ Italic
□ Underline

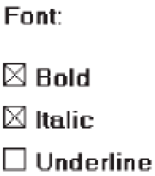
- ❖ Alternately, the caption may be located to the left of the topmost choice description.



Font

☒ Bold
□ Italic
□ Underline

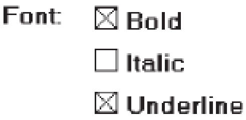
- Without an enclosing control border, position the caption:
    - ❖ Left-justified above the choice descriptions separated by one space line.



Font:

☒ Bold
☒ Italic
□ Underline

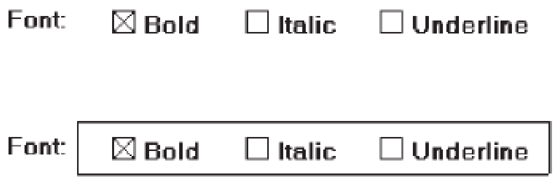
- ❖ Alternately, the caption may be located to the left of the topmost choice description.



Font

☒ Bold
□ Italic
☒ Underline

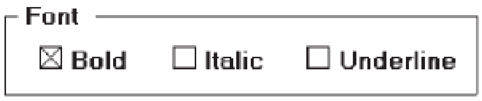
- Horizontal orientation
    - Position the caption to the left of the choice descriptions.



Font: ☒ Bold □ Italic □ Underline

Font: ☒ Bold □ Italic □ Underline

-
    - Alternately, with an enclosing control border, it should be left-justified within the border.

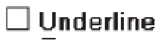


Font

☒ Bold □ Italic □ Underline

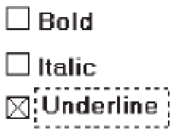
-
    - Be consistent in caption style and orientation within a screen.

- **Keyboard Equivalents**
    - Assign a keyboard mnemonic to each check box.
    - Designate the mnemonic by underlining the applicable letter in the choice description.



□ Underline

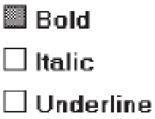
- **Selection Method and Indication**
    - Pointing:
        - The selection target area should be as large as possible. Include the check box and the choice description text.
        - Highlight the selection choice in some visually distinctive way when the cursor's resting on it and the choice is 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.



□ Bold
□ Italic
☒ Underline

-
    - 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.



▨ Bold
□ Italic
□ Underline

  - 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.

5.3.3 Palettes

- **Description**
  - Palette is 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.

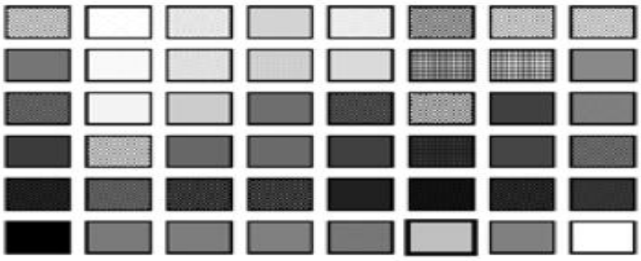
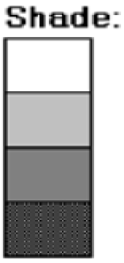


Figure 7.55 Palette.

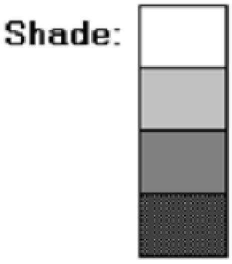
- **Purpose**: To set one of a series of mutually exclusive options presented graphically or pictorially.
- **Advantages**:
  - Palettes provide pictures aid comprehension.
  - Palettes provide easy-to-compare choices.
  - Palettes usually consume less screen space than textual equivalents.

- **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**
  - Palettes are used for setting attributes, properties, or values and for mutually exclusive choices.
  - Palettes can be used were 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 and Rarely changed in content.
  - Do not use: Where the alternatives cannot be meaningfully and clearly represented pictorially, where words are clearer than images and 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 and Permit ease in pointing and selecting.
  - Create images of equal size and always test illustrations before implementing them.
- **Size**
  - Present all available alternatives within the limits imposed by the size of the graphical representations and the screen display’s capabilities.

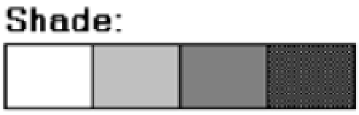
- **Layout**
  - Create boxes large enough to effectively illustrate the available alternatives and Permit ease in pointing and selecting.
  - Create boxes of equal size and Position the boxes adjacent to, or butted up against, one another.
  - A columnar orientation is the preferred manner and if vertical 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 and 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.



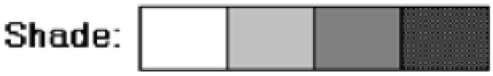
    - Alternately, the caption may be positioned to the left of the topmost alternative.



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



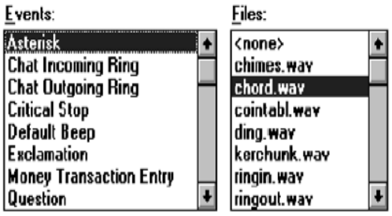
  - Alternately, the caption may be positioned to the left of the alternatives.



- **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.

5.3.4 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.

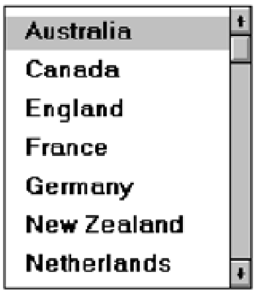


- **Purpose**
    - To display a collection of items containing mutually exclusive or non-mutually-exclusive options.
- **Advantages**
    - List Boxes can provide unlimited number of choices.
    - List Boxes remind users of available options.
    - List boxes are always visible.
- **Disadvantages**
    - List Boxes consume screen space.
    - List boxes 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) or 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 and fixed or variable in list length.
    - When screen space or layout considerations make radio buttons or check boxes impractical.

5.3.4.1 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 and 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.

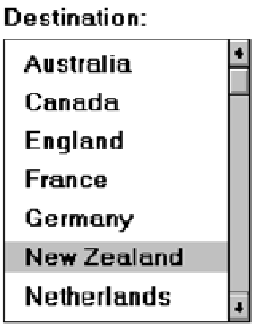
- Box Size
  - Must be long enough to display 6 to 8 choices without requiring scrolling.
  - Exceptions: If screen space constraints exist, the box may be reduced in size to display at least three items. If it is the major control within a window, the box may be larger.
  - If more items are available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



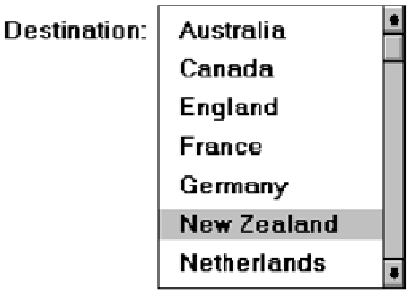
Australia
Canada
England
France
Germany
New Zealand
Netherlands

  - 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.

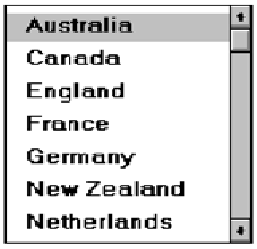
- Captions
  - Use mixed-case letters.
  - The preferred position of the control caption is above the upper-left corner of the list box.



  - 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. Display a list box's caption and entries as dimmed or grayed when the list box is entirely disabled.
- 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:



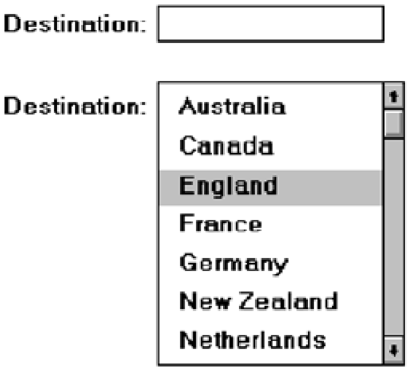
- ❖ Use a reverse video or reverse color bar to surround the choice description when it is selected.
- ❖ The cursor should be as wide as the box itself.
- ❖ Mark the selected choice in a distinguishing way.

- ▪ Activation: Require the pressing of a command button when an item, or items, is selected.

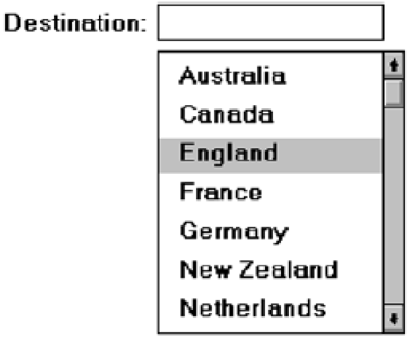
5.3.4.2. Single-Selection List Boxes (guidelines)

- ○ 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.

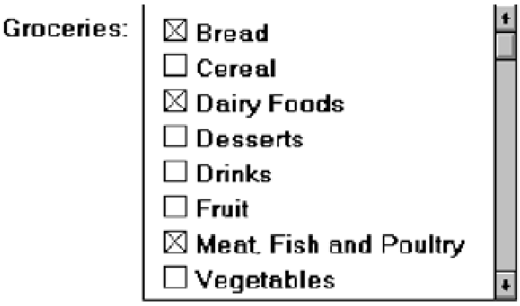


- ▪ Use the same background color for the text box as is used in the list box.

- Defaults:
  - When the list box is first displayed:
    - Present the currently active choice highlighted or marked with a circle or 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.

5.3.4.3 Extended and Multiple-Selection List Boxes (Guidelines)

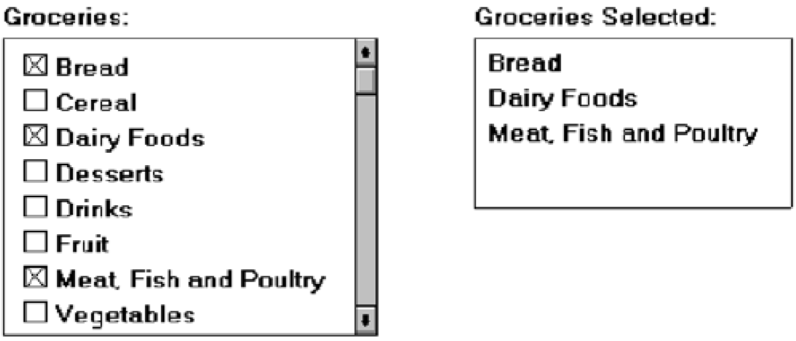
- Purpose:
  - To permit selection of more than one item in a long listing.
  - Extended list box: Optimized for individual item or range selection.
  - Multiple-selection list box: Optimized for independent item selection.
- Design guidelines:
  - Selection indication:
    - Mark the selected choice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



Groceries:

☒ Bread
☐ Cereal
☒ Dairy Foods
☐ Desserts
☐ Drinks
☐ Fruit
☒ Meat. Fish and Poultry
☐ Vegetables

- 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.



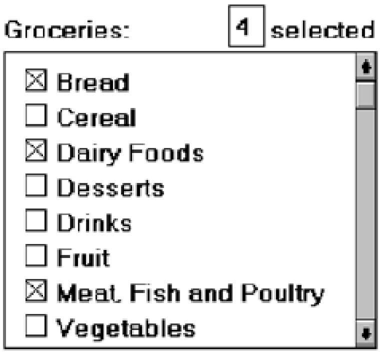
Groceries:

☒ Bread
☐ Cereal
☒ Dairy Foods
☐ Desserts
☐ Drinks
☐ Fruit
☒ Meat, Fish and Poultry
☐ Vegetables

Groceries Selected:

Bread
Dairy Foods
Meat, Fish and Poultry

- ❖ 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.



Groceries: 4 selected

☒ Bread
☐ Cereal
☒ Dairy Foods
☐ Desserts
☐ Drinks
☐ Fruit
☒ Meat, Fish and Poultry
☐ Vegetables

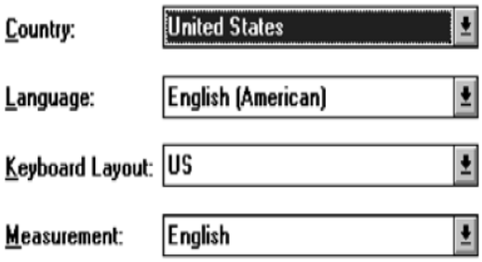
- Select All and Deselect All buttons
  - ❖ Provide command buttons to accomplish fast Select All and Deselect All actions, when these actions must be frequently or quickly performed.
- Defaults:
  - ❖ When the list box is first displayed:
    - ➢ Display the currently active choices highlighted.
    - ➢ 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.

5.3.5. 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.

5.3.6 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.
  - They are called drop-down lists because they appear to drop down from the single-selection field.
  - Microsoft Windows calls them drop-down list boxes. Other common names are pull-down lists, option menus, and simply list boxes.



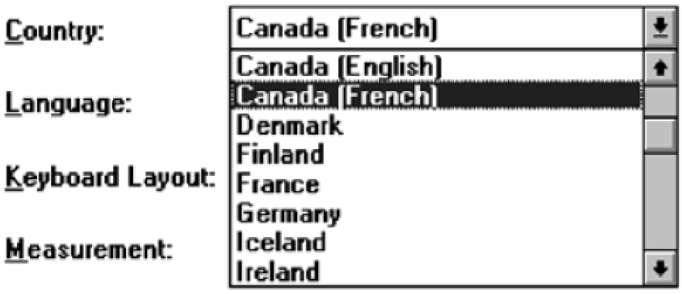
Country: United States

Language: English (American)

Keyboard Layout: US

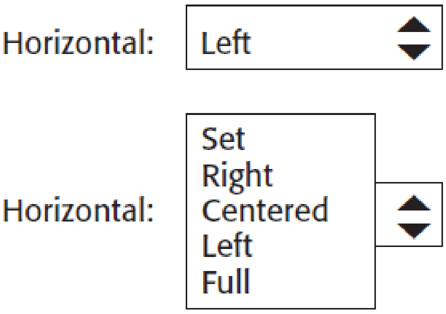
Measurement: English

Figure 7.70 Drop-down list boxes. There are four unopened boxes, Country, Language, Keyboard Layout, and Measurement.



Country: Canada (French)

Figure 7.71 Drop-down list box opened for Country.



Horizontal: Left

Figure 7.72 Pop-up list box, closed and opened.

- **Purpose:** To select one item from a large list of mutually exclusive options when screen space is limited.
- **Advantages**
  - Drop-down/pop-up list boxes provide unlimited number of choices.
  - Drop-down/pop-up list boxes reminds users of available options.
  - The most useful feature of drop-down/pop-up list boxes is that they conserve screen space.

- **Disadvantages**
  - Drop-down/pop-up list boxes requires an extra action to display the list of choices.
  - Drop-down/pop-up list boxes 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**
  - Drop-down/pop-up list boxes useful for selecting values or setting attributes.
  - Drop-down/pop-up list boxes can be used for choices that are mutually exclusive.
  - Drop-down/pop-up list boxes are suitable 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 an unpredictable fashion, Large in number and 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.
- **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.

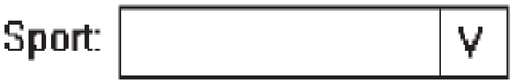


Figure 7.73

- **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. A list being displayed in a drop-down/pop-up list box has no size limit.
  - Present all available alternatives. All available alternatives should be capable of being displayed.
- **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 a field caption in mixed-case letters, with each significant word capitalized is necessary.
  - 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**
  - Display a drop-down/pop-up list box’s caption and entries as dimmed or grayed out when the list box is entirely disabled.
- **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.

5.4 Custom Controls

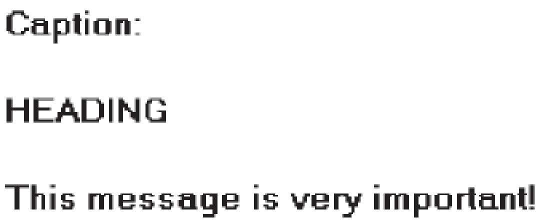
- Many toolkits and interface builders provide the ability to create custom controls, implement them with caution.
- If custom controls are used, make the look and behavior of custom controls different from that of standard controls.

5.5 Presentation Controls

- Presentation controls are purely informational. They provide details about other screen elements or controls, or assist in giving the screen structure.
- Common presentation controls are static text fields, group boxes column headings, ToolTips, balloon tips, and progress indicators.

5.5.1 Static Text Fields

- **Description**
  - Read-only textual information. A static text field, as illustrated in Figure 7.84, provides read-only textual information. It is a standard Microsoft Windows control.



Caption:

HEADING

This message is very important!

Figure 7.84 Static text field.

- **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.
  - Examples are headings, subheadings, slider scales, progress indicator text, and so on
- **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.

5.5.2 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.
  - Standard Microsoft Windows Group boxes are illustrated in below figure.

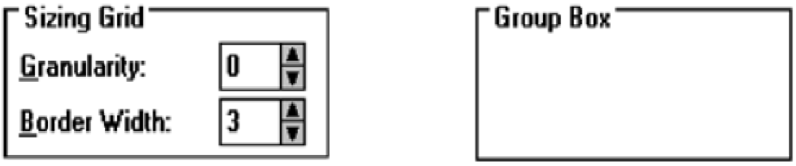
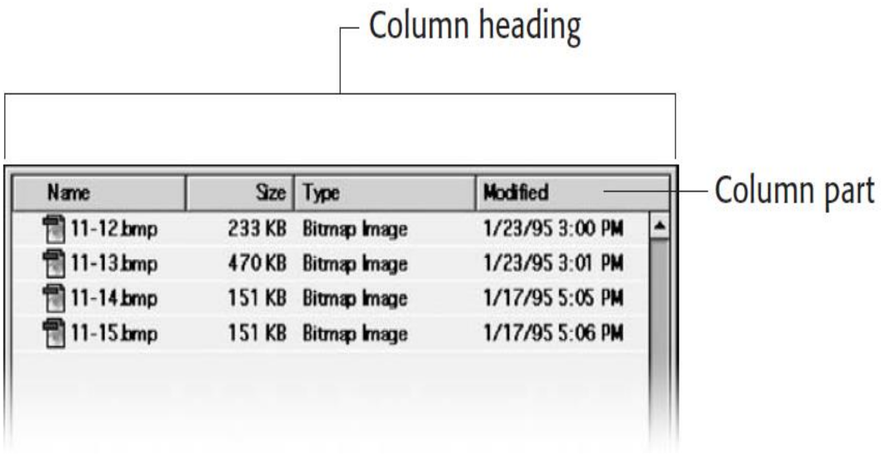


Figure 7.85 Group boxes.

- **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 (:).

5.5.3 Column Headings

- **Description**
  - Read-only textual information that serves as a heading above columns of text or numbers.
  - A column heading is illustrated in Figure 7.86. It can be divided into two or more parts. Can be divided into two or more parts.



Name	Size	Type	Modified
11-12.bmp	233 KB	Bitmap Image	1/23/95 3:00 PM
11-13.bmp	470 KB	Bitmap Image	1/23/95 3:01 PM
11-14.bmp	151 KB	Bitmap Image	1/17/95 5:05 PM
11-15.bmp	151 KB	Bitmap Image	1/17/95 5:06 PM

Figure 7.86 Column heading control.

- **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.

5.5.4 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 or in need of additional descriptive or status information.
  - A ToolTip, sometimes called a Screen Tip, is a standard Microsoft Windows control, a small pop-up window that displays descriptive text when the pointer is moved over a control that does not possess a caption, or that possesses only an abbreviated caption. A ToolTip is illustrated in below figure.

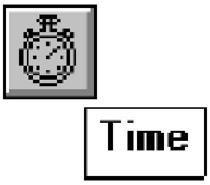


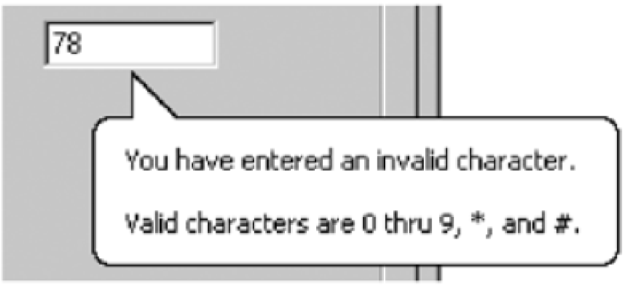
Figure 7.87 ToolTip

- **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 Guidelines**
  - 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.

5.5.5 Balloon Tips

- **Description**
  - A small pop-up window that contains information in a word balloon.
  - Components can include, Title, Body text and 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.


78

You have entered an invalid character.

Valid characters are 0 thru 9, *, and #.

Figure 7.88 Balloon tip.

- **Purpose:** To provide additional descriptive or status information about a screen element.
- **Advantage:** 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 and 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 and 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.

5.5.6 Progress Indicators

- **Description**
  - A rectangular bar that fills as a process is being performed, indicating the percentage of the process that has been completed.

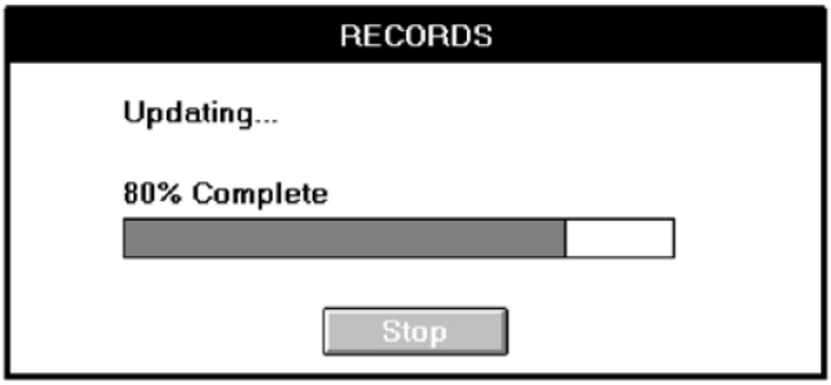


Figure 7.89 Progress indicator.

- **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.

5.5.7 Sample Box

- **Description**
  - A box illustrating what will show up on the screen based upon the parameter or parameters selected.
  - May include text, graphics, or both.

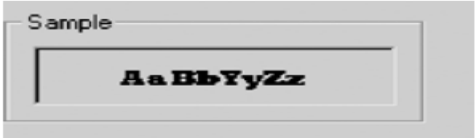


Figure 7.90 Sample box.

- **Purpose:** To provide a representation of actual screen content based upon the parameter or 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.

5.5.8 Scrolling Tickers

- **Description:** Text that scrolls horizontally through a container window.
- **Advantage** is that it consume less screen space than full text.
- **Disadvantages**
  - Scrolling Tickers are hard to read.
  - Scrolling Tickers are time-consuming to interpret.
  - Scrolling Tickers are sometimes distracting.
- **Guideline:** The most prudent guideline is do not use scrolling tickers.

5.6 TEST 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.

5.6.1 Kinds of prototypes

- **Hand Sketches and Scenarios**
  - Description
    - Screen sketches created by hand. These can start early in the design process and before any attempt is made to create a prototype using an available toolkit or interface builder.
    - Focus is on the design, not the interface mechanics. This sketch will be useful in defining and refining task organization, conceptual ideas, and the general layout of screens.
    - It is a low-fidelity prototype.
  - Advantages
    - Sketches can be used very early in the development process.
    - Sketches are suited for use by entire design team.
    - There is no large investment of time and cost.
    - Sketches do not require any programming skills.
    - Sketches are easily portable and are fast to modify and iterate.
    - A rough approximation often yields more substantive critical comments.
    - Sketches are easier to comprehend than functional specifications.
    - They can also be used to define requirements.
  - Disadvantages
    - Sketches provide only a rough approximation.
    - Sketches are limited in providing an understanding of navigation and flow.
    - It is a demonstration, not an exercise.
    - It is 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.
        - The components are manually manipulated to reflect the dynamics of the software.
        - It is 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 is possible.
        - It is a demonstration, not an exercise.
        - It is driven by a facilitator, not the user.
        - Limited usefulness for usability testing.

- **Programmed Facades**
  - Description
    - To provide a realistic surface view of a real program and to illustrate some of the program’s functioning, a programmed facade can be created.
    - Examples of finished dialogs and screens for some important aspects of the system are constructed and viewed
    - Created by prototyping tools such as hypercard, supercard, and toolbook.
    - It is a medium-fidelity to high-fidelity prototypes.
  - Advantages
    - Provides a good detailed specification for writing code.
    - Acts as 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 and 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
    - To present an example of finished dialogs and screens for some important aspects of the system.
    - Created through programming languages that support the actual programming process.
    - It is a high-fidelity prototype.
  - Advantages
    - May include the final code.
    - Otherwise, generally the same as those of programmed facades.

- 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 and 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.

5.6.2 Comparisons of Prototypes

- Does fidelity affect a prototype’s usefulness as a testing tool. Two recent studies have addressed this issue.
- The first study, by Catani and Biers (1998), examined prototypes created at three fidelity levels: low (paper), (medium) screen shots, and high (using a prototype-oriented language—Visual Basic). There were no significant differences in the number and severity of problems identified with each kind of prototype. There was also a high degree of commonality in the specific problems uncovered.
- The second study, reported by Uceta, Dixon, and Resnick (1998), compared a paperbased prototype with a computer-based prototype. Both interfaces were identical except for the medium of presentation. Identifying problems using the paper prototype, however, took about 30 percent longer than using the computer-based prototypes.
- The results of these studies indicate that prototype fidelity seems to have no impact on the identification of usability problems.

5.6.3 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 and Retention of learning over time.
    - Speed of task completion, need fulfilment, subjective user satisfaction and error rates.

- **Guidelines Review**
    - Description
        - A review of the interface in terms of an organization’s standards and design guidelines.
        - A checklist summarizing a system’s standard or guideline document is prepared and is used as the basis for the comparison.
        - Failure to comply with a guideline or standard indicates that a design modification may be necessary.
    - Advantages
        - It can be performed by developers.
        - Low cost incurred.
        - Can identify general and recurring problems
        - Particularly useful for identifying screen design and layout problems.
    - Disadvantage: May miss severe conceptual, navigation, and operational problems.

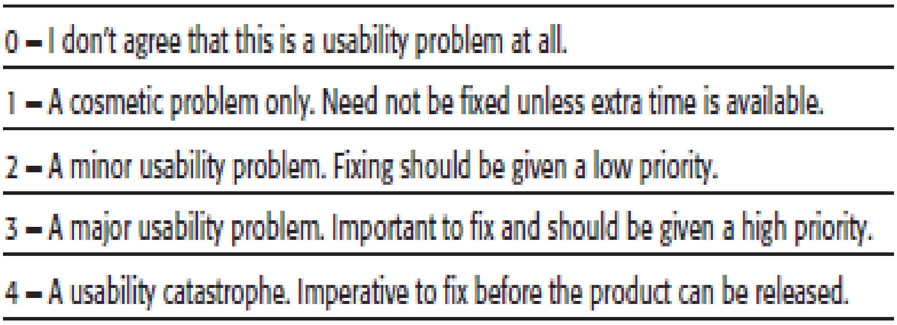
- **Heuristic Evaluation**
    - Description
        - A detailed evaluation of a system by interface design specialists to identify problems.
        - The interface is judged for its compliance with recognized usability principles, the heuristics.
    - Advantages
        - Easy to do and of relatively low cost.
        - Does not waste user’s time and 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
  - Based upon a study, Nielsen (1992) suggests that the optimum expert group size to satisfactorily perform a heuristic evaluation is 3 to 5 people.
  - Choose knowledgeable people who are familiar with the project situation and possessing a long-term relationship with the organization.

- **Heuristic Evaluation Process**
  - Preparing the session:
    - ➢ Select evaluators and prepare or assemble a project overview and a checklist of heuristics.
    - ➢ Provide briefing to evaluators to:
      - ✓ Review the purpose of the evaluation session and preview the evaluation process.
      - ✓ Present the project overview and heuristics.
      - ✓ Answer any evaluator questions and 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 or 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.

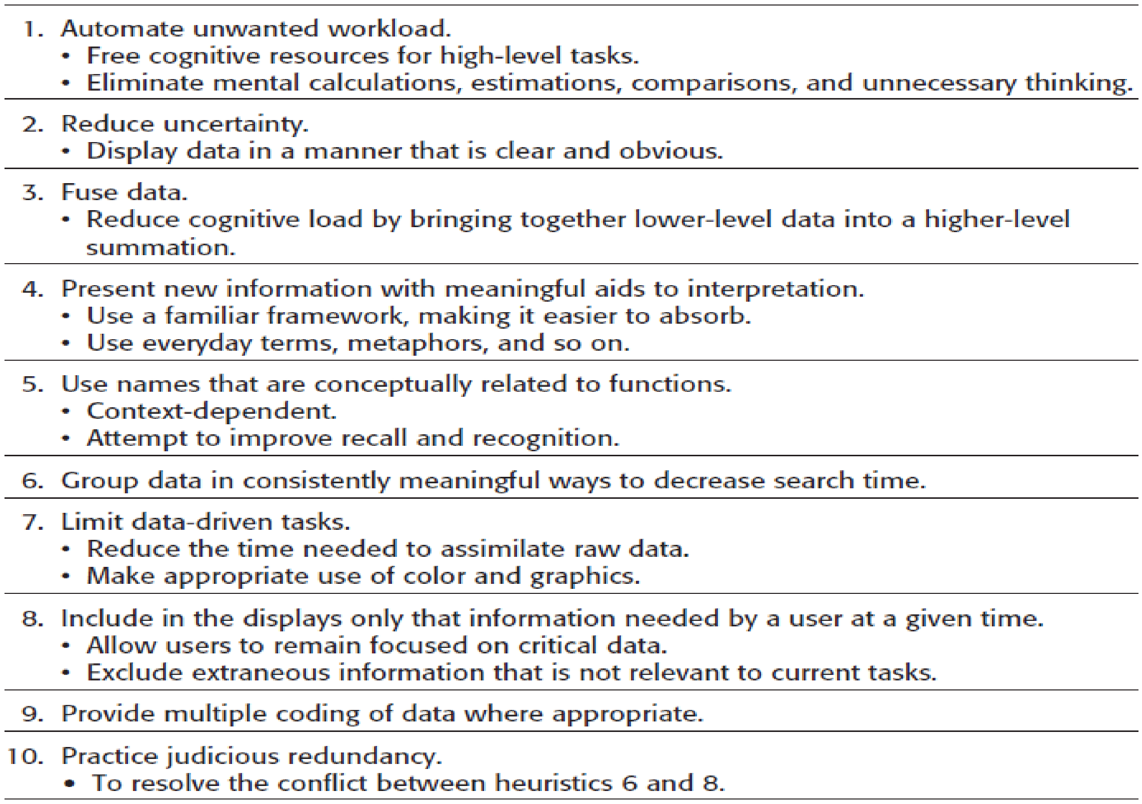
Table 14.1 Severity Ratings in Heuristic Evaluation



0 – I don't agree that this is a usability problem at all.
1 – A cosmetic problem only. Need not be fixed unless extra time is available.
2 – A minor usability problem. Fixing should be given a low priority.
3 – A major usability problem. Important to fix and should be given a high priority.
4 – A usability catastrophe. Imperative to fix before the product can be released.

- ○ **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.
    - Bailey suggests a better research-based set of heuristics will lead to improved evaluation results, for example, those proposed byGerhardt-Powals (1996). This set of heuristics is summarized in Table 14.2.

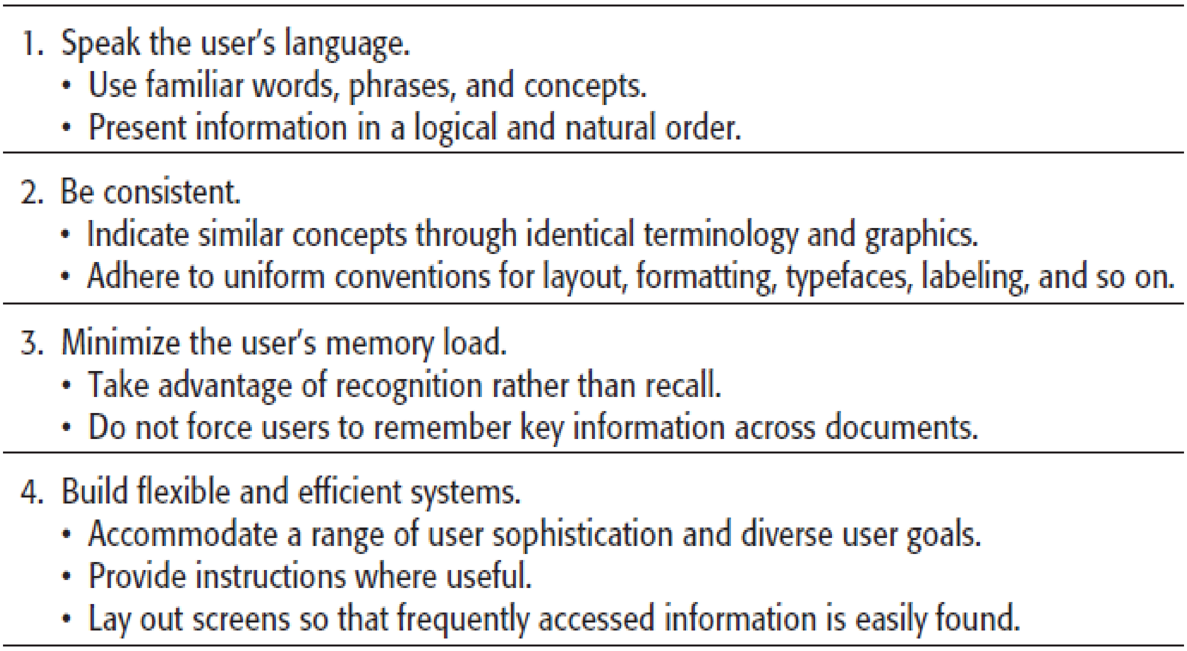
Table 14.2 Research-Based Set of Heuristics



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.

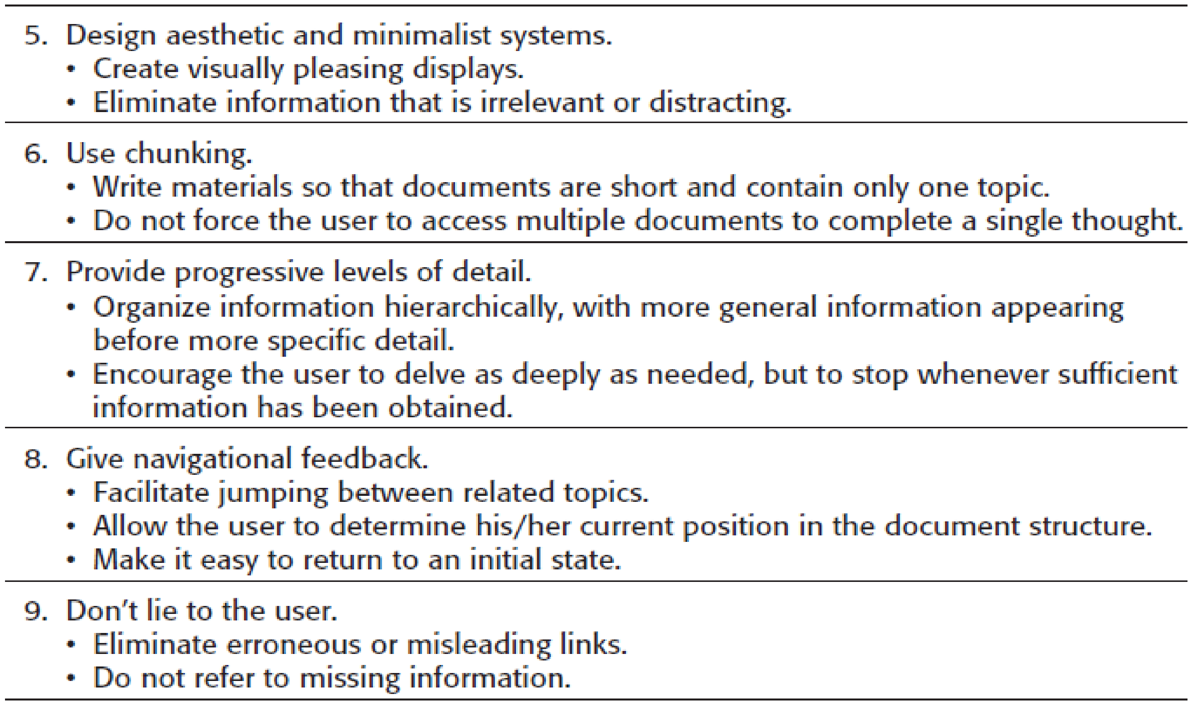
- Web heuristics are still an evolving entity and must also be validated through research. The set proposed by Levi and Conrad (1996), and summarized in Table 14.3, seem a good place to start.

Table 14.3 Possible Web Page Heuristics



1. Speak the user's language.
   - Use familiar words, phrases, and concepts.
   - Present information in a logical and natural order.
2. Be consistent.
   - Indicate similar concepts through identical terminology and graphics.
   - Adhere to uniform conventions for layout, formatting, typefaces, labeling, and so on.
3. Minimize the user's memory load.
   - Take advantage of recognition rather than recall.
   - Do not force users to remember key information across documents.
4. Build flexible and efficient systems.
   - Accommodate a range of user sophistication and diverse user goals.
   - Provide instructions where useful.
   - Lay out screens so that frequently accessed information is easily found.

Table 14.3 Continued



5. Design aesthetic and minimalist systems.
   - Create visually pleasing displays.
   - Eliminate information that is irrelevant or distracting.
6. Use chunking.
   - Write materials so that documents are short and contain only one topic.
   - Do not force the user to access multiple documents to complete a single thought.
7. Provide progressive levels of detail.
   - Organize information hierarchically, with more general information appearing before more specific detail.
   - Encourage the user to delve as deeply as needed, but to stop whenever sufficient information has been obtained.
8. Give navigational feedback.
   - Facilitate jumping between related topics.
   - Allow the user to determine his/her current position in the document structure.
   - Make it easy to return to an initial state.
9. Don't lie to the user.
   - Eliminate erroneous or misleading links.
   - Do not refer to missing information.

- **Cognitive Walkthroughs**
  - Description
    - Reviews of the interface in the context of tasks users perform.
    - Walkthroughs require that the task definition methodology must have been properly accomplished in the system requirements stage. The user's goals and assumptions must also be clearly defined before the walkthrough is performed.
  - Advantages
    - Allow a clear evaluation of the task flow early in the design process.
    - Do not require a functioning prototype.
    - Low cost incurred.
    - Can be used to evaluate alternate solutions and can be performed by developers.
    - More structured than a heuristic evaluation and useful for assessing "exploratory learning."
  - Disadvantages
    - It is 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.
  - Disadvantage: May be difficult to get users to think out loud.
  - Guidelines
    - Develop several core or representative tasks and 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 analysing 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.
        - In a focus group, a small group of knowledgeable users and a moderator are brought together to discuss an interface design prototype or proposed design tasks.
        - The discussion is loosely structured but must be focused on a specific topic or topics.
    - Advantages
        - Useful for obtaining initial user thoughts and trying out ideas.
        - Easy to set up and run.
        - Low cost incurred.
    - Disadvantages
        - Requires experienced moderator.
        - Not useful for establishing, how people really work and 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 using a 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. One such example is a system, described by Hornof and Kieras (1997), called Executive Process Interactive Control (EPIC). Formal evaluations by a tool such as this have the potential to greatly improve the quality of many user interfaces.
  - 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.

QUESTION BANK

1. What are operable controls? Explain Usage of buttons along with their advantages and disadvantages. (10 Marks)
2. Explain the following controls with an example for each: (10 Marks)
   a. Radio buttons.
   b. Check boxes.
   c. Tool tips.
   d. Progress indicators.
   e. Sample boxes.
3. Explain the purpose of prototypes. Discuss any two types of prototypes with their importance to system developers. (10 marks)
4. Write a note on think-Aloud-Evaluations and Usability test (10 marks)
5. Explain Radio buttons and List boxes selection control. (10 marks)
6. Explain cognitive walkthroughs and Heuristic Evaluation tests conducted in user interface design. (10 marks)
7. Explain the following text controls:
   a. Text Boxes
   b. Captions (10 marks)
8. Define selection control. Briefly explain List Boxes and List view controls. (10 marks)
9. List all and explain any two presentation controls. (10 marks)
10. Explain the effectiveness of heuristic evaluation. (10 marks)