FUTURE VISION BIE

By K B Hemanth Raj

Visit : https://hemanthrajhemu.github.io

A Small Contribution Would Support Us.

Dear Viewer,

Future Vision BIE is a free service and so that any Student/Research Personal **Can Access Free of Cost**.

If you would like to say **thanks**, you can make a **small contribution** to the author of this site.

Contribute whatever you feel this is worth to you. This gives **us support** & to bring **Latest Study Material** to you. After the Contribution Fill out this Form (https://forms.gle/tw3T3bUVpLXL8omX7). To Receive a **Paid E-Course for Free**, from our End within 7 Working Days.

Regards

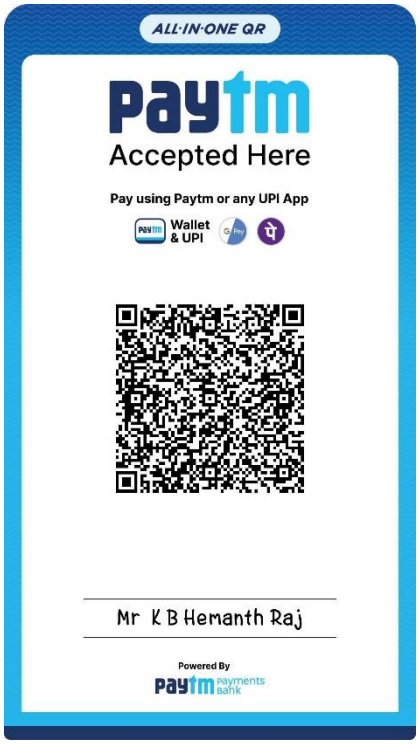
- K B Hemanth Raj (Admin)

Contribution Methods

UPI ID

1. futurevisionbie@oksbi
2. futurevisionbie@paytm

Scan & Pay



Account Transfer

Account Holder's Name: K B Hemanth Raj

Account Number: 39979402438

IFSC Code: SBIN0003982

MICR Code: 560002017

More Info: https://hemanthrajhemu.github.io/Contribution/

Gain Access to All Study Materials according to VTU,
CSE – Computer Science Engineering,
ISE – Information Science Engineering,
ECE - Electronics and Communication Engineering & MORE...

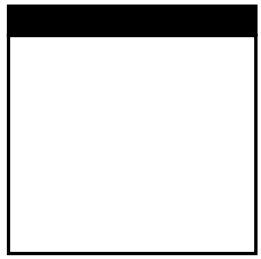
Stay Connected... get Updated... ask your queries...

Join Telegram to get Instant Updates: https://bit.ly/VTU_TELEGRAM

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/futurevisionbie/

WHATSAPP SHARE: https://bit.ly/FVBIESHARE



The Essential Guide to User Interface Design

An Introduction to GUI Design Principles and Techniques

Third Edition

Wilbert O. Galitz

Wiley Publishing, Inc.

STEP

5

Select the Proper Kinds of Windows

A window is an area of the screen, usually rectangular in shape, defined by a border that contains a particular view of some area of the computer or some portion of a person's dialog with the computer. It can be moved and rendered independently on the screen. A window may be small, containing a short message or a single field, or it may be large, consuming most or all of the available display space. A display may contain one, two, or more windows within its boundaries. In this step the following is addressed:

- A window's characteristics.
- A window's components.
- A window's presentation styles.
- The types of windows available.
- Organizing window system functions.
- A window's operations.
- Web system frames and pop-up windows.

Window Characteristics

A window is seen to possess the following characteristics:

- A name or title, allowing it to be identified.
- A size in height and width (which can vary).

- A state, accessible or active, or not accessible. (Only active windows can have their contents altered.)
- Visibility — the portion that can be seen. (A window may be partially or fully hidden behind another window, or the information within a window may extend beyond the window's display area.)
- A location, relative to the display boundary.
- Presentation, that is, its arrangement in relation to other windows. It may be tiled, overlapping, or cascading.
- Management capabilities, methods for manipulation of the window on the screen.
- Its highlight, that is, the part that is selected.
- The function, task, or application to which it is dedicated.

The Attraction of Windows

The value of windowing is best seen in the context of a task or job. A person performs a variety of tasks, often in a fairly unstructured manner. A person is asked to monitor and manipulate data from a variety of sources, synthesize information, summarize information, and reorganize information. Things are seldom completed in a continuous time frame. Outside events such as telephone calls, supervisor or customer requests, and deadlines force shifts in emphasis and focus. Tasks start, stop, and start again. Materials used in dealing with the tasks are usually scattered about one's desk, being strategically positioned in the workspace to make handling the task as efficient as possible. This spatial mapping of tools helps people organize their work and provides reminders of uncompleted tasks. As work progresses and priorities change, materials are reorganized to reflect the changes.

Single-screen technology supported this work structure very poorly. Because only one screen of information could be viewed at one time, comparing or integrating information from different sources and on different screens often required extensive use of one's memory. To support memory, a person was often forced to write notes or obtain printed copies of screens. Switching between tasks was difficult and disruptive, and later returning to a task required an extensive and costly restructuring of the work environment.

The appeal of windowing is that it allows the display workspace to mirror the desk workspace much more closely. This dramatically reduces one's short-term memory load. One's ability to do mental calculations is limited by how well one keeps track of one's place, one's interim conclusions and products, and, finally, the results. Windows act as external memories that are an extension of one's internal memory. Windows also make it much easier to switch between tasks and to maintain one's context, because one does not have to reestablish one's place continually. In addition, Windows provide access to more information than would normally be available on a single display of the same size. Overwriting, or placing more important information on top of that of less importance at that moment, does this.

While all the advantages and disadvantages of windows are still not completely understood, windows do seem to be useful in the following ways.

Presentation of Different Levels of Information

Information can be examined in increasing levels of detail. A document table of contents can be presented in a window. A chapter or topic selected from this window can be simultaneously displayed in more detail in an adjoining window. Deeper levels are also possible in additional windows.

Presentation of Multiple Kinds of Information

Variable information needed to complete a task can be displayed simultaneously in adjacent windows. An order-processing system window could collect a customer account number in one window and retrieve the customer's name and shipping address in another window. A third window could collect details of the order, after which another window could present factory availability of and shipping dates for the desired items. Significant windows could remain displayed so that details may be modified as needed prior to order completion. Low inventory levels or delayed shipping dates might require changing the order.

Sequential Presentation of Levels or Kinds of Information

Steps to accomplish a task can be sequentially presented through windows. Successive windows are presented until all the required details are collected. Key windows may remain displayed, but others appear and disappear as necessary. This sequential preparation is especially useful if the information-collection process leads down various paths. An insurance application, for example, will include different types of coverage. A requested type of coverage might necessitate the collection of specific details about that type of coverage. This information can be entered into a window presented to collect the unique data. The windows disappear after data entry, and additional windows appear when needed.

Access to Different Sources of Information

Independent sources of information may have to be accessed at the same time. This information may reside in different host computers, operating systems, applications, files, or areas of the same file. It may be presented on the screen alongside the problem, greatly facilitating its solution. For instance, a writer may have to refer to several parts of a text being written at the same time. Or, a travel agent may have to compare several travel destinations for a particularly demanding client.

Combining Multiple Sources of Information

Text from several documents may have to be reviewed and combined into one. Pertinent information is selected from one window and copied into another.

Performing More Than One Task

More than one task can be performed at one time. While waiting for a long, complex procedure to finish, another can be performed. Tasks of higher priority can interrupt less important ones. The interrupted task can then be resumed without the necessity to "close down" and "restart."

Reminding

Windows can be used to remind the viewer of things likely to be of use in the near future. Examples might be menus of choices available, a history of the path followed or the command choices to that point, or the time of an important meeting.

Monitoring

Changes, both internal and external, can be monitored. Data in one window can be modified and its effect on data in another window can be studied. External events, such as changes in stock prices, out of normal range conditions, or system messages can be watched while another major activity is carried out.

Multiple Representations of the Same Task

The same thing can be looked at in several ways — for example, alternate drafts of a speech, different versions of a screen, or different graphical representations of the same data. A maintenance procedure may be presented in the form of textual steps and illustrated graphically at the same time.

Constraints in Window System Design

Windowing systems, in spite of their appeal and obvious benefits, have failed to completely live up to their expectations. In the past, a windows user interface has been described as "chaotic" because of the great amount of time users must spend doing such things as pointing at tiny boxes in window borders, resizing windows, moving windows, closing windows, and so forth. The problems with windowing systems can be attributed to three factors: historical considerations, hardware limitations, and human limitations.

Historical Considerations

Historically, system developers have been much more interested in solving hardware problems than in user considerations. Because technical issues abound, they have received the most attention. There has been little research addressing design issues and their impact on the usability of window systems. Therefore, few concrete window design guidelines are available to aid designers.

This lack of guidelines makes it difficult to develop acceptable and agreeable window standards. While many companies have developed style guides, they are very general and limited in scope to their products. Standardization is also made more difficult by the complexity and range of alternatives available to the designer. Without user performance data, it is difficult to compare realistically the different alternatives, and design choices become a matter of preference.

Standardization of the interface is also inhibited by other factors. Some software developers, who are proud of their originality, see standards as a threat to creativity and its perceived monetary rewards. Some companies are wary of standards because they fear that other companies are promoting standards that reflect their own approach. Finally, some companies have threatened, or brought, legal action against anyone who adopts an approach similar to their own.

The result is that developers of new systems create another new variation each time they design a product, and users must cope with a new interface each time they encounter a new windowing system.

Hardware Limitations

Many of today's screens are not large enough to take full advantage of windowing capabilities. As a result, many windows are still of "Post-it" dimensions. As already mentioned, there is some evidence that many users of personal computers expand their windows to cover a full screen. Either seeing all the contents of one window is preferable to seeing small parts of many windows or the operational and visual complexity of multiple windows is not wanted.

Also, the slower processing speeds and smaller memory sizes of some computers may inhibit the use of windows. A drain on the computer's resources may limit feedback and animation capabilities, thereby reducing the system's usability. Poor screen resolution and graphics capability may also deter effective use of windows by not permitting sharp and realistic drawings and shapes.

Human Limitations

A windowing system, because it is more complex, requires the learning and using of more operations. Much practice is needed to master them. These window management operations are placed on top of other system operations, and window management can become an end in itself. This can severely detract from the task at hand. In one study comparing full screens with screens containing overlapping windows, task completion times were longer with the window screens, but the non-window screens generated more user errors. After eliminating screen arrangement time, however, task solution times were shorter with windows. The results suggest that advantages for

windows do exist, but they can be negated by excessive window manipulation requirements.

It is also suggested that to be truly effective, window manipulation must occur implicitly as a result of user task actions, not as a result of explicit window management actions by the user.

Other Limitations

Other possible window problems include the necessity for window borders to consume valuable screen space, and that small windows providing access to large amounts of information can lead to excessive, bothersome scrolling.

Where To?

In spite of their problems, windows do have enormous benefits and are here to stay. So, we must cope with their constraints for now and, in the meantime, enjoy the benefits they possess.

Components of a Window

A typical window may be composed of up to a dozen or so elements. Some appear on all windows, others only on certain kinds of windows, or under certain conditions. For consistency purposes, these elements should always be located in the same position within a window. Most windowing systems provide consistent locations for elements in their own windows. Some inconsistencies do exist in element locations between different systems, however, as do some differences in what the elements are named, or what graphic images or icons are chosen to identify them. What follows is a description of typical window components and their purposes, with emphasis on the most popular windowing system, *Microsoft Windows*. Specifically reviewed will be primary windows, secondary windows, and a form of secondary window called the dialog box. An illustration of a primary window is found in Figure 5.1. Illustrations of secondary windows and dialog boxes are illustrated in Figures 5.8 and 5.13. How these different types of windows are used will be described in a later section in this step. A summary of window components for these types of windows is also found in Table 5.1.

Frame

A window will have a frame or border, usually rectangular in shape, to define its boundaries and distinguish it from other windows. While a border need not be rectangular, this shape is a preferred shape for most people. Also, textual materials, which are usually read from left to right, fit most efficiently within this structure. The border comprises a line of variable thickness and color. This variation can be used as an aid in identifying the type of window being displayed. Windows filling an entire screen may use the screen edge as the border. If a window is resizable, it may contain control points for sizing it. If the window cannot be resized, the border coincides with the edge of the window.

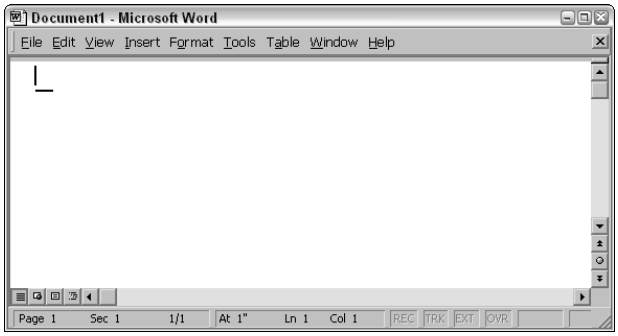


Figure 5.1: Microsoft Windows primary window.

Title Bar

The title bar is the top edge of the window, inside its border and extending its entire width. This title bar is also referred to by some platforms as the *caption*, *caption bar*, or *title area*. The title bar contains a descriptive title identifying the purpose or content of the window. In Microsoft Windows, the title bar may also possess, at the extreme left and right ends, control buttons (described later) for retrieving the system menu and performing window resizing. The title bar also serves as a control point for moving the window and as an access point for commands that apply to a window. For example, as an access point, when a user clicks on the title bar using the secondary mouse button, the pop-up or shortcut menu for the window appears. Pressing the Alt+Spacebar key combination also displays the shortcut menu for the window. Title bars are included on all primary and secondary windows. Title bar text writing guidelines are described in Step 8.

Microsoft recommends that one never place application commands or other controls in the title bar. Doing so may conflict with the special user controls Windows adds for configurations that support multiple languages.

Title Bar Icon

Located at the left corner of the title bar in a primary window, this button is used in Windows to retrieve a pull-down menu of commands that apply to the object in the window. It is a 16×16 version of the icon of the object being viewed. When clicked with the secondary mouse button, the commands applying to the object are presented. Microsoft suggests that

- If the window contains a tool or utility (that is, an application that does not create, load, and save its own data files), a small version of the application's icon should be placed there instead.
- If the application creates, loads, and saves documents or data files and the window represents the view of one of its files, a small version of the icon that represents its document or data file type should be placed there.
- Even if the user has not yet saved the file, display the data file icon rather than the application icon, and again display the data file icon after the user saves the file.

Table 5.1: Microsoft Windows Components

	WINDOWS CONTAINING COMPONENT		
COMPONENT	**PRIMARY**	**SECONDARY**	**DIALOG BOX**
Frame or Border • Boundary to define shape. • If sizable, contains control points for resizing.	X	X	X
Title Bar Text • Name of object being viewed in window. • Control point for moving window.	X	X	X
Title Bar Icon • Small version of icon for object being viewed. • Access point for commands that apply to the object.	X		
Title Bar Buttons • Shortcuts to specific commands.	X	X	X
Close	X	X	X
Minimize/Maximize/Restore	X		
What's This? • Displays context-sensitive Help about any object displayed on window.		X	X
Menu Bar • Provides basic and common application commands.	X		
Status Bar • *An area used to display status information about what is displayed in window.*	X		
Scroll Bar • Standard control to support scrolling.	X		
Size Grip • *Control to resize window, located at right side of status bar.*	X		

Window Sizing Buttons

Located at the right corner of the title bar, these buttons are used to manipulate the size of a window. The leftmost button, the *minimize* button — inscribed with a short horizontal line toward the bottom of the button — is used to reduce a window to its minimum

size, usually an icon. It also hides all associated windows. The *maximize* button — typically inscribed with a large box — enlarges a window to its maximum size, usually the entire screen. When a screen is maximized, the restore button replaces the maximize button, because the window can no longer be increased in size. The restore button — typically inscribed with a pair overlapping boxes — returns a window to the size it had before a minimize or maximize action was performed. A close button — typically inscribed with an X — closes the window. Minimize, maximize, and close buttons are shown in Figure 5.1. These command buttons are graphical equivalents to the actions available through the Title Bar icon.

Sizing buttons are included on primary windows only. All buttons on a primary window's title bar must have equivalent commands on the pop-up or shortcut menu for that window.

When these buttons are displayed, use the following guidelines:

- When a window does not support a command, do not display its command button.
- The Close button always appears as the rightmost button. Leave a gap between it and any other buttons.
- The Minimize button always precedes the Maximize button.
- The Restore button always replaces the Maximize button or the Minimize button when that command is carried out.

What's This? Button

The *What's This?* button, which appears on secondary windows and dialog boxes, is used to invoke the What's This? Windows command to provide contextual Help about objects displayed within a secondary window. When provided, it is located in the upper-right corner of the title bar, just to the left of the close button. It is inscribed with a question mark, as illustrated in Figure 5.2.

On a primary window this command is accessed from the Help drop-down menu. This command may also be included as a button on a toolbar or as a command on a pop-up menu for a specific object. This command is described more fully in Step 9.

Menu Bar

A menu bar is used to organize and provide access to actions. It is located horizontally at the top of the window, just below the title bar. A menu bar contains a list of topics or items that, when selected, are displayed on a pull-down menu beneath the choice. A system will typically provide a default set of menu actions that can be augmented by an application. In the past, some platforms have called the menu bar an *action bar*. Menu bar design guidelines were presented in Step 4. The contents of the menu bar and its pull-downs are determined by the application's functionality and the context in which the user is interacting with it.

Figure 5.2: What's This? button.

Status Bar

Information of use to the user can be displayed in a designated screen area or areas. They may be located at the top of the screen in some platforms and called a *status area*, or at the screen bottom. Microsoft recommends the bottom location and refers to this area as the *status bar*. It is also referred to by other platforms as a *message area* or *message bar*.

Microsoft Windows suggests using the status bar to display information about the current state of what is being viewed in the window, descriptive messages about a selected menu or toolbar button, or other noninteractive information. It may also be used to explain menu and control bar items as the items are highlighted by the user.

Scroll Bars

When all display information cannot be presented in a window, the additional information must be found and made visible. This is accomplished by scrolling the display's contents through use of a scroll bar. A scroll bar is an elongated rectangular container consisting of a scroll area or shaft, a slider box or elevator, and arrows or anchors at each end. For vertical scrolling, the scroll bar is positioned at the far right side of the work area, extending its entire length. Horizontal scrolling is accomplished through a scroll bar located at the bottom of the work area. Scroll bars are more fully described in Step 7.

Split Box

A window can be split into two or more pieces or panes by manipulating a *split box* located above a vertical scroll bar or to the left of a horizontal scroll bar. A split box is sometimes referred to as a *split bar*. A window can be split into two or more separate viewing areas that are called *panes*. Splitting a window permits multiple views of an object. A split window enables the user to

- Examine two parts of a document at the same time.
- Display different, yet simultaneous, views of the same information.

To support the splitting of a window that is not presplit by design, include a split box. The split box should be just large enough for the user to successfully target it with the pointer; the default size of a size handle, such as the window's sizing border, is a good guideline.

Toolbar

Toolbars, illustrated in Figure 5.3, are panels or arrays of choices or commands that must be accessed quickly. They are sometimes called *command bars*. Toolbars are designed to provide quick access to specific commands or options. Specialized toolbars are sometimes referred to as *ribbons*, *toolboxes*, *rulers*, or *palettes*. Toolbars may occupy a fixed position on a window, be movable, or be contained in a pop-up window. The design of toolbars is discussed in Step 7.

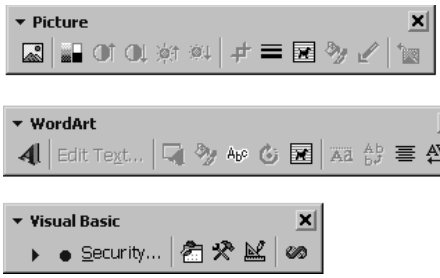


Figure 5.3: Toolbars.

Command Area

In situations where it is useful for a command to be typed into a screen, a command area can be provided. The desired location of the command area is at the bottom of the window. If a horizontal scroll bar is included in the window, position the command area just below it. If a message area is included on the screen, locate the command area just above it.

Size Grip

A size grip is a Microsoft Windows special handle included in a window to permit it to be resized. When the grip is dragged the window resizes, following the same conventions as the sizing border. Three angled parallel lines in the lower-right corner of a window designate the size grip. If the window possesses a status bar, the grip is positioned at the bar's right end. Otherwise, it is located at the bottom of a vertical scroll bar, the right side of a horizontal scroll bar, or the junction point of the two bars. A size grip is shown in the lower-right corner of Figure 5.1.

Work Area

The work area is the portion of the screen where the user performs tasks. It is the open area inside the window's border and contains relevant peripheral screen components such as the menu bar, scroll bars, or message bars. The work area may consist of an open area for typing, or it may contain controls (such as text boxes and list boxes) or customized forms (such as spreadsheets). The work area may also be referred to as the *client area*.

Window Presentation Styles

The presentation style of a window refers to its spatial relationship to other windows. There are two basic styles, commonly called tiled or overlapping. In early windowing days, most systems commonly used one or the other style exclusively, seldom using both at the same time. Now, the user is usually permitted to select the style to be presented on the display.

Tiled Windows

Tiled windows, illustrated in Figure 5.4, derive their name from the common floor or wall tile. Tiled windows appear in one plane on the screen and expand or contract to fill up the display surface, as needed. Most systems provide two-dimensional tiled windows, adjustable in both height and width. Some less-powerful systems, however, are only one-dimensional, the windows being adjustable in only one manner (typically the height). Tiled windows, the first and oldest kind of window, are felt to have these advantages:

- The system usually allocates and positions windows for the user, eliminating the necessity to make positioning decisions.
- Open windows are always visible, eliminating the possibility of them being lost and forgotten.
- Every window is always completely visible, eliminating the possibility of information being hidden.
- They are perceived as less complex than overlapping windows, possibly because there are fewer management operations or they seem less "magical."
- They are easier, according to studies, for novice or inexperienced people to learn and use.
- They yield better user performance for tasks where the data requires little window manipulation to complete the task.

Perceived disadvantages include the following:

- Only a limited number can be displayed in the screen area available.
- As windows are opened or closed, existing windows change in size. This can be annoying.
- As windows change in size or position, the movement can be disconcerting.
- As the number of displayed windows increases, each window can get very tiny.
- The changes in sizes and locations made by the system are difficult to predict.
- The configuration of windows provided by the system may not meet the user's needs.
- They are perceived as crowded and more visually complex because window borders are flush against one another, and they fill up the whole screen. Crowding is accentuated if borders contain scroll bars or control icons. Viewer attention may be drawn to the border, not the data.
- They permit less user control because the system actively manages the windows.

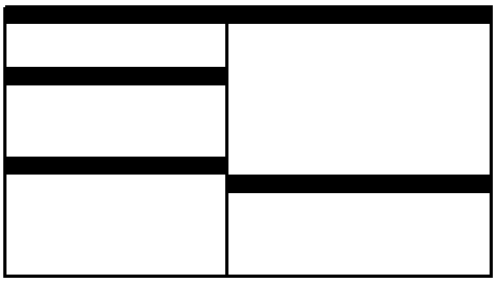


Figure 5.4: Tiled windows.

Overlapping Windows

Overlapping windows, illustrated in Figure 5.5, may be placed on top of one another like papers on a desk. They possess a three-dimensional quality, appearing to lie on different planes. Users can control the location of these windows, as well as the plane in which they appear. The sizes of some types of windows may also be changed. Most systems today normally use this style of window. They have the following advantages:

- Visually, their look is three-dimensional, resembling the desktop that is familiar to the user.
- Greater control allows the user to organize the windows to meet his or her needs.
- Windows can maintain larger sizes.
- Windows can maintain consistent sizes.
- Windows can maintain consistent positions.
- Screen space conservation is not a problem, because windows can be placed on top of one another.
- There is less pressure to close or delete windows no longer needed.
- The possibility exists for less visual crowding and complexity. Larger borders can be maintained around window information, and the window is more clearly set off against its background. Windows can also be expanded to fill the entire display.
- They yield better user performance for tasks where the data requires much window manipulation to complete the task.

Disadvantages include the following:

- They are operationally much more complex than tiled windows. More control functions require greater user attention and manipulation.
- Information in windows can be obscured behind other windows.
- Windows themselves can be lost behind other windows and be presumed not to exist.

- That overlapping windows represent a three-dimensional space is not always realized by the user.
- Control freedom increases the possibility for greater visual complexity and crowding. Too many windows, or improper offsetting, can be visually overwhelming.

Cascading Windows

A special type of overlapping window has the windows automatically arranged in a regular progression. Each window is slightly offset from others, as illustrated in Figure 5.6. Advantages of this approach include the following:

- No window is ever completely hidden.
- Bringing any window to the front is easier.
- It provides simplicity in visual presentation and cleanness.

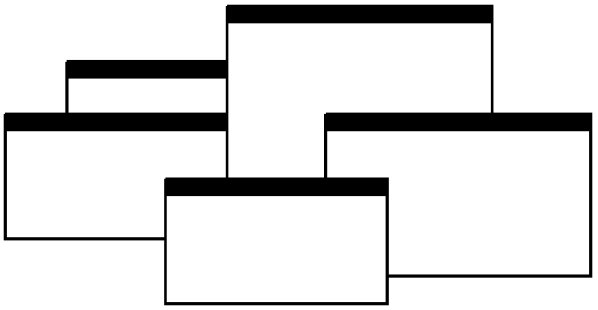


Figure 5.5: Overlapping windows.

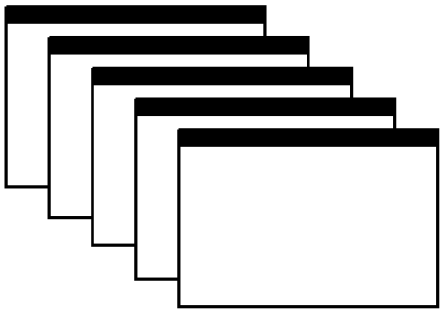


Figure 5.6: Cascading windows.

Picking a Presentation Style

- Use tiled windows for
  - — Single-task activities.
  - — Data that needs to be seen simultaneously.
  - — Tasks requiring little window manipulation.
  - — Novice or inexperienced users.
- Use overlapping windows for
  - — Switching between tasks.
  - — Tasks necessitating a greater amount of window manipulation.
  - — Expert or experienced users.
  - — Unpredictable display contents.

Tiled windows. Tiled windows seem to be better for single-task activities and data that must be seen simultaneously. A study found that tasks requiring little window manipulation were carried out faster using tiled windows. They also found that novice users performed better with tiled windows, regardless of the task.

MAXIM **An activity will be pursued only if its benefits are equal to or greater than the cost.**

Overlapping windows. Overlapping windows seem to be better for situations that necessitate switching between tasks. A research study concluded that tasks requiring much window manipulation could be performed faster with overlapping windows but only if user window expertise existed. For novice users, tasks requiring much window manipulation were carried out faster with tiled windows. Therefore, the advantage to overlapping windows comes only after a certain level of expertise is achieved. Overlapping windows are the preferred presentation scheme.

Types of Windows

People's tasks must be structured into a series of windows. The type of window used will depend on the nature and flow of the task. Defining standard window types is again difficult across platforms because of the varying terminology and definitions used by different windowing systems, and changes in terminology for new versions of systems. For simplicity, the Microsoft Windows windowing scheme will be described. Summarized are a description of the window, its purpose, and its proper usage. Any other platform's windows may not behave exactly as presented, and some platform windows may exhibit characteristics common to more than one of the described window types.

Primary Window

- Proper usage
  - — Use to represent an independent function or application.
  - — Use to present constantly used window components and controls.
    - Menu bar items that are
      - Used frequently.
      - Used by most, or all, primary or secondary windows.
    - Controls used by dependent windows
  - — Use for presenting information that is continually updated, such as date and time.
  - — Use for providing context for dependent windows to be created.
  - — Do not
    - Divide an independent function into two or more primary windows.
    - Present unrelated functions in one primary window.

The *primary* window is the first one that appears on a screen when an activity or action is started. It is required for every function or application, possessing a menu bar and some basic action controls, as previously described. It should present the framework for the function's commands and data, and provide top-level context for dependent windows. It has also been variously referred to as the *application* window or the *main* window. In addition, it may be referred to as the *parent* window if one or more *child* windows exist. A Microsoft Windows primary window is shown in Figure 5.7.

The primary window is the main focal point of the user's activities and should represent an independent function. Avoid dividing an independent function into two or more primary windows, and avoid presenting unrelated functions in a single primary window. This tends to confuse people.

Independent functions should begin in a primary window. A primary window should contain constantly used window components such as frequently used menu bar items and controls (for example, control bars) used by dependent windows. Also include in a primary window continually updated information such as the date and time. The components of a primary window are summarized in Table 5.2.

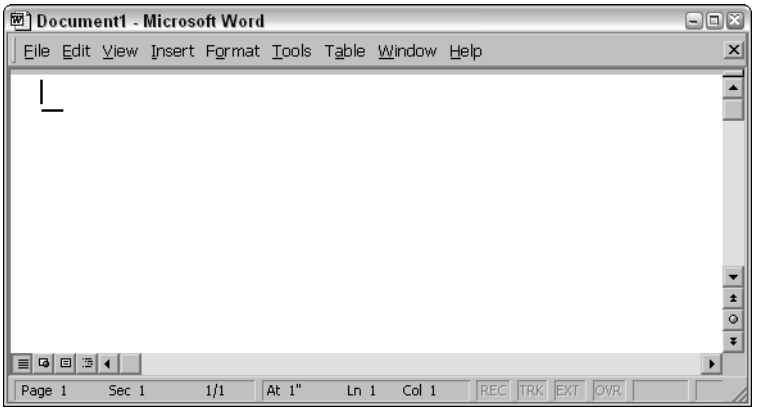


Figure 5.7: Microsoft Windows primary window.

Secondary Windows

- Proper usage:
  - — For performing subordinate, supplemental, or ancillary actions that are
    - Extended or more complex in nature.
    - Related to objects in the primary window.
  - — For presenting frequently or occasionally used window components.
- Important guidelines:
  - — Should typically not appear as an entry on the taskbar.
  - — A secondary window should not be larger than 263 dialog units x 263 dialog units.

Secondary windows are supplemental windows. Secondary windows may be dependent upon a primary window or displayed independently of the primary window. They structurally resemble a primary window, possessing some of the same action controls (Close button) and possibly a What's This? button.

A *dependent* secondary window is one common type. It can only be displayed from a command on the interface of its primary window. It is typically associated with a single data object, and appears on top of the active window when requested. It is movable, and scrollable. If necessary, it uses the primary window's menu bar. Most systems permit the use of multiple secondary windows to complete a task. In general, dependent secondary windows are closed when the primary window closes, and hidden when their primary window is hidden or minimized.

An *independent* secondary window can be opened independently of a primary window — for example, a property sheet displayed when the user clicks the Properties command on the menu of a desktop icon. An independent secondary window can be typically closed without regard to the state of any primary window unless there is an obvious relationship to the primary window.

A Microsoft Windows secondary Window is illustrated in Figure 5.8.

Proper usage. Although secondary windows share many characteristics with primary windows, they also differ from primary windows in behavior and use. Secondary windows are used to perform supplemental or subordinate tasks, or tasks that are extended in nature. Frequently and occasionally used window components should also be presented in secondary windows. Microsoft Windows possesses several types of secondary windows called *dialog boxes*, *property sheets*, *property inspectors*, *message boxes*, *palette windows*, and *pop-up* windows.

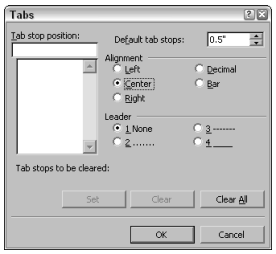


Figure 5.8: Microsoft Windows secondary window.

Guidelines. A secondary window should typically not appear as an entry on the taskbar. Secondary windows obtain or display supplemental information that is usually related to the objects that appear in a primary window.

A secondary window is typically smaller than its associated primary window and smaller than the minimum display resolution. Microsoft recommends not displaying any secondary window larger than 263 dialog units x 263 dialog units. Microsoft defines size and location of user-interface elements not in pixels but in *dialog units* (DLUs), a device-independent unit of measure.

- One horizontal DLU is equal to one-fourth of the average character width for the current system font.
- One vertical DLU is equal to one-eighth of the average character height for the current system font.

These sizes keep the window from becoming too large to display at most resolutions. However, they still provide reasonable space to display supportive information, such as Help information, that applies. The components of a secondary window are summarized in Table 5.2.

Table 5.2: Microsoft Window Types and Components

PRIMARY WINDOW	
Purpose:	To perform a major interaction.
Components:	Frame or border. Title bar. —Access point for commands that apply to the window, with commands displayed in a pop-up menu. Title Bar icon. —Small version of the icon of the object being viewed. —Access point for commands that apply to the object being displayed in the window, with commands displayed in a pop-up window. Title bar text. Title bar buttons to close/minimize/maximize /restore a window. Menu bar. Status bar. Scroll bar. Size grip.
SECONDARY WINDOWS	
Purpose:	To obtain or display supplemental information related to the objects in the primary window.
Components:	Frame or border. Title bar. Title bar text. Close button. What's This? button. —Context-sensitive Help about components displayed in the window; this is optional.

Table 5.2 *(continued)*

SECONDARY WINDOWS	
Kinds:	Modal and modeless.
Dialog Boxes	
Purpose:	To obtain additional information needed to carry out a particular command or task.
Description:	Secondary window. Contains the following common dialog box interfaces: — Open/Replace/Find. — Save As /Print/Print Setup. — Page Setup/Font/Color.
Property Inspectors	
Purpose:	To display the most common or frequently accessed properties of a current selection, usually of a particular type of object.
Description:	A modeless secondary window. Typically modal with respect to the object for which it displays properties.
Usage:	Displayed when requested from selected object.
Property Sheets	
Purpose:	For presenting the complete set of properties for an object.
Description:	A modeless secondary window. Typically modal with respect to the object for which it displays properties.
Usage:	Displayed when requested from selected object.
Message Boxes	
Purpose:	To provide information about a particular situation or condition.
Description:	Secondary window. Types of message boxes: — Information/Warning/Critical.
Palette Windows	
Purpose:	To present a set of controls such as palettes or toolbars.
Description:	Modeless secondary window.
Pop-Up Windows	
Purpose:	To display additional information when an abbreviated form of the information is the main presentation.
Description:	Secondary window. Does not contain standard secondary window components such as title bar and close button. Example: ToolTip.

Modal and Modeless

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

A secondary window can be modal or modeless.

Modal. Most secondary windows will be *modal*. Modal windows will not permit interaction with another window until the current dialog is completed. It remains displayed until the appropriate action is taken, after which it is removed from the screen. Modal dialog boxes typically request critical information or actions that must be reacted to before the dialog can continue. Because modal dialog boxes constrain what the user can do, limit their use to situations in which additional information is required to complete a command or when it is important to prevent any further interaction until satisfying a condition.

Modeless. A *modeless* dialog box permits the user to engage in parallel dialogs. Switching between the box and its associated window is permitted. Other tasks may be performed while a modeless dialog box is displayed, and it may be left on the screen after a response has been made to it. Actions leading to a modeless dialog box can be canceled, causing the box to be removed from the screen.

Use a modeless dialog box when interaction with a primary window or another secondary window must be permitted, for example, during the accessing of the Help function. Also, use a modeless dialog box when interaction with other windows must be repeated; for example, in a word search operation.

Cascading and Unfolding

- Cascading
  - — Purpose:
    - To provide advanced options at a lower level in a complex dialog.
  - — Guidelines:
    - Provide a command button leading to the next dialog box with a "To a Window" indicator, an ellipsis (. . .).
    - Present the additional dialog box in cascaded form.

- Provide no more than two cascades in a given path.
- Do not cover previous critical information.
  — Title Bar.
  — Relevant displayed information.
- If independent, close the secondary window from which it was opened.

■ Unfolding
— Purpose:
- To provide advanced options at the same level in a complex dialog.

— Guidelines:
- Provide a command button with an expanding dialog symbol (>>).
- Expand to right or downward.

Access to additional options can be accomplished by inclusion of a command button that opens another secondary window. These multiple secondary windows needed to complete a task may be presented in two forms, cascading or expanding.

Cascading. A *cascading* window keeps the original window displayed, with the dependent window displayed on top, offset slightly to the right and below the original secondary window. A cascade, illustrated in Figures 5.9 and 5.10, is generally used when advanced options at a lower level in a complex dialog must be presented. An indication that the dialog will be cascading is signaled by an ellipsis placed in the command button used to display the additional dialog box. Because of the confusion that can develop with too many cascades, restrict the number of cascades to no more than two in a given path. Do not cover information on the upper-level dialog boxes that may have to be referred to, such as box title bars and other critical or relevant information. If the cascaded window is independent in its operation, close the secondary window from which it was opened and display only the new window.

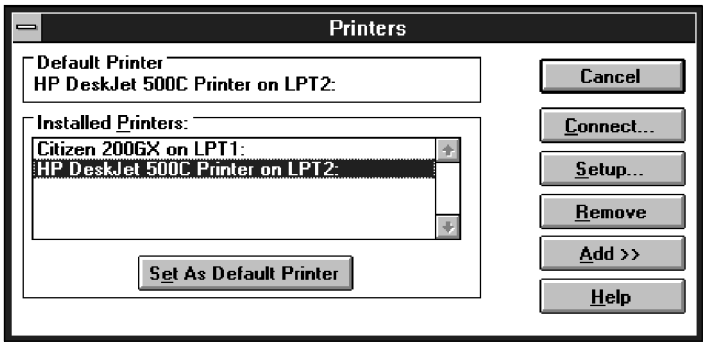


Figure 5.9: Printers secondary window with Connect cascade button.

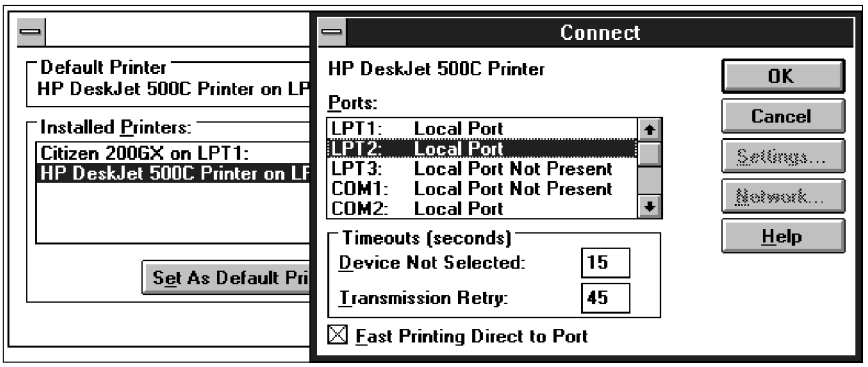


Figure 5.10: Cascading Connect secondary window.

Unfolding. An *unfolding* secondary window expands to reveal additional options, a form of progressive disclosure. Unfolding windows, sometimes called *expanding* windows, are generally used to provide advanced options at the same level in a complex dialog. They are good alternatives when the interface contains a fixed set of options or controls that seldom need to be accessed. An unfolding window is illustrated in Figures 5.11 and 5.12. An indication that the dialog will be expanding is signaled by a double arrow (>>) placed in the command button used to display the additional dialog box. Expand the box to right, preferably, or downward if screen space constraints exist. As an option, the same button can be used to "refold" the additional part of the window.

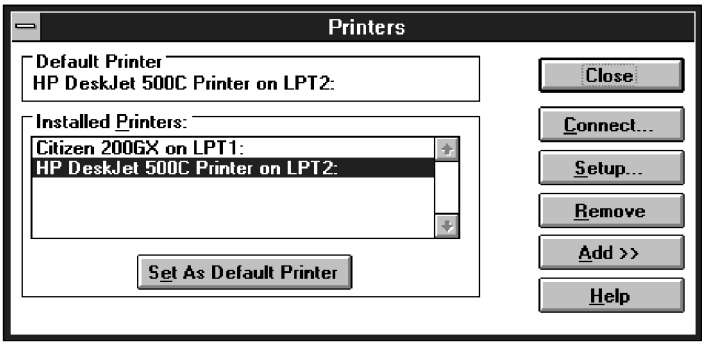


Figure 5.11: Printers secondary window with Add >> unfolding button.

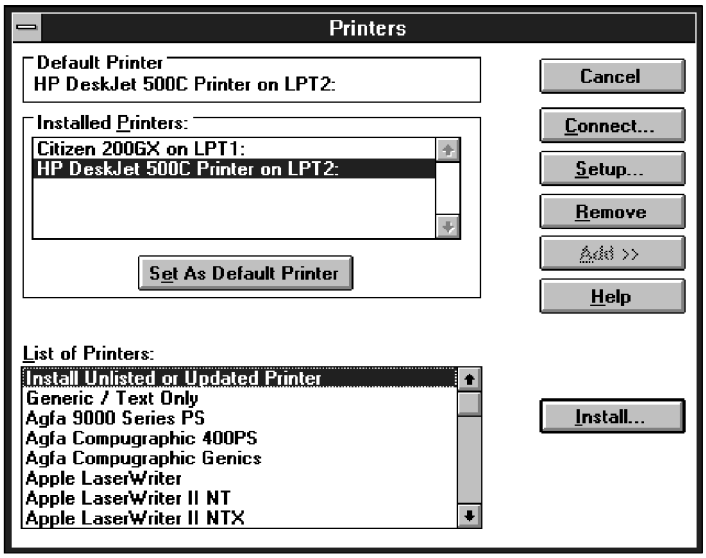


Figure 5.12: Unfolded Printers secondary window.

Dialog Boxes

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

Dialog boxes are used to extend and complete an interaction within a limited context. Dialog boxes are always displayed from another window, either primary or secondary, or another dialog box. They may appear as a result of a command button being activated or a menu choice being selected, or they may be presented automatically by the system when a condition exists that requires the user's attention or additional input. They may possess some basic action controls (Close button and possibly a What's This? button), but do not have a menu bar. A Microsoft Windows dialog box is illustrated in Figure 5.13.

Most windowing systems provide standard dialog boxes for common functions, some examples being Open, Save As, and Print. Many platforms also recommend a set of standard command buttons for use in the various kinds of dialog boxes, such as OK, Cancel, and so on. Dialog boxes are of two types, modal and modeless, as recently described. They may also cascade or unfold.

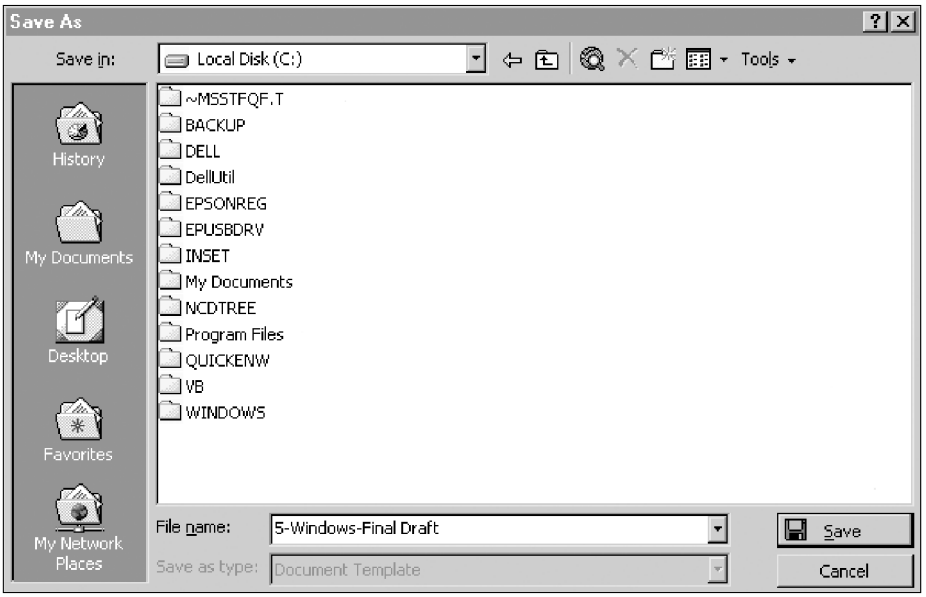


Figure 5.13: Microsoft Windows dialog box.

Uses. Dialog boxes are used for presenting brief amounts of information or to request specific transient actions. Dialog box actions will usually be those that do not occur frequently.

Command buttons. Dialog boxes commonly include OK and Cancel command buttons. OK and Cancel buttons work best in dialog boxes that allow the user to set the parameters for a particular command. OK is typically defined as the default command button when the dialog box window opens. Other command buttons may be included in a dialog box in addition to or instead of the OK and Cancel buttons. Follow the design conventions for command buttons found in Step 7.

Property Sheets and Property Inspectors

The properties of an object in an interface can be displayed in a variety of ways. For example, the image and name of an icon on the desktop reflect specific properties of that object, as do other interface components such as toolbars, status bars, and even scroll bars. Secondary windows provide two other techniques for displaying properties, *property sheets* and *property inspectors.*

Property Sheets

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

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

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

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

Use. A property sheet is the most common way to present an object's complete set of properties in a secondary window. A property sheet is a modeless secondary window that displays the user-accessible properties of an object, properties that may be viewed but not necessarily edited. A single page property sheet is illustrated in Figure 5.14.

Property pages. Because there can be many properties for an object and the object's context, the categorization and grouping of properties within sets may be necessary. A technique for supporting navigation to groups of properties in a property sheet is a tabbed *property page*, where each set of properties is presented within the window as a separate page. Each page tab is labeled with the name of the set, as shown in Figure 5.15. Use tabbed property pages for grouping peer-related property sets. Tabs are described in Step 7.

Size. The sizes recommended for property sheets by Microsoft shown earlier will create a window smaller than its associated window and smaller than the minimum display resolution.

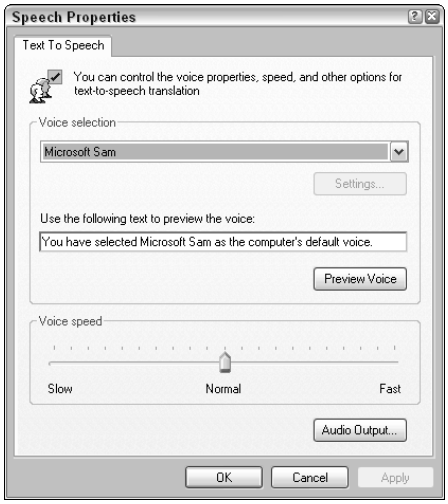


Figure 5.14: Microsoft Windows property sheet.

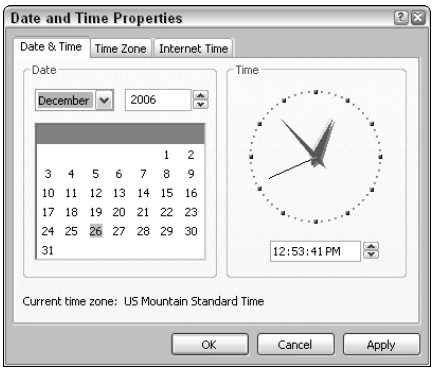


Figure 5.15: Microsoft Windows property sheet tabbed pages.

Command buttons. Property sheets typically allow the values for a property to be changed, and then applied. Include the following common command buttons for handling the application of property changes. For common property sheet transaction buttons, use OK, Cancel, and Apply. A Reset command button to cancel pending changes without closing the window can also be included. Other command buttons can be included in property sheets. Avoid including a Help command button. If a Help button seems necessary, the best solution is to simplify the window.

Command buttons on *tabbed property* pages should be located outside of the tabbed page but still within the window. Buttons placed on a page imply that the action being performed applies *only* to that page. Buttons outside the pages imply the action performed applies to *all* pages. This is the desired positioning because, most often, the tabs are considered by the user as simple grouping or navigation techniques. If the properties are to be applied on a page-by-page basis, however, then place the command and buttons on the property pages, and always in the same location on each page. When the user switches pages without selecting a command button, any property value changes for that page are applied. In these situations, it is useful to prompt the user by displaying a message box that asks whether to apply or discard any changes made.

Property Inspectors

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

Use. Display only the most common or frequently accessed properties in a property inspector. Properties of an object are displayed by using a dynamic viewer or browser that reflects the properties of the current selection. A property inspector differs from a property sheet. Even when a property sheet window is modeless, the window is typically modal with respect to the object for the properties being displayed. If the user selects another object, the property sheet continues to display the properties of the original object. A property inspector, on the other hand, always reflects the current selection.

Figure 5.16: Microsoft Windows property inspector.

A palette window (described shortly) or a toolbar is used to create a property inspector, as shown in Figure 5.16. An even better alternative is to use a palette window that the user can also configure. Another control in a property inspector can be used to enable the user to display the properties of various objects in the primary window. For example, as the first control in the property inspector, include a drop-down list box that displays the name of the object being viewed. To view another object's properties within the inspector, the object is selected in the drop-down list box.

Dynamic changes. Changes a user makes in a property inspector should be made dynamically. That is, the property value in the selected object should be changed as soon as the user makes the change in the related property control.

Property inspectors and property sheets are not exclusive interfaces. Both can be included in an interface. The most common or frequently accessed properties can be displayed in a property inspector and the complete set in the property sheet. Multiple property inspectors can also be included, each optimized for managing certain types of objects. An interface's behavior can also be changed between that of a property sheet and that of a property inspector. A control can be provided that "locks" its view, making it modal to the current object, rather than tracking the entire selection.

Message Boxes

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

Use. A message box, as illustrated in Figure 5.17, is a secondary window that displays a message about a particular situation or condition.

Figure 5.17: Microsoft Windows message box.

Command buttons. Typically, message boxes contain only command buttons with the appropriate responses or choices offered to the user. The command buttons used should allow the message box interaction to be simple and efficient. Microsoft suggests providing the following:

- If a message requires no choices to be made but only acknowledgment, include an OK button and, optionally, a Help button.
- If the message requires the user to make a choice, include a command button for each option.
- Include OK and Cancel buttons only when the user has the option of continuing or stopping the action.
- Use Yes and No buttons when the user must decide how to continue.
- If these choices are too ambiguous, label the command buttons with the names of specific actions, for example, Save and Delete.

Command buttons to correct the action that caused the message box to be displayed can also be included in a message box. For example, if the message box indicates that the user must switch to another application window to take corrective action, a button that opens that application window can also be included.

Stop. If Cancel is used as a command button in a message box, remember that to users, cancel implies that the state of the process or task that started the message is being restored. If you use Cancel to interrupt a process and the state cannot be restored, use Stop instead.

Help. A Help button can be included in a message box for messages needing more detail. This allows the message text to be more succinct.

If other command buttons are needed, consider the potential increase in complexity that their inclusion will cause.

Close box. Enable the title bar Close box only if the message includes a Cancel button. Otherwise, the meaning of the Close operation may be ambiguous.

Default. Designate the most frequent or least destructive option as the default command button.

A definition of message types and guidelines for writing messages are found in Step 8.

Palette Windows

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

Use. Palette windows are modeless secondary windows that present a set of controls, as shown in Figure 5.18. Palette windows are distinguished by their visual appearance, a collection of images, colors or patterns. The title bar for a palette window is shorter and includes only a close button.

Sizing. A palette window can be defined as fixed in size, or, more typically, resizable by the user. Two techniques should indicate when the window is resizable: changing the pointer image to the size pointer, and placing a *size* command in the window's shortcut menu. Preserve the window's size and position so the window can be restored if it, or its associated primary window, is closed and then reopened.

Pop-Up Windows

- Use pop-up windows to
  — Display additional information when an abbreviated form of the information is the main presentation.
  — Collect secondary information.
  — Display textual labels for graphical controls.
  — Display context-sensitive Help information.
- Present the pop-up at the front of the screen.
- Make the pop-up one-quarter to one-third of the window size.
- Provide OK or Save and Cancel buttons.
- Never display unsolicited pop-up windows.

Figure 5.18: Microsoft Windows palette window.

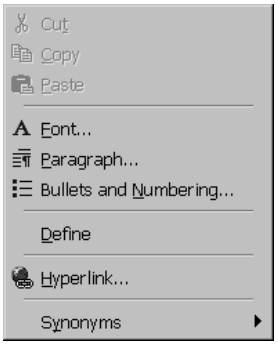


Figure 5.19: Microsoft Windows pop-up window.

Pop-up windows can be used to display additional information when an abbreviated form of the information is the main presentation technique. Other examples of pop-up windows used to display contextual information are *ToolTips* and *balloon tips* that provide the names for controls in graphical toolbars. Pop-up windows are also used to collect secondary information whenever the user's flow through an application should not be interrupted, and to provide context-sensitive Help information.

Always present a pop-up at the front of the screen so it will not be missed, especially if it is reused. The pop-up should be a quarter to a third of a window size. If it is too small it may not be seen; too large and it will cover too much of the screen. Always provide OK or Save and Cancel buttons to remind people of the methods for getting rid of the pop-up. Displaying unsolicited pop-up windows are aggravating and distracting to people. Never display unsolicited windows. Wait for people to take actions necessitating the pop-up.

Organizing Window Functions

Information and functions must be presented to people when and where they need them. Proper organization and support of tasks by windows will be derived only through a thorough and clear analysis of user tasks.

Window Organization

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

- Use dialog boxes for
  — Infrequently used or needed information.
  — "Nice-to-know" information.

People most often think in terms of tasks, not functions or applications. Windows must be organized to support this thinking. The design goal is to support the most common user tasks in the most efficient manner or fewest steps. Less frequently performed tasks are candidates for less efficiency or more steps.

Mayhew (1992) suggests that poor functional organization usually occurs because of one of, or a combination of, these factors:

- Emphasis on technical ease of implementation rather than proper analysis of user tasks.
- Focus on applications, features, functions, or data types instead of tasks.
- Organization of the design team into applications, with little cross-team communication.
- Blindly mimicking the manual world and carrying over manual inefficiencies to the computer system.

Emphasis on implementation ease puts the needs of the designer before the needs of the customer. Focusing on tasks conforms to the model of how people think. Application orientation imposes an unnatural boundary between functions, and lack of cross-team communication seldom yields consistent task procedures. Mimicking "what is" will never permit the capabilities of the computer system to be properly exploited.

MAXIM **The information to make a decision must be there when the decision is needed.**

Recommended usages for the various window types are summarized in the above guidelines. These recommendations were discussed more fully earlier in this chapter.

Number of Windows

- Minimize the number of windows needed to accomplish an objective.

A person does not work with windows simply for the joy of working with windows. Windows are a means to an end, a method of accomplishing something. Multiple windows on a display, as discussed elsewhere in this text, can be confusing, can increase the load on the human visual system, or may be too small to effectively present what needs to be contained within them.

Guidelines that appeared in early stages of window evolution concerning the maximum number of windows that a person could deal with were quite generous, a limit of seven or eight being suggested. As experience with windows has increased, these numbers have gradually fallen. One study found the mean number of windows maintained for experienced users was 3.7. Today, based on expressions of window users, a recommendation of displaying no more than two or three at one time seems most realistic. The guidelines on limitations for items like cascades (1–2) reflect today's feelings. The exact number of windows a person can effectively deal with at one time will ultimately depend on both the capabilities of the user and the characteristics of the task. Some users and situations may permit handling of more than three windows; for other users and situations, three windows may be two too many.

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

Sizing Windows

- Provide large-enough windows to
  - — Present all relevant and expected information for the task.
  - — Avoid hiding important information.
  - — Avoid crowding or visual confusion.
  - — Minimize the need for scrolling.
    - But use less than the full size of the entire screen.
- If a window is too large, determine
  - — Is all the information needed?
  - — Is all the information related?
- Otherwise, make the window as small as possible.
  - — Optimum window sizes:
    - For text, about 12 lines.
    - For alphanumeric information, about 7 lines.

Larger windows seem to have these advantages:

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

Disadvantages include the following:

- Longer pointer movements are required.
- Windows are more crowded.
- More visual scanning is required.

- Other windows more easily obscure parts of the window.
- It is not as easy to hide inappropriate data.

Always provide large enough windows to present all the relevant and expected information for the task. Never hide important or critical information, and minimize the need for scrolling. A study has found that very small windows requiring a significant amount of scrolling appear to increase decision-making time. Scrolling is also a cumbersome operation. To avoid scrolling, consider using unfolding dialog boxes, cascading windows, or a tab control. Avoid, however, making a window's default size the full size of the display. Doing so leads to any underlying windows being completely hidden from the user's view. The option to maximize primary and secondary windows always exists.

If, through analysis and design, a window appears to be too large, determine the following:

- Is all the information needed?
- Is all the information related?

Important, critical, or frequently used information must be maintained on a screen, but perhaps information exists that is needed or used infrequently, for example, only 10 to 20 percent of the time. This kind of information is a good candidate for placement on another window or dialog box. Perhaps information is included on a screen that is not related to the task being performed. This kind of information should be located with the information to which it is related. As a last resort, consider shortening some window control captions or other included window text to achieve a proper fit.

At least two studies have looked at optimum window sizes. Procedural text in window sizes of 6, 12, and 24 lines were evaluated by one study. Fastest and most accurate completion occurred with the 12-line window. The retrieval of alphanumeric information was compared in 7-, 13-, and 19-line windows in another study. A 7-line window was found to be more than adequate.

Window Placement

- Considerations:
  - — In placing a window on the display, consider
    - The use of the window.
    - The overall display dimensions.
    - The reason for the window's appearance.
- General:
  - — Position the window so it is entirely visible.
  - — If the window is being restored, place the window where it last appeared.
  - — If the window is new, and a location has not yet been established, place it
    - At the point of the viewer's attention, usually the location of the pointer or cursor.
    - In a position convenient to navigate to.
    - So that it is not obscuring important or related underlying window information.

— For multiple windows, give each additional window its own unique and discernible location.
  • A cascading presentation is recommended.
— In a multiple-monitor configuration, display the secondary window on the same monitor as its primary window.
— If none of the above location considerations apply, then
  • Horizontally center a secondary window within its primary window just below the title bar, menu bar, and any docked toolbars.
— If the user then moves the window, display it at this new location the next time the user opens the window.
  • Adjust it as necessary to the current display configuration.
— Do not let the user move a window to a position where it cannot be easily repositioned.

■ Dialog boxes:
— If the dialog box relates to the entire system, center it on screen.
— Keep key information on the underlying screen visible.
— If one dialog box calls another, make the new one movable whenever possible.

Considerations. In placing a window on the display, consider how the window is used in relation to other windows, the overall dimensions of the display, and the reason that the window is being presented.

General. First, locate the window so it is fully visible. If the window is being restored, locate it where it last appeared. If the window is new and the location has not yet been established, place it at the point of the viewer's attention. This will usually be the location of the pointer or cursor. Also, place the window in a position where it will be convenient to navigate to, and where it will not obscure important underlying screen information. Preferred positions are essentially below and right. The suggested order of placement is below right, below, right, top right, below left, top, left, top left.

In a multiple-monitor configuration, display the secondary window on the same monitor as its primary window. If none of these situations applies, horizontally center a secondary window within the primary window, just below the title bar, menu bar, and any docked toolbars. Give each additional window its own unique location. A cascading presentation, from upper left to lower right is recommended. If the user then moves the window, display it at this new location the next time the user opens the window, adjusted as necessary for the current display configuration. Do not let the user move a window to a position where it cannot be easily repositioned.

Dialog boxes. If a dialog box relates to the entire system, center it on display, keeping key information on an underlying window visible. If one dialog box calls another, make the new one movable whenever possible.

The Web and the Browser

An entity called the browser is the user interface for the World Wide Web. In structure it resembles a standard window in many aspects, but differs in others. The most popular browser is Microsoft's Internet Explorer, which, as of this writing, is the one utilized by about 83 percent of Web users. The second most popular is Mozilla's Firefox, currently being used by about 11 percent of users. Other available browsers include Netscape, Lynx, Opera, and Safari. This section describes the major components of a browser using Internet Explorer as the model. Some unique features of Firefox will also be described.

Browser Components

In appearance, Internet Explorer presents the same visual style found in Windows. It also contains many of the same standard components, including a title bar, menu bar, toolbar, sizing buttons, size grip, command field (called URL address bar), status bar, enclosing frame border, and if necessary, a scroll bar. It also includes a content area for displaying information, buttons, data fields, and so forth.

Standard toolbar actions include the following:

- Back - Displays the previous page viewed.
- Forward - Displays the next page in the viewing sequence if already viewed.
- Stop - Stops page from being loaded.
- Refresh - Refreshes and redisplays the page being viewed.
- Home - Redisplays the Web site homepage.
- Search - Displays a Search field.
- Favorites - Displays a listing of favorite URLs that have been saved.
- History - Displays a history of viewed pages.

Mozilla Firefox, the second most popular browser, visually resembles Internet Explorer but provides one unique feature called *tabs*. With the tabs feature, multiple Web pages can be displayed on different tabs, resulting in fast and easy movement between each page.

Content Area

The content area of a window is discussed in Step 3, and will be further addressed in Step 13. By way of a brief summary, the content area will normally contain the following:

- A global navigation panel at the top.
- A local navigation panel on the left.
- A bottom navigation panel.
- Information, data fields, buttons, and so forth.

The content area may also contain limited windowing capabilities. The frame concept does provide window-like ability, and JavaScript does provide pop-up windows.

Frames

- Description:
  - — Multiple Web screen panes that permit the displaying of multiple documents on a page.
  - — These documents can be independently viewed, scrolled, and updated.
  - — The documents are presented in a tiled format.
- Proper usage:
  - — For content expected to change frequently.
  - — To allow users to change partial screen content.
  - — To permit users to compare multiple pieces of information.
- Guidelines:
  - — Use only a few frames (three or less) at a given time.
  - — Choose sizes based upon the type of information to be presented.
  - — Never force viewers to resize frames to see information.
  - — Never use more than one scrolling region on a page.

Description. Historically, the Web is essentially a single page (or, by analogy, a single window) entity. While providing significant interface benefits, it is also a reversal of the interface evolution process that led from single-screen technology to windowing. To counteract this shortcoming, *frames* were created. A frame is an independent pane of information presented in a Web page, or, again by analogy, as multiple windows. Frames, however, are presented as tiled, with no overlapping capability. The interaction richness, support, and contextual cues provided by overlapping windows are lacking. Frames, then, allow the displaying of multiple documents on a single Web page. These multiple documents can be independently viewed, scrolled, and updated.

Proper usage. Frames are useful in situations where portions of the page content are expected to change frequently. The volatile information can be separated from other page content and placed within a frame, thereby requiring only a portion of the page's content to be modified. Frames are also useful for allowing the user to change page content; navigation links can be placed in one frame and the resulting content displayed within another frame. As different links are selected, the content in the related frame changes. Frames more effectively allow users to compare multiple pieces of related information displayed within the different frames.

Advantages and disadvantages. Frames, like most interface entities, have advantages and disadvantages. At this moment in their existence, the disadvantages seem to outweigh the advantages. These disadvantages, however, are being whittled away as Web technology advances.

Frames, *advantages* mostly cluster around their ability to reduce the user's content comprehension mental load. These include the following:

- They decrease the user's need to jump back and forth between screens, thereby reducing navigation-related cognitive overhead.
- They increase the user's opportunity to request, view, and compare multiple sources of information.
- They allow content pages to be developed independently of navigation pages.

The *disadvantages* mostly cluster around navigational shortcomings, including

- The difference between a single Web page and a page with frames is not always obvious to the user.
- They suffer some of the shortcomings of tiled screens.
- Only a limited number can be displayed in the available screen area.
- They are perceived as crowded and more visually complex because frame borders are flush against one another and they fill up the whole screen. Crowding is accentuated if the borders contain scroll bars and/or control icons. Viewer attention may be drawn to the border, not the data.
- Frames-based pages behave differently from regular Web pages.
- Page-printing difficulties and problems can exist.
- Page interaction can be clumsy.
- URLs cannot be e-mailed to other users.
- Frames will not work on older browsers.

Past problems, now being addressed and mostly solved, have included difficulties in bookmarking pages, difficulties in creating browser history lists, and inconsistencies in behavior of the browser's Back button.

Guidelines. Guidelines to consider in using frames are the following: Use no more than three frames at a time. Using more will shrink each frame's usable area to the point where little space will be available for presenting content. Then, users will not be able to see much and be forced to scroll. Choose frame sizes based upon the type of information you want to present. Navigational links, for example, should be presented in a small frame and the corresponding information in a larger adjacent frame. Never force people to resize frames to see information. If people feel they must resize frames, the page design is poor. Do not use more than one scrolling region in frames contained on a page. This may be confusing to many users.

Technological advances in frames will continue to occur. Knowledge related to frame usability will also advance. Always be aware of the latest developments.

Pop-Up Windows

- Be extremely cautious in the use of pop-up windows.

JavaScript pop-up windows began appearing on the Web in 1996. Their use is multiplying and, in the view of almost all Web users, polluting screens. Because they are most frequently used in advertising, they have become a source of great aggravation to almost every user. Anecdotal evidence suggests that when a pop-up window begins to appear, most people close them before they are rendered. So, if a pop-up window is used, it may never be completely seen or read by the user. Use them with extreme caution.

Step 5 Exercise

An exercise for Step 5 can be found on this book's companion Web site, www.wiley.com/college/galitz.

STEP

6

Select the Proper Interaction Devices

Interaction devices are the input mechanisms or devices through which people communicate their needs and desires to the computer, and the output mechanisms or devices through which the computer responds to people. The fast-paced evolution of computer systems has seen greatly expanded families of devices to assist and enhance this communication. The distinction between input and output devices is not always clear-cut, however. The common keyboard, for example, is essentially a device for keying information, commands, and so forth. But the keyboard also has an output component. A properly designed keyboard provides feedback when a key is pressed; an audible click and a tactile resistance change when it activates or bottoms-out. For this discussion, devices will be categorized and reviewed based upon their primary purpose — input or output.

Input Devices

For years the device of choice in display-based systems was the standard keyboard and some human-engineered variations. As graphical systems evolved, emphasis shifted to another device, the mouse and some of its cousins: the trackball and joystick. These new mechanisms are most commonly referred to as pointing devices. A few other devices have also been around and have seen extended service through the years: the light pen and the graphic tablet. Some unique human devices also exist: touch and voice. A finger has been used in conjunction with touch-sensitive screens. Our vocal chords are being harnessed to speak meaningfully to the computer, not simply to shout words of exasperation. These various alternatives have both strengths and

weaknesses. Selecting the proper device-based control to do the required job is critical to system success. A good fit between user and control will lead to fast, accurate performance. A poor fit will result in lower productivity, produce more errors, and increase user fatigue. The characteristics and capabilities of various input devices will be reviewed including the following:

- Trackball
- Joystick
- Graphic tablet or trackpad
- Touch screen
- Light pen
- Voice
- Mouse
- Keyboard

We'll begin by reviewing the kinds of input tasks being performed. We'll discuss each device and identify its advantages and disadvantages. Then, we'll focus on the most popular control, the mouse, describing it in more detail and presenting a series of design guidelines for its use. The keyboard, because of its versatility and usefulness for text entry tasks, will also be examined in more detail. Finally, pertinent research will be reviewed and guidelines presented to aid in selecting the proper device.

Characteristics of Input Devices

Several specific tasks are performed using today's systems:

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

The various devices vary in how well they can perform these actions. Among the considerations to be reviewed are two very important factors. First, is the mechanism a direct or indirect pointing device? *Direct* devices are operated on the screen itself. Examples include the light pen, the finger, and voice. *Indirect* devices are operated in a location other than the screen, most often on the desktop. Examples include the mouse, trackball, and keyboard. The psychomotor skills involved in learning to use, and using, a direct device are much simpler than those required for an indirect device. Most of these direct device skills were instilled in our formative years.

Input devices are either discrete or continuous in their action. *Discrete* devices are used to enter individual bits of information — letters, numbers, or commands. *Continuous* input devices operate sequentially in nature — best exemplified by tasks such as dragging or drawing.

Another consideration is what is the relationship between movement of the hand-operated device and the corresponding pointer movement on the screen in terms of *direction*, *distance*, and *speed*? Does the pointer movement track control movement exactly or does it not? The mouse achieves a coupled relationship in all three aspects of direction, distance, and speed: The pointer on the screen moves in the direction the mouse is pushed, at the speed the mouse is pushed, and the distance the mouse is pushed (there may be some ratios applied). A trackball does not achieve this relationship in all three aspects. The pointer moves the direction the trackball is turned and the speed the ball is turned, but not the distance the ball is moved because the ball does not move forward or backward; its socket is stationary. Devices possessing coupled relationships in these three aspects require less psychomotor skill learning than those not possessing a coupled relationship in all three aspects.

Trackball

- Description:
  - — A spherical object (ball) that rotates freely in all directions in its socket.
  - — Direction and speed is tracked and translated into cursor movement.
- Advantages:
  - — Direct relationship between hand and pointer movement in terms of direction and speed.
  - — Does not obscure vision of screen.
  - — Does not require additional desk space (if mounted on keyboard).
- Disadvantages:
  - — Movement is indirect, in a plane different from the screen.
  - — No direct relationship exists between hand and pointer movement in terms of distance.
  - — Requires a degree of eye-hand coordination.
  - — Requires hand to be removed from keyboard keys.
  - — Requires different hand movements.
  - — Requires hand to be removed from keyboard (if not mounted on keyboard).
  - — Requires additional desk space (if not mounted on keyboard).
  - — May be difficult to control.
  - — May be fatiguing to use over extended time.

Description. Commonly used with notebook PCs, the trackball is a ball that rotates freely in all directions in its socket. Essentially it is an inverted mouse. The ball is rotated with one's fingertips, and its direction and speed are tracked and translated into equivalent screen cursor movement. Trackballs are well suited for navigational control, as in video games or exploration of 3-D environments. In these tasks, smooth movement is more important than fine target acquisition. A miniature trackball may also be mounted on a mouse, as is done by Apple.

Advantages. In terms of direction and speed, a trackball possesses a direct relationship between how it is rolled and how the cursor moves on the screen. The cursor moves in the same direction and speed ratio as the ball is rotated. Many trackballs

are mounted on the keyboard itself, permitting the user's hands to remain close to the keys. Trackballs on the keyboard do not require additional desk space, although the keyboard must often be expanded to allow for their inclusion. People with limited finger movement may find it easier to manipulate a trackball than a mouse. Trackballs are available in a variety of sizes and shapes to satisfy varying user needs.

Disadvantages. Trackballs share a common problem with several other controls: Control movement is in a different plane from the screen, or indirect. The cursor, or pointer, is separated from the control itself — the pointer being on the screen, the control on the keyboard. To effectively use a trackball requires learning the proper psychomotor skills, fine finger movements for accurate pointing, and gross hand movements for moving longer distances. The fine finger movements necessary to use them can be difficult to perform. Over longer periods of use, they can be fatiguing. When paired with keyboard tasks, they require a shift in motor activity from keystrokes to finger/hand movement.

Joystick

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

Description. A joystick, like its aircraft namesake, is a stick or bat-shaped device usually anchored at the bottom. They come in variable sizes, smaller ones being operated by fingers, larger ones requiring the whole hand. The smaller joysticks require fine motor coordination, the larger ones more gross coordination. Some,

called *force* joysticks, are immovable, responding to pressure exerted against them. The direction and amount of pressure is translated into pointer movement direction and speed. Others, called *movable* joysticks, can be moved within a dish-shaped area. The direction and distance of the movements create a similar pointer movement on the screen. Some kinds of joysticks permit continuous movements, others only horizontal and vertical movements. Some joysticks may be mounted on the keyboard. They are also well suited for navigational control where smooth movement is most important. Small joysticks, sometimes called "eraser heads" may be found on Notebook computers.

Advantages. Joysticks typically possess a direct relationship between hand and cursor movement in terms of direction. They do not obscure vision of the screen and, when mounted on the keyboard, do not require additional desk space.

Disadvantages. Joysticks are also indirect devices, the control and its result being located in different planes. They require developing a skill to use and can be slow and inaccurate. Use over extended time may also be fatiguing. When paired with keyboard tasks, they require a shift in motor activity from keystrokes to finger/hand movement.

Graphic Tablet or Trackpad

- Description:
  - — Pressure-, heat-, light-, or light-blockage-sensitive horizontal surfaces that lie on the desktop or keyboard.
  - — May be operated with fingers, light pen, or objects like a stylus or pencil.
  - — Pointer imitates movements on tablet.
- Advantages:
  - — Direct relationship between touch movements and pointer movements in terms of direction, distance, and speed.
  - — More comfortable horizontal operating plane.
  - — Does not obscure vision of screen.
- Disadvantages:
  - — Movement is indirect, in a plane different from screen.
  - — Requires hand to be removed from keyboard.
  - — Requires hand to be removed from keyboard keys.
  - — Requires different hand movements to use.
  - — Requires additional desk space.
  - — Finger may be too large for accuracy with small objects

Description. A graphic tablet, also called a "trackpad," "touch tablet," "touchpad," or simply "tablet," is a device with a horizontal surface sensitive to pressure, heat, light, or the blockage of light. It may lie on the desk or may be incorporated on a keyboard, and it is operated with fingers, light pen, or objects like a pencil or stylus. The screen pointer imitates movement on the tablet.

Advantages. With graphic tablets, a direct relationship exists between touch movements and pointer movements in terms of direction, distance, and speed. The screen mimics the tablet. When used with objects like styluses, light pens, or pencils, the operational angle, horizontal, is more comfortable than those vertically oriented.

Disadvantages. Tablets are also indirect controls, creating coordination problems. To use them requires moving one's hand from the keyboard and, if using another device, picking it up. If the finger is the tablet-activation object, accuracy with small objects is difficult. Tablets may also require desk space.

Trackpads operated with one's finger are increasingly found on Notebook PCs. Learning to use a touchpad is more difficult than learning to use a mouse and people with limited finger dexterity frequently find them difficult to operate.

Touch Screen

- Description:
  — A special surface on the screen sensitive to finger or stylus touch.
- Advantages:
  — Direct relationship between hand and pointer location in terms of direction, distance, and speed.
  — Movement is direct, in the same plane as screen.
  — Requires no additional desk space.
  — Stands up well in high-use environments.
- Disadvantages:
  — Finger may obscure part of screen.
  — Finger may be too large for accuracy with small objects.
  — Requires moving the hand far from the keyboard to use.
  — Very fatiguing to use for extended period of time.
  — May soil or damage the screen.
- Design Guidelines:
  — Screen objects should be at least ¾ × ¾ inches in size.
  — Object separation should be at least ⅛ inch.
  — Provide visual feedback in response to activation. Auditory feedback may also be appropriate.
  — When the consequences are destructive, require confirmation after selection to eliminate inadvertent selection.
  — Provide an instructional invitation to begin using.

Description. A touch screen is a screen that consists of a special surface sensitive to finger or stylus touch. Objects on the screen are pointed to and touched to select them.

Advantages. Touch screens possess a direct relationship between hand and pointer movement in terms of direction, distance, and speed. This relationship is direct, not indirect, because the control (finger or stylus) is on the same plane as the

pointer. Another significant advantage of a touch screen is that it does not require any additional desk space.

Disadvantages. A disadvantage of touch screens is that they are fatiguing to use over an extended period of time. If a finger is the touch mechanism, it may obscure part of the screen and be too large to be accurate with small objects. A stylus is usually more accurate than the finger. Fingers may also soil the screen, and a stylus may damage it. Both finger and stylus require moving a hand from the keyboard, and if a stylus is used, it must also be picked up.

Guidelines. When using touch screens, larger screen objects should always be provided to foster accuracy in use. Objects should be ¾ inches square at a minimum and separated by at least ⅛ inch. Visual, and perhaps auditory, feedback should be provided in response to activation. When the consequences of selection are destructive, require a confirmation to avoid inadvertent selection. Observational research indicates that touch screen devices placed in public places, for use by the general public, should possess an instructional invitation to begin their use.

Today other forms of touch screen devices are being used. One type allows placement of a finger on the screen without item selection, selection being accomplished by lifting the finger off the screen. This may allow more accurate item selection. Another method involves placing a cursor on the screen directly above one's finger and moving the cursor as the finger is moved. The cursor permits better target visibility, as well as the detection of smaller targets.

Light Pen

- Description:
  — A special surface on a screen sensitive to the touch of a special stylus or pen.
- Advantages:
  — Direct relationship between hand and pointer movement in terms of direction, distance, and speed.
  — Movement is direct, in the same plane as screen.
  — Requires minimal additional desk space.
  — Stands up well in high-use environments.
  — More accurate than finger touching.
- Disadvantages:
  — Hand may obscure part of screen.
  — Requires picking it up to use.
  — Requires moving the hand far from the keyboard to use.
  — Very fatiguing to use for extended period of time.

Description. A light pen, or pen, also utilizes a touch screen, but one that is sensitive in a specific way to one kind of pen or stylus. Advantages and disadvantages are similar to those of the touch screen.

Advantages. Light pens possess a direct relationship between hand and pointer movement in terms of direction, distance and speed, and are also classified as

direct pointing devices because the control (pen or stylus) is on the same plane as the pointer. Another advantage of a light pen is that it does not require any additional desk space, except for a place for the pen to rest. A light pen or stylus is usually more accurate than the finger. Stylus pens are useful for the very small screens found on personal digital assistants (PDAs) because of their small pointer size.

Disadvantages. A disadvantage is that they are also fatiguing to use over an extended period of time. Light pens require moving a hand from the keyboard to pick up and use.

Voice

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

Description. Automatic speech recognition technology has been under development for more than a quarter of a century. Its progress has been hindered by the following disadvantages listed. Recently, however, it has evolved to the point where telephone-answering systems can now handle simple voice input.

Advantages. Speech is a simple and direct communication medium. It is very useful for people who cannot use a keyboard, or whose hands are otherwise occupied.

Disadvantages. Speech recognition errors are fundamentally different from keying errors. Most keying errors result from a user's inability to always press the correct key. Most speech recognition errors result from the computer speech recognizers' inability to correctly recognize words. People can dictate to a computer at a fairly fast rate, about 105 words per minute (Karat et al., 1999; Lewis, 1999). After making the required corrections, the input rate becomes about 25 words per minute when transcribing the input. New users had even lower transcribing rates. As summarized in Step 1, typists, even those of the two-finger variety, have much higher keying rates. Error correction also takes much longer with a speech recognition system. The most commonly used correction methods are deleting and repeating the last phrase; deleting and repeating a specific word; deleting and selecting a correct word from a list of alternative words; and retyping the selection.

Several research studies have shown that correcting voice recognition errors using a method other than additional voice recognition speeds up the correction process. Suhm, Myers, and Waibel (1999) found that fast typists made almost three times more corrections per minute than people who made corrections by voice only. Lewis (1999) and Karat et al. (1999) uncovered very similar results.

Speech recognition is also, of course, difficult to utilize in an improper environment. Noise can hinder the process, and it is very impractical, and disturbing, to try and use it in a very quiet location such as a library.

Mouse

- Description:
  - — A rectangular or dome-shaped, movable, desktop control containing from one to three buttons used to manipulate objects and information on the screen.
  - — Movement of screen pointer mimics the mouse movement.
- Advantages:
  - — Direct relationship between hand and pointer movement in terms of direction, distance, and speed.
  - — Permits a comfortable hand resting position.
  - — Selection mechanisms are included on mouse.
  - — Does not obscure vision of the screen.
- Disadvantages:
  - — Movement is indirect, in a plane different from screen.
  - — Requires hand to be removed from keyboard.
  - — Requires additional desk space.
  - — May require long movement distances.
  - — Requires a degree of eye-hand coordination.

Description. A mouse is a rectangular or dome-shaped, movable, desktop control typically possessing one, two, or three buttons, but may include more. It is used to manipulate objects and information on the screen. The movement of the screen pointer mimics the mouse movement. It is also common for a mouse to possess a scroll wheel and, in the case of Apple, a miniature trackball that acts as an omnidirectional scroll device. In the mid 1960s, Doug Engelbart, a researcher at the Stanford Research Institute, invented what became the mouse. While using a trackball, he was inspired to turn it upside down and let the ball become the bottom of a control that, attached to a cord, was moved across the desk. It was patented as the "x-y position indicator," and finally christened the "mouse" by a colleague of Engelbart's (a colleague whose name is lost in time). In 1997, Engelbart was at long last rewarded for his invention when he received the annual Lemelson-Mit Prize for American Innovation, including a well-deserved and very substantial monetary reward (cnn.com, 1997).

Advantages. There is a direct relationship between hand and pointer movement in terms of direction, distance, and speed. The mouse itself contains some basic controls (buttons) useful for manipulating screen objects. The hand position when

using the mouse is generally fairly comfortable, and the mouse does not obscure the screen.

Disadvantages. Disadvantages are that they are also indirect devices, the control and its result being located in different planes. They require developing a skill to use and, when paired with keyboard tasks, they require movement away from the keyboard and a shift in motor activity from keystrokes to finger/hand movement. The mouse also requires extensive additional desk space and long positioning movements. The mouse comes in a variety of configurations, performs some basic functions, and is operated in several ways.

Configurations. A mouse may possess one, two, or three buttons. Most, but not all, windowing systems permit operation using all configurations. Buttons are used to perform three functions to be described. When three mouse buttons are not available, the pointer location or keyboard qualifiers must be used to determine the function to be performed. A multibutton mouse permits a more efficient operation, but a person must remember which button to use to perform each function. A multibutton mouse may usually be configured for left- or right-hand use.

Functions. The functions performed by a mouse are Select, Menu, and Adjust. The *Select* function is used to manipulate controls, to select alternatives and data, and to select objects that will be affected by actions that follow. Select is a mouse's most important function and is the function assigned to a one-button mouse. For a multibutton mouse, it is usually assigned to the leftmost button (assuming a right-handed operation).

The *Menu* function is typically used to request and display a pop-up menu on a screen. A menu appears when the button is depressed within a particular defined area of the screen. This area may be, for example, the entire screen, within a window, or on a window border. This button eliminates the need for a control icon, which must be pointed at and selected. The user, however, must remember that a menu is available. The *Adjust* function extends or reduces the number of items selected. It is the least used of the three functions and is usually assigned last and given the least prominent location on a mouse.

Operations. Several operations can be performed with a mouse. The first, point, is the movement and positioning of the mouse pointer over the desired screen object. It prepares for a selection or control operation. To press is to hold the button down without releasing it. It identifies the object to be selected.

To *click* is to press and immediately release a button without moving the mouse. This operation typically selects an item or insertion point, operates a control, or activates an inactive window or control. To *double-click* is to perform two clicks within a predefined time limit without moving the mouse. It is used as a shortcut for common operations such as activating an icon or opening a file.

To *drag* is to press and hold the button down, and then move the pointer in the appropriate direction. It identifies a range of objects or moves or resizes items. To *double-drag* is to perform two clicks and hold the button down, and then move the pointer in the appropriate direction. It identifies a selection by a larger unit, such as a group of words.

Mouse Usage Guidelines

- Provide a "hot zone" around small or thin objects that might require extremely fine mouse positioning.
- Never use double-clicks or double-drags as the only means of carrying out essential operations.
- Do not use mouse plus keystroke combinations.
- Do not require a person to point at a moving target.

If an object is very small and might require fine mouse positioning, provide a large "hot zone" around it. This will increase target size and speed selection. Do not require double-clicks or double-drags as the only way to carry out essential operations. Rapid double-pressing is difficult for some people. Do not use mouse plus keystroke combinations to accomplish actions. This can be awkward. One exception: multiple selections of items in a list. Do not require a person to point at a moving target, except, of course, for a game.

Keyboard

- Description:
  — Standard typewriter keyboard and cursor movement keys.
- Advantages:
  — Familiar.
  — Accurate.
  — Does not take up additional desk space.
  — Very useful for
    - Entering text and alphanumeric data.
    - Editing text and alphanumeric data.
    - Keyed shortcuts — accelerators.
    - Keyboard mnemonics — equivalents.
  — Advantageous for
    - Performing actions when less than three mouse buttons exist.
    - Use with very large screens.
    - Touch typists.
- Disadvantages:
  — Slow for non-touch-typists.
  — Can be over-elaborate.
  — Slower than other devices in pointing.
  — Requires discrete actions to operate.
  — No direct relationship between finger or hand movement on the keys and cursor movement on screen in terms of speed and distance.

Description. A keyboard is a discrete input device. Christopher Latham Sholes invented the standard typewriter keyboard in 1870. Commonly called the QWERTY layout, Sholes' placement of letters was intended to slow down a

typist's keying movements so that the potential for key jams was minimized. From a strictly human-engineering perspective, its layout inadequacies included a dominance of the left hand in making keystrokes, frequent successive keystrokes with the same hand, frequent movement between keyboard key rows, and frequently used letter pairs being placed far from each other. In 1936, August Dvorak created a revised and well-human-engineered keyboard that overcame many of these deficiencies. The advantages of the Dvorak layout, as it came to be called, included a right-hand dominance in keying, much less frequent row changes, and more systematic alternation between the right and left hand. With this new layout, finger travel distances were reduced by at least one order of magnitude. Acceptance of this new keyboard was, and continues, to be slow. Most people have seemed unwilling to invest the time and effort to change.

In the 1980s IBM performed a series of studies comparing the QWERTY keyboard with various sequential key layouts such as ABCDEF or JIHGFE (starting from the upper-left or QWERTY location). IBM wanted to determine if a sequential layout was better for users who professed to be non-touch-typists. Their findings were surprising. Non-touch-typist performance results were as good, or better, using the QWERTY layout as using the various systematic layouts. IBM's conclusion: Why change? So they didn't. IBM researchers could only speculate about why the new systematic layouts fared so poorly. Perhaps, they said, while non-touch-typists profess no knowledge of the QWERTY layout, through experience they have learned the layout, at least well enough to permit effective two-finger typing to be accomplished. Another possibility, they said, was that perhaps some characteristic of the QWERTY layout makes it easy to scan and find needed keys. Speculation number one seems to be the most reasonable explanation, but it may never be known for sure.

Advantages. The standard keyboard is familiar, accurate, and does not consume additional desk space. It is useful and efficient for entering or inserting text or alphanumeric data. For tasks requiring heavy text or data entry, shifting the hands between a keyboard and an alternative control, such as a mouse, can be time-consuming and inefficient, especially for a touch typist. The keyboard is flexible enough to accept keyed shortcuts, either keyboard accelerators or mnemonic equivalents. Some systems also permit navigation across a screen through use of keyboard keys such as the space bar, arrows, Tab, and Enter. Inefficiencies in using other graphical device-based controls can occur, making it advantageous to use a keyboard. A mouse with a limited number of buttons will require use of the keyboard to accomplish some functions, possibly causing frequent shifting between devices. When operations are being performed on very large screens, the user may also find keyboard window management preferable to the long mouse movements frequently required. Therefore, to compensate for these possible inefficiencies, many windowing systems provide alternative keyboard operations for mouse tasks.

Disadvantages. Disadvantages of a keyboard include its requiring discrete finger actions to operate instead of the more fine positioning movements. As a result, no direct relationship exists in terms of speed and distance between finger or hand movement on the keys and cursor movement on the screen. Depending on the

layout of the keyboard cursor control keys, direct-relationship direction problems may also exist, because fingers may not move in the same direction as the cursor. Keyboards will also be slower for non-touch-typists and slower than other controls in pointing tasks. Unfortunately, many keyboards are over-elaborated in design, possessing many keys that are rarely, if ever, used by most people.

Keyboard Guidelines

- Provide keyboard accelerators.
  - — Assign single keys for frequently performed, small-scale tasks.
  - — Use standard platform accelerators.
  - — Assign Shift+key combinations for actions that extend or are complementary to the actions of the key or key combination used without the Shift+key.
  - — Assign Ctrl+key combinations for
    - Infrequent actions.
    - Tasks that represent larger-scale versions of the task assigned to the unmodified key.
- Provide keyboard equivalents.
  - — Use standard platform equivalents.
  - — Use the first letter of the item description.
  - — If first letter conflicts exist, use
    - Another distinctive consonant in the item description.
    - A vowel in the item description.
- Provide window navigation through use of keyboard keys.

Keyboard accelerators. Accelerators provide a way to access menu elements without displaying a menu. They are useful for frequent tasks performed by experienced users. Keys assigned for accelerators should foster efficient performance and be meaningful and conceptually consistent to aid learning. Use standard accelerators as shown in Table 4.2 in Step 4. Microsoft suggests that frequently performed, small-scale tasks should be assigned single keys as the keyboard alternative. Actions that extend or are complementary to the actions of a key (or key combination) should be assigned a Shift key in conjunction with the original action. Microsoft, for example, uses a single key, F6, as the key to move clockwise to the next pane of an active window. To move counterclockwise to the next pane, use Shift+F6. Infrequent actions, or tasks that represent larger-scale versions of the task assigned to the unmodified key, should be assigned Ctrl+key combinations. The left arrow key in Microsoft Windows, for example, moves the cursor one character; Ctrl+left arrow moves it one word.

Keyboard equivalents. Keyboard mnemonics enable the selection of a menu choice through the keyboard instead of by pointing. This enables a person's hands to remain on the keyboard during extensive keying tasks. Keyboard mnemonics should be chosen in a meaningful way to aid memorability and foster predictability of those things that may be forgotten. Mnemonics need only be unique within a menu. A simple rule is always to use the first letter of a menu item description. If the first letter of one item conflicts with that of another, choose

another distinctive consonant in the item description, preferably, but not always necessarily, the second in the item word (occasionally another consonant may be more meaningful). The last choice would be a vowel in the item description. If standard platform equivalents are available, use them. Standard equivalents are shown in Table 4.1 in Step 4.

Window navigation. Also provide ways of navigating through windows by the use of keyboard keys.

MAXIM **The greater the effort to accomplish a task, the less likely that the task will be accomplished successfully.**

Other Input Devices

As technology advances, other types of input devices are in various forms of development. *Gesture* recognition involves understanding the gestures people use in day-to-day communication. Through hand movements and facial expressions people can communicate moods, feelings, and actions. (A smile is recognized as OK, a frown as cancel. A downward pointed finger means stop. A turn of the head left means back.) Imagine the possibilities.

Eye tracking can be used to control an interaction. The cursor is moved to where the user is looking. An action is performed through looking at the proper button. *Iris* and *fingerprint* recognition can be used to verify that you are who you say you are. *Handwriting* is accurately recognized and can be used to enter data or information. Some day these devices may be as common as the mouse or keyboard.

Selecting the Proper Input Device

Many studies have been performed comparing the various input devices for assorted office tasks. Significant findings include the following:

Keyboard versus Mouse

Why do many skilled typists prefer a keyboard to a mouse? Speed is obviously one reason. An experienced typist, through kinesthetic memory, has memorized the location of keyboard keys. The keying process becomes exceptionally fast and well learned. The mouse is slower, and it has a tendency to move about the desk. Its location cannot be memorized. The keyboard keys always remain in the same spot.

Consider the following: When using the mouse, the time to move one's hand from the keyboard, grasp the mouse, and point at a screen object ranges from 1.5 to 2 seconds. A very skilled typist can type 13 to 15 characters in that amount of time; an average typist can type 4 to 6 characters. No wonder the keyboard is often preferred.

Control Research

Which devices work better for which tasks and under what conditions has been addressed by many investigators. A survey of the research literature comparing and

evaluating different devices yields the following summarization concerning tasks involving pointing and dragging:

- The fastest tools for pointing at stationary targets on screens are the devices that permit direct pointing: the touch screen and light pen. This is most likely because of their high level of eye-hand coordination and because they use an action familiar to people.
- In terms of positioning speed and accuracy for stationary targets, the indirect pointing devices — the mouse, trackball, and graphic tablet — do not differ greatly from one another. The joystick is the slowest, although it is as accurate as the others. Of most importance in selecting one of these devices will be its fit to the user's task and working environment.
- A separate confirmation action that must follow pointer positioning increases pointing accuracy but reduces speed. The mouse offers a very effective design configuration for tasks requiring this confirmation.
- For tracking small, slowly moving targets, the mouse, trackball, and graphic tablet are preferred to the touch screen and light pen because the latter may obscure the user's view of the target.

Another common manipulation task is dragging an object across the screen. Pointing and dragging using a mouse, graphic tablet, and trackball for this task was studied by MacKenzie et al., (1991). They report the following:

- The graphic tablet yielded best performance during pointing.
- The mouse yielded best performance during dragging.
- The trackball was a poor performer for both pointing and dragging, and it had a very high error rate in dragging.

Guidelines for Selecting the Proper Input Device

- Consider the characteristics of the task.
  - — Provide keyboards for tasks involving
    - Heavy text entry and manipulation.
    - Movement through structured arrays consisting of a few discrete objects.
  - — Provide an alternative pointing device for graphical or drawing tasks. The following are some suggested best uses:
    - Mouse — pointing, selecting, drawing, and dragging.
    - Joystick — selecting and tracking.
    - Trackball — pointing, selecting, and tracking.
    - Touch screen — pointing and selecting.
    - Graphic tablet — pointing, selecting, drawing, and dragging.
  - — Provide touch screens under the following conditions:
    - The opportunity for training is minimal.
    - Targets are large, discrete, and spread out.
    - Frequency of use is low.
    - Desk space is at a premium.
    - Little or no text input requirement exists.

- Consider user characteristics and preferences.
  — Provide keyboards for touch typists.
- Consider the characteristics of the environment.
- Consider the characteristics of the hardware.
- Consider the characteristics of the device in relation to the application.
- Provide flexibility.
- Minimize eye and hand movements between devices.

Selection of the proper device for an application or system, then, depends on a host of factors.

Task characteristics. Is the device suited for the task? Standard typewriter keyboards are always necessary for tasks requiring text entry and manipulation; keyboards (cursor control keys) are usually faster when moving through structured arrays consisting of a few discrete objects. For graphical and drawing tasks, alternative pointing devices are easier and faster. Use a mouse, joystick, trackball, or graphic tablet for pointing, selecting, drawing, dragging, or tracking. The devices best suited for each kind of task are summarized in the preceding list.

Provide touch screens when the opportunity for training is minimal; targets are large, discrete, and spread out; frequency of use is low; desk space is at a premium; and little or no text input requirement exists. The pointing device should also reflect the degree of accuracy needed to perform the task. Touch screens are generally inaccurate, unless the target is large enough, comprising space availability.

Consider how much time the user will spend on the system. Using a touch screen for a long period of time can be fatiguing. Devices where the forearm can be rested on a table are best for long periods of use.

User characteristics and preferences. Will the user be able to easily and comfortably operate the control? Are the fine motor movements required by some devices capable of being performed? Some children and people with disabilities such as hand tremor may find it difficult to use devices requiring a high degree of accuracy. A study found elderly adults enjoy significant improvements in their interaction accuracy with a touch screen relative to a mouse, whereas younger adults do not do any better with a touch screen (Iwase and Murata, 2002).

Is the user familiar with the standard keyboard? What are the user's preferences? While preferences do not always correspond to performance, it is important that the user be comfortable with the selected device. Also consider device-learning requirements. Direct pointing devices (light pen, stylus, and touch screen) are intuitive and easy to learn. They are best in situations where people cannot be expected to spend time learning, and for infrequent system users.

Environmental characteristics. Will the device fit easily into the work environment? If desk space is necessary, does it exist and is it large enough? Some devices require very little room (trackball, joystick, and graphic tablet). A mouse requires significantly more space. Public access systems will require a sturdy device such as the touch screen. A pen or stylus can easily be broken or stolen. Touch screens work well when the usage environment is dirty.

Hardware characteristics. Is the device itself of a quality that permits easy performance of all the necessary tasks? Joysticks, for example, are quite variable in their movement capabilities.

The device in relation to the application. Is the device satisfactory for the application?

Flexibility. Often task and user needs will vary within an application. Providing more than one kind of device will give the user choices in how to most efficiently accomplish whatever tasks must be performed. A keyboard paired with another kind of pointing device is almost always necessary.

Minimizing eye and hand movements. When multiple devices are used, eye and hand movements between them must be minimized. Structure the task, if possible, to permit the user to stay in one working area. If shifts must be made, they should be as infrequent as possible. It is estimated that, for a good typist, it costs 3 to 8 keystrokes for each jump between the keyboard and a mouse. The general rule is that more than 80 percent of the tasks should be doable using only one device.

Pointer Guidelines

- The pointer
  - — Should be visible at all times.
  - — Should contrast well with its background.
  - — Should maintain its size across all screen locations and during movement.
  - — The hotspot should be easy to locate and see.
  - — Location should not warp (change position).
- The user should always position the pointer.
- The shape of a pointer
  - — Should clearly indicate its purpose and meaning.
  - — Should be constructed of already defined shapes.
  - — Should not be used for any purpose other than its already defined meaning.
  - — Do not create new shapes for already defined standard functions.
- Use only as many shapes as necessary to inform the user about current location and status. Too many shapes can confuse a person.
- Be conservative in making changes as the pointer moves across the screen.
  - — Provide a short "time-out" before making noncritical changes on the screen.
- Animation should not
  - — Distract.
  - — Restrict one's ability to interact.

The focus of the user's attention in most device operations is usually the pointer. Therefore, the pointer image should be used to provide feedback concerning the function being performed, the mode of operation, and the state of the system. For example, the pointer shape image can be changed when it is positioned over a selectable object, signaling to the user that a button action may be performed. When an action is being

performed, the pointer can assume the shape of a progress indicator such as an hourglass, providing an indication of processing status.

A pointer should contrast well with its background and be visible at all times. The user should always be in control of its location on the screen. The shape of a pointer should clearly indicate its purpose and meaning. Always use predefined shapes provided by graphical systems. Microsoft Windows, for example, provides about two dozen standard shapes. To aid learning and avoid user confusion, never create new shapes for already defined standard functions or use a shape for any purpose other than its previously defined meaning. Also, use only as many shapes as absolutely necessary to keep the user informed about current position and status. Too many shapes can also confuse a person.

Be conservative in making changes as the pointer moves across the screen. Excessive changes can be distracting to a person. To avoid frequent changes while crossing the screen, establish a short time-out before making noncritical pointer changes. Any pointer animation should not distract the viewer or restrict one's ability to interact with the system.

Output Devices

The computer communicates to the user through output devices. The most common is the display screen.

Screens

Screens are very useful for presenting a wide range of visual elements and complex data. They have been the workhorse of computer systems for decades. For many years the visual display terminal, or VDT (as it was called before the term monitor became popular when the keyboard and monitor were detached), has been constructed using a raster-scan cathode ray tube (CRT). In its earlier years CRT's were very prone to flickering, but technological improvements have resulted in flicker-free CRTs with excellent resolution. Their one drawback is their size.

Recent years have seen tremendous improvement in an alternative display, the liquid crystal display (LCD). LCDs are much smaller, thinner, and lighter than CRTs while possessing the same viewing area. They consume less power than a CRT and are now much more affordable. Stone et al. (2005) suggests the following should be considered in choosing a screen:

Image. How detailed does the image need to be? For highly graphic applications involving images and photographs, a high-resolution screen will be desirable. For text work and larger letter sizes, a lower resolution may be satisfactory.

Colors. How many colors are needed? The number of colors can range from monochrome to millions. The application will determine this requirement.

Size. Like televisions, screen sizes are measured across the diagonal. Common desktop sizes are 17, 19, and 21 inches. The larger the screen, the more advantages exist. More information can be displayed, as can larger text and images. Size will

depend upon the needs of the application and the needs of the user. A visually impaired person, for example, may well want larger text and images. Much smaller screens are required for hand-held devices.

Portability. Does the screen need to be portable? CRTs are heavy and not portable. LCDs are lightweight and very portable.

Usage space. CRTs have a much larger desktop footprint. LCDs take up much less space. Crowded desks and environments will find benefits in an LCD.

Speakers

Computer sounds have advanced from a simple beep to the reproduction of speech, music, and sound effects. The quality of the speaker should reflect the quality of the sound being presented.

Step 6 Exercise

An exercise for Step 6 can be found on this book's companion Web site, www.wiley.com/college/galitz.

STEP
7

Choose the Proper Screen-Based Controls

Screen controls, sometimes called *widgets*, are the elements of a screen that constitute its body. By definition, they are graphic objects that represent the properties or operations of other objects. A control may

- Permit the entry or selection of a particular value.
- Permit the changing or editing of a particular value.
- Display only a particular piece of text, value, or graphic.
- Cause a command to be performed.
- Possess a contextual pop-up window.

In the last decade, some platforms have expanded the definition of a control to include all specifiable aspects of a screen, including screen text, headings, and group boxes. For the purposes of this discussion, this broader definition of a control will be assumed. This step will encompass

- Identifying the characteristics and capabilities of the various screen controls, including
  - Buttons.
  - Text entry/read-only controls.
  - Selection controls.
  - Combination entry/selection controls.
  - Specialized operable controls.
  - Custom controls.

  - Presentation controls.
  - Web controls.
- Selecting the proper controls for the user and tasks.

The screen designer is presented with an array of screen controls to choose from. Selecting the right one for the user and the task is often difficult. But, as with input devices, making the right choice is critical to system success. A proper fit between user and control will lead to fast, accurate performance. A poor fit will result in lower productivity, more errors, and dissatisfaction.

We'll start by describing the types of controls and identifying their advantages, disadvantages, and proper usage. Relevant control design guidelines will also be presented. Not all toolkits or platforms will necessarily possess all the kinds of controls to be described. After describing the controls, we'll look at several research studies addressing the way to choose the best control or controls for particular situations. By the time these studies are reviewed, their findings will have been incorporated into the control usage and design guidelines already presented. This organization has been chosen because it is more meaningful to first clearly describe each control before discussing it in a research context. We'll finish by providing some general guidance in choosing the proper kind of control to enable tasks to be performed quickly and efficiently by the user.

In describing the controls, we'll break them down into categories that reflect the way they are used. We'll begin with operable controls, those that are manipulable, changeable, or settable. We'll then review presentation controls, those used to inscribe permanent information on a screen or used to give the screen structure. Before starting this review, three extremely important principles regarding controls should be noted:

- A control must
  - Look the way it works.
  - Work the way it looks.
- A control must be used exactly as its design intended.
- A control must be presented in a standard manner.

The look of a control should make it obvious that it is a control. Its design characteristics should signal "enterability" or "clickability." Microsoft Windows, for example, presents the following simple rules:

- Raised elements can be pressed.
- Recessed elements cannot be pressed.
- Elements on a flat white background can be opened, edited, or moved.

A control must also be presented in a standard and consistent manner, and used exactly as its design intended. The nonstandard design use of controls destroys consistency and aggravates and frustrates users, who have developed expectations based upon their past experiences. Using standard controls allows people to focus on their tasks or the content of the screens with which they are interacting, instead of having to figure out what to do.

Web page design has unleashed and exposed thousands of instances where these basic principles (and others to be described) have been violated. Page designers, all too often it seems, have been placing greater value on personal creativity than on interface usability. Some examples will be presented throughout the following pages.

Operable Controls

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

Buttons

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

A button comes in three styles. The first resembles the control commonly found on electrical or mechanical devices and is sometimes called a pushbutton. These are most often rectangular, with text that indicates the action to be taken when they are selected or pressed. These buttons are usually placed within a window, and activating them causes the action or command described on them to be performed immediately. This kind of button may take a variety of forms, some of which are illustrated in Figure 7.1. They are often referred to as *command buttons.*

The second style is square or rectangular in shape with an icon or graphic inside. It may have an associated label. This kind of button is illustrated in Figure 7.2. The label may be permanently affixed to the screen within the button, adjacent to it, or only appear when the pointer is moved to the button (called ToolTip, to be discussed). These buttons may appear singly or be placed in groupings commonly called button bars or toolbars. We'll refer to them as *toolbars* in this text. They are most frequently used to quickly access commands, many of which are normally accessed through the menu bar, or to initiate other actions or functions. These button groupings are usually placed at the screen's top or side. They are usually relocatable and removable by the user.

The third style is square or rectangular in shape with a symbol inscribed inside, as illustrated in Figure 7.3. The symbol, when learned, identifies the button and the action to be performed when the button is selected. 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 obtain a system menu or resize a window. They are discussed in more detail in Step 5 and will not be addressed in this step. This step will focus on command and toolbar buttons.

Figure 7.1: Command buttons.

Figure 7.2: Toolbar buttons without labels.

Figure 7.3: A symbol button.

Command button advantages. An advantage of a command button is that 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.

Command button disadvantages. Among the disadvantages of command buttons is their larger size, which consumes considerable screen space and limits the number that can be displayed.

Toolbar advantages. Advantages of toolbar buttons include their continuous visibility and ease and speed of use. They also, individually, consume a relatively small amount of space.

Toolbar disadvantages. Disadvantages include their location being away from the main work area and their small size, which slows down selection. Another disadvantage is that 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.

Proper usage. Buttons are best for frequently used actions in a window. They can be used to cause actions to occur immediately, such as saving a document, quitting a system, or deleting text. They can be used to display a menu of options, such as colors or fonts. Microsoft Windows calls a button that leads to a menu a *menu* button. Buttons can also be used to display other secondary windows or dialog boxes, and to expand the dialog or invoke dialog features. Windows calls a button that expands the dialog an *unfolding* button. Buttons may also be used to reflect a mode or property value setting similarly to the use of radio buttons or check boxes. In some kinds of windows, command buttons may be the only command method available to the user.

In Web application or page design, buttons should be only used to cause an action to occur. They should never be used to retrieve or show information. A button is designed to imply it can be pressed. When it is pressed it does something. Always use links to show information. Maintaining this distinction aids understanding and learning.

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.

For fast access to commands contained in a menu bar, especially those frequently used or critical, also provide access by command buttons. Buttons must also be provided for situations where a command is not available through the menu bar. For windows without menu bars, buttons must be provided to provide access to all window 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.

The shape of a button can vary. Generally, rectangular-shaped buttons are preferred because they provide the best fit for horizontally arrayed textual captions. Square-cornered rectangles are found in some platforms including Microsoft Windows, while rounded-cornered rectangles are found in others. The button style chosen must reflect the three cornerstone principles presented at the beginning of this step, including giving it a raised appearance to make it obvious that it is a command button. To do this, drop shadows are used in some platforms, beveled edges in others. "Invisible" buttons must never exist. Web command button styles are noted for their variety in shape and size. The button style chosen is mostly a matter of preference. Web-specific button styles should be consistently designed and maintained throughout the Web site.

Labels

- Use standard button labels when available.
- Provide meaningful descriptions of the actions that will be performed.
- Use single-word labels whenever possible.
  — Use two to three words for clarity, if necessary.
- Use mixed-case letters with the first letter of each significant label word capitalized.
- Display labels
  — In the regular system font.
  — In the same size font.
- Do not number labels.
- Center the label within the button borders, leaving at least two pixels between the text and the button border.
- Provide consistency in button labeling across all screens.

Labels. Button labels should be clearly spelled out, with meaningful descriptions of the actions they will cause to be performed. Choices should be composed of mixed-case single words. Multiple words are preferred, however, to single words lacking clarity in their intent. If multiple-word labels are used, capitalize the first letter of each word (headline style). Use the same size and style of font in all buttons. The regular system font is preferred. Never change font style or size within buttons; these kinds of changes can be very distracting to the viewer. Center each label within the button borders, leaving at least two pixels between the text and the border.

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.

Always follow all platform presentation and usage guidelines for standard button functions.

Size

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

Provide as large a button as possible, consistent with Fitts' Law (see Step 1). Buttons must, at minimum, be wide enough to accommodate the longest label. Leave at least two pixels between labels and button borders. A command button's minimum height should be 25 pixels. Create, however, standard, equal-sized buttons encompassing the majority of system functions. When a button's label will not fit within the standard size, expand the button's size to achieve a proper label fit. Never reduce the font size of some labels to create equal-sized buttons. In this case, buttons of different widths are preferable. Also, do not create an unnecessarily wide button for aesthetic balance purposes, as illustrated by the Color Palette button in Figure 7.4. The perceptual model we possess in our memory for a button will be lost.

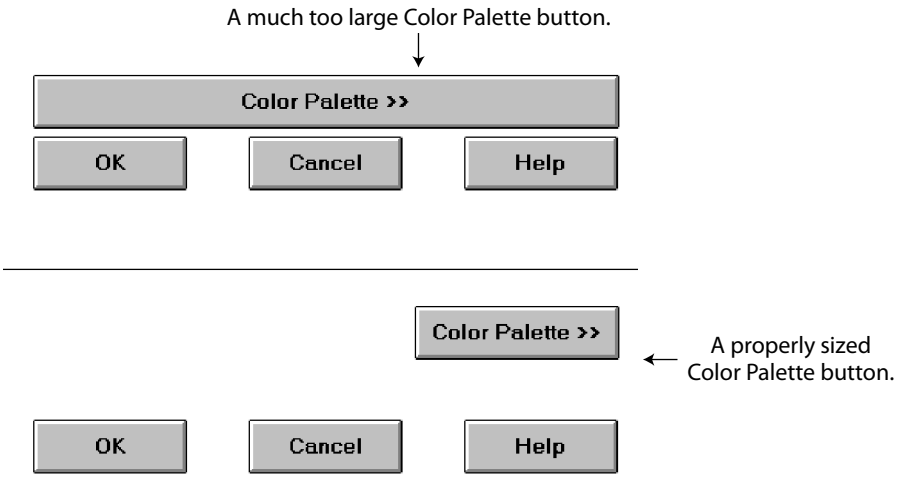


Figure 7.4: Improper and proper button sizes.

Number

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

The maximum number of buttons on a window must reflect a balance between effectiveness, real estate efficiency, and operational simplicity. Having no more than six buttons per window or dialog box seems to appropriately balance these issues. If an extra button or two is necessary and space is available, they may be included.

Location and Layout

- Maintain consistency in button location between windows.
- Never simply "fit" buttons in available space.
- If buttons are for exiting the dialog
  — Position them centered and aligned horizontally at the bottom.
- If buttons are used for invoking a dialog feature or expanding the dialog
  — Position them centered and aligned vertically on the right side.
- If a button has a contingent relationship to another control
  — Position it adjacent to the related control.
- If a button has a contingent relationship to a group of controls
  — Position it at the bottom or to right of related controls.
- If there are space constraints, exiting and expanding/invoking feature buttons must be placed together
  — If at the bottom, place exiting buttons to the right, separating the groupings by one button's width.
  — If along the right side, place exiting buttons at the bottom, separating the groupings by one button's height.

- For exiting and expanding/invoking feature buttons, do not
  — Align with the other screen controls.
  — Present displayed within a line border.
- Provide equal and adequate spacing between adjacent buttons.
- Provide adequate spacing between buttons and the screen body controls.
- For Web pages spanning more than one screen
  — Repeat the buttons at the top and bottom of the page.

Command buttons should be positioned in consistent positions within a window. This enables a person to memorize button locations and predict where they will appear when a window is presented. For an experienced user this permits faster pointing and activation because a button may be identified simply by its location without its label having to be read, and a mouse movement to that location may be commenced before a window is even displayed. Consistent locations also aid in quickly discriminating the different kinds of buttons described below. A common failing of many windows is that buttons are positioned within windows *after* locations for the other window controls are established. When this occurs, buttons are positioned where there is space available. The result is usually a hodgepodge of locations. Never simply "fit" buttons in available space. Allocate a space for buttons before the other control locations are established.

Button location within a window is dependent upon the type of button. Buttons *exiting* a dialog, and usually closing the window, should be positioned horizontally and centered across the lower part of the window. This positioning places the buttons at the end of the dialog. A study of Web pages (Spool et al., 1997) found that people preferred to scroll to the bottom of a page to press the final buttons. If a button *invokes* a dialog feature or *expands* the dialog, position it centered and aligned vertically along the right side of the window. Maintaining these consistent locations will enable a person to quickly identify what general kind of button it is, and what kind of action will occur if the button is activated. The location of the exiting buttons across the bottom will also allow more efficient use of window real estate when invoking/expanding buttons are not included within a window. Exiting and expanding/invoking feature button locations are illustrated in Figure 7.5. If exiting and expanding/invoking feature buttons must be positioned together at the screen bottom because of screen space constraints, place the exiting buttons to the right, separating the groupings by one standard button's width. If they are located together along the right side, place exiting buttons at the bottom, separating the groupings by one button's height.

If a button has a *contingent* relationship to another control, position it adjacent to the related control in the order in which the controls are usually operated, as illustrated in Figure 7.6. If a button possesses a contingent relationship to a group of controls, position it at the bottom or to the right of the grouping, again in logical flow order, as illustrated in Figure 7.7.

For Web pages containing buttons and longer than one screen, repeat the buttons at the top and bottom of the page. This will provide easier access to the buttons from varying locations within the page, minimize the chance of people missing the buttons, and comply with the expectancies of people who assume they will be found both at the top and bottom.

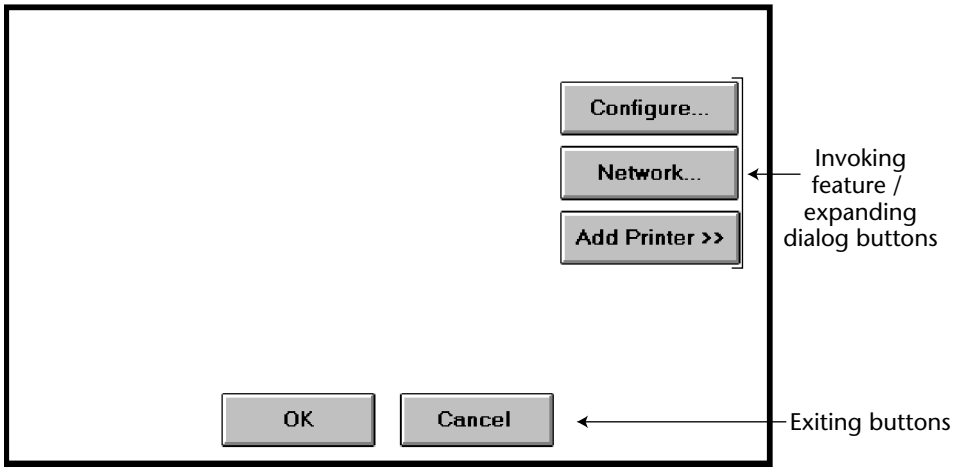


Figure 7.5: Exiting and invoking feature/expanding dialog buttons.

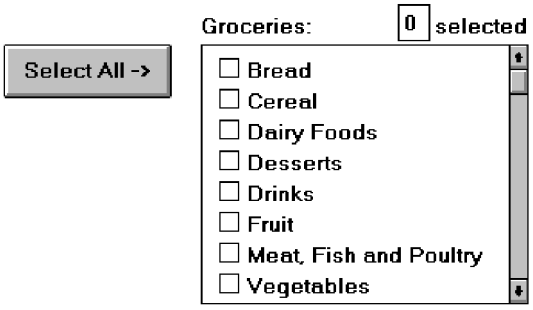


Figure 7.6: Button with contingent relationship to a control.

For exiting and expanding/invoking feature buttons, do not provide alignment with the other screen controls. Maintain alignment and spacing only within the buttons themselves. Trying to align the buttons to other screen controls will most often create variable spacing between the buttons themselves, which is visually distracting. Also, do not display buttons within a line border; instead present them on the background of the window itself. The unique physical look of the buttons is strong enough for them to create their own visual grouping. Reserve line borders for individual controls or groups of controls that are in greater need of closure. Too many borders can also create visual clutter.

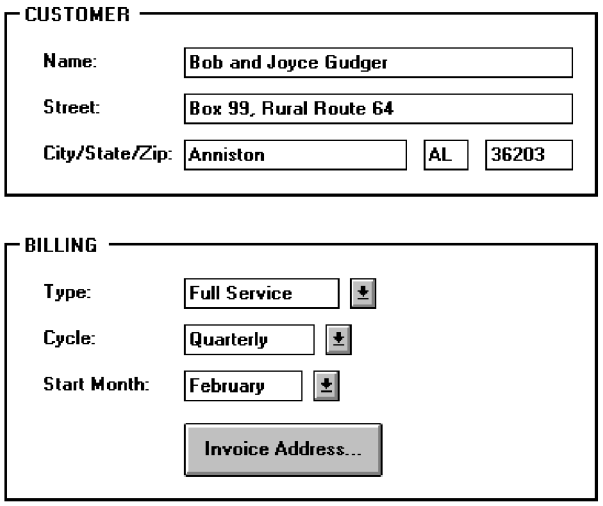


Figure 7.7: Button with contingent relationship to a grouping.

Provide equal and consistent spacing between adjacent buttons, and groups of buttons. Also, maintain adequate separation between screen buttons and other screen 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.

Other platforms may suggest a different ordering. If differences exist, and people may be using more than one platform, some organizational compromises may be necessary.

Buttons should be ordered logically, such as by frequency of use, sequence of use, or importance. For buttons arrayed left to right, start the ordering from left to right. For buttons arrayed top to bottom, start the ordering from top to bottom.

Keep related buttons grouped together. Separate potentially destructive buttons from frequently chosen selections to avoid inadvertent activation and potentially catastrophic results. Always locate the same buttons that appear on different windows in consistent positions. For mutually exclusive actions, avoid using one button that toggles changing text. This can be confusing. Use two buttons with labels that clearly describe the two actions.

Intent Indicators

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

Figure 7.8

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

Figure 7.9

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

Figure 7.10

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

Figure 7.11

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

Figure 7.12

Button intent indicators will follow, where applicable, the same conventions used for menu items. When a button causes a command to be performed immediately, no special intent indicator is necessary on the button. When a button leads to a cascading dialog box, include an *ellipsis* with the label. When a button leads to a menu of choices, include a *triangle* after the label; point the triangle in the direction the menu will appear. If a button expands the dialog, include a *double arrow* with the label. When a button has a contingent relationship to another control, include a *single arrow* pointing at the control. Intent indicators are very useful because they enable the user to predict the consequences of an intended action.

Expansion Buttons

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

When a button that expands a dialog is activated, and the dialog is expanded, display the button dimmed or grayed out. If the dialog can again be contracted, provide a contraction button beneath the expansion button or to the right of it. Gray this button out when the dialog is contracted; display it at normal intensity when the dialog is expanded.

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.

When a window with buttons is first displayed, provide a default action whenever practical. The default action should be the most likely action within the window. It may be a confirmation, an application of the activity being performed, or a positive response such as OK. If the default is irreversible or destructive (such as Delete), the default should be Cancel, requiring a person to change the selection in order to perform the destructive action. If none of the buttons is destructive in nature, the default button should be the one most frequently selected.

The default can be changed as the user interacts with a window. When the user navigates to a button, it can temporarily become the default. Return the button to its original state when the focus leaves a button. Permit use of the Enter key to activate a default button. If another control requires use of the Enter key, temporarily disable the default while the focus is on the other control. Permit double-clicking on a single selection control in a window to also carry out the default command.

Unavailable Choices

- Temporarily unavailable choices should be dimmed or grayed out.

A button should visually indicate whether it is available for activation. Dim or gray-out 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.

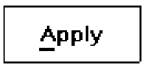


Figure 7.13

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

Enabling the user to select button actions through the typewriter keyboard provides flexibility and efficiency in the dialog. To do this, provide keyboard equivalent, single-character mnemonic codes that, when typed, will cause the action to be performed. The suggested method is to indicate the accelerator by underlining the proper character in the button label.

Keyboard accelerators, a keyboard key or combination of keys, may also be assigned to buttons to facilitate keyboard activation.

Keyboard equivalents and accelerators, including Microsoft Windows standard ones, are discussed in more detail in Step 4.

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.

If scrolling the contents of a window, never scroll the buttons. They should be available at all times. Web page screens, whose content cannot be scrolled independently of buttons, are exceptions to this rule at the moment. Use buttons to move between multipage forms, not scroll bars. Paging is, conceptually, easier for people to use and understand, and is discussed in detail in Step 3. Label the 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.

Highlight the button in some visually distinctive manner when the pointer is resting on it and it is available for selection. This will provide the user with feedback indicating that the selection process may be performed. Some platforms display a brighter button.

Highlight the button in another visually distinctive manner when it has been activated or pressed, to indicate that the action is successful. One platform subdues or grays out the button. Another has raised beveled buttons that appear to sink into the screen when selected. Another alternative is to move the button slightly as if it has been depressed. If a button can be pressed continuously, permit the mouse button to be held down and the action repeated.

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.

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.

Provide toolbars to allow fast and easy access to a system's most frequently used commands, functions, or options. Also provide toolbars for easily invoking subapplications within an application. Toolbars are considered "fast paths" or expert aids. All toolbar functions must also be obtainable by normal textual menu means, including through use of the menu bar. One exception: If menu text cannot clearly explain an item and a graphical toolbar representation can, the toolbar item may replace the menu item.

Structure

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

Create meaningful and unique images and icons utilizing the icon design guidelines in Step 11. Center the image within the button and provide an associated textual label. A label is always necessary to ensure button comprehensibility. One study has found that placing graphics and words on buttons, makes them more usable than including graphics only (Vaughan, 1998). Create the label following the guidelines for command buttons. The label may be located within the button, positioned beneath it, or presented on demand through a ToolTip control. Labels beneath toolbar button images will provide a larger pointing target. If the label is located within the button and the system will be translated into one or more other languages, allow extra space for the label. See "International Considerations" in Step 10 for further important considerations. A ToolTip control is discussed later in this chapter. Give the button a raised appearance to convey that it is a screen navigation element to be pressed. Ensure that toolbar images are discernible from all Web page graphical images.

For text-only toolbar buttons, create a meaningful label, adhering to the label guidelines for command buttons. Provide consistent icons throughout all 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.

A toolbar button should be large but not too large because of the number that may need to be displayed. Microsoft provides the preceding guidelines. Other sizing guidelines and much more detailed image guidelines are presented in Step 11.

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.

Toolbar buttons should be ordered logically, such as by frequency of use, sequence of use, or importance. If the buttons reflect a quality on a continuum such as colors or shades, follow standard and expected ordering schemes. For buttons arrayed left to right, start the ordering from left to right. For buttons arrayed top to bottom, start the ordering from top to bottom.

Keep related buttons grouped together. Separate potentially destructive buttons from frequently chosen selections to avoid inadvertent activation and potentially catastrophic results. Permit the user to reconfigure the button organizational structure to better meet his or her unique needs.

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.
  — Also provide access through standard menus.

Locate the main features and functions tool bar horizontally across the top of the window just below the menu bar. Locate subtask and subfeature tool bars along sides of window. Permit the location of the toolbar to be changed by the user. Because a toolbar can create visual noise, permit its display to be turned on or off. Always also provide access to the toolbar actions through standard menus.

Active Items

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

As the user moves around through an application, items at various points that are not applicable do not have to be displayed. Temporarily unavailable items may be grayed out.

Customization

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

Permit the toolbars to be turned off by the user, should their use not be necessary or should more screen space be desired. Also, allow users to customize the toolbar, determining what they would like to add or remove. 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.

Provide keyboard equivalents and accelerators to facilitate keyboard selection. Maintain the same mnemonic on all identical buttons on all screens. One caution, if a

particular mnemonic is being used somewhere else in the window: It may not be available for use on the toolbar.

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.

Highlight the button in some visually distinctive manner when the pointer is resting on it and the button is available for selection. This will provide the user with feedback indicating that the selection process may be performed. Highlight the button in another visually distinctive manner when it has been activated or pressed, to indicate that the action is successful.

Text Entry/Read-Only Controls

A Text Entry control contains text, free-form in nature that is exclusively entered or modified using the keyboard. A Read-Only control will contain text or values being presented for reading or display purposes only. Through tradition, these controls are usually referred to as *fields*. In graphical system terminology they are called *text boxes*. A text box into which information can be keyed is called an *unprotected* field. A text box used for display purposes only is referred to as a *protected* field. Historically, they have been called *inquiry*, *display* or *read-only* fields.

An unprotected text entry field can also be designated as required or optional. A *required* unprotected field is one in which the necessary information must be always be keyed. That the entry is complete and valid may also be a requirement. An *optional* field is one in which information need not always be keyed. Whether information is keyed into it depends on the circumstances of the moment. General design guidelines for text boxes are the following.

Text Boxes

- Description:
  - — A control, usually rectangular in shape, in which
    - Text may be entered or edited.
    - Text may be displayed for read-only purposes.
  - — Usually possesses a caption or label describing the kind of information contained within it.
  - — An outline field border
    - Is included for enterable/editable text boxes.
    - Is not included for read-only text boxes.

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

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

- Purpose:
  — To permit the display, entering, or editing of textual information.
  — To display read-only information.
- Advantages:
  — Very flexible.
  — Familiar.
  — Consumes little screen space.
- Disadvantages:
  — Requires use of typewriter keyboard.
  — Requires user to remember what must be keyed.
- Proper usage:
  — Most useful for data that is
    - Unlimited in scope.
    - Difficult to categorize.
    - Of a variety of different lengths.

  — When using a selection list is not possible.

While display-only text boxes are not operable in the true sense of the word, the information contained within them is capable of being modified by other controls. Hence, they will be reviewed as an operable control because their characteristics, and the characteristics of an entry field, are very similar.

Description. Text boxes almost always possess a separate caption describing the kind of information to be keyed. An enterable text box is visually presented on the screen, its shape being defined by an outline border appearing recessed, and a lighter background so that it contrasts with the screen background. The information in a read-only text field is most effectively displayed on the screen background, not in a box. Therefore, a box does not surround the information contained in read-only text boxes. Text boxes may be single-line or multiple-line. Text boxes will usually contain captions or labels. Because the most important component of a text box is the data, captions should be lighter or less visible than the data. When first displayed, a text box may be blank or contain an initial value.

Advantages/Disadvantages. Text boxes are very flexible, accepting almost any keyed entry. They are familiar entities and do not consume much screen space. Disadvantages include the requirement that the user must find and/or remember what information must be keyed. One's powers of recall are often tested. Because a keyboard must be used for typing, one's typing skill influences speed and accuracy of use.

Proper Usage. Text boxes are most useful when the data for entry is unlimited in scope, difficult to categorize, and quite variable in length. Text boxes are usually the only alternative when creating a selection list is not possible.

Single-Line and Multiple-Line Text Boxes

- Single-line:
  - — Description:
    - A control consisting of no more than one line of text.
  - — Purpose:
    - To make textual entries when the information can be contained within one line of the screen.
  - — Typical uses:
    - Typing the name of a file to save.
    - Typing the path of a file to copy.
    - Typing variable data on a form.
    - Typing a command.
- Multiple-line:
  - — Description:
    - A control consisting of a multiline rectangular box for multiple lines of text.
  - — Purpose:
    - To type, edit, and read passages of text.
  - — Typical uses:
    - Creating or reading an electronic mail message.
    - Displaying and editing text files.

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. Multiple-line boxes are used when the information exceeds a single line. Text boxes are illustrated in Figure 7.14.

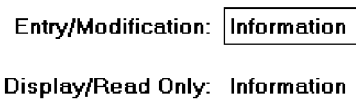


Figure 7.14: Text boxes.

Captions or Labels

- 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.

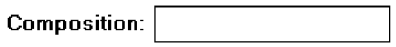


Figure 7.15

- 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.

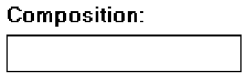


Figure 7.16

  - — Multiple occurrence fields:
    - For entry/modification text boxes, position the caption left-justified one line above the column of entry fields.

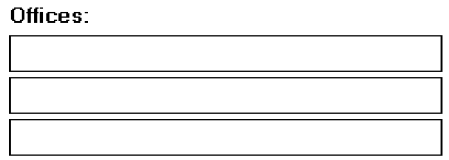


Figure 7.17

- 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.

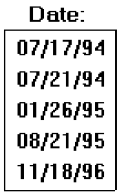


Figure 7.18

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

Figure 7.19

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

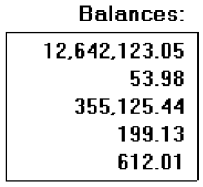


Figure 7.20

Structure and size. Captions or labels are usually added to text boxes using *static text* fields, to be described shortly. Many toolsets do not include captions as part of text box controls. Captions must be understandable and fully spelled out in a language meaningful to the user. In general, abbreviations and contractions should not be used. To achieve the alignment recommendations (to be discussed shortly), however, an occasional abbreviation or contraction may be necessary. If so, choose those that are common in the everyday language of the user or those that are meaningful or easily learned. Use mixed-case text in the headline style, capitalizing only the first letter of each word (except for articles, conjunctions, and prepositions — a, the, and, for, and so on). Acronyms, abbreviations, or proper nouns that are normally capitalized, however, may be capitalized. If the caption is of a sentence-style nature, sentence-style capitalization should be followed. In this case, capitalize only the first letter of the first word of the caption.

In comparison to the text box, the caption should be captions should be lighter or less visible than the data. Visual emphasis should always be given to the information in the text box.

Single fields. For single fields, it is recommended that the caption precede the text box. Place a colon (:) directly following the caption to visually separate the caption from the data; separate the colon from the text box by one space.

Multiple occurrence fields. For multiple-occurrence fields, the caption should be positioned above the columnized text boxes. The exact location of the caption will depend on the kind of screen and the kind of data displayed. For entry screens, the caption should be left-justified above the columnized entry fields.

This will signal the starting point of the text box and ensure that the caption is positioned directly above the keyed data.

For display/read-only or inquiry screens in which text box information already exists, the positioning of the caption depends on the kind of information displayed within the box. The objective is to center the caption over the information. If the box is fixed-length, or the information to be displayed within it usually fills, or almost fills, the box, center the caption above the data. If the information is alphanumeric and can be quite variable in length, left-justify the caption. This will keep the caption directly above the data when it appears in the box. Similarly, for numeric fields, right-justify the caption to keep it above the data that will be right-justified when it appears.

Data Fields

- Structure:
  - — Present entry/modification text boxes surrounded by a line border.
    - Present the box in a recessed manner to indicate that it is an enterable field,
    - Provide a lighter color box background contrasting with the screen background.

Figure 7.21

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

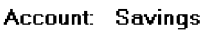


Figure 7.22

  - — Segment long text boxes through
    - Incorporation of slashes (/), dashes (-), spaces, or other common delimiters.
    - If fixed-length data, segment into logical boxes using auto-skip to move between.

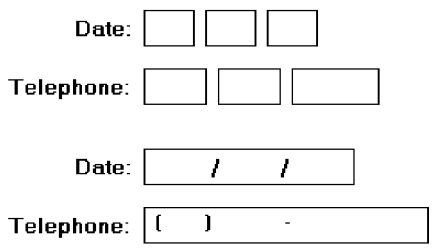


Figure 7.23

- 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.

Structure. A text box should attract attention, but not detract from the legibility of the data contained within it, be capable of allowing an indication of the structure of the data contained within it, and indicate the appropriate number of characters to be keyed into it. An early study found that, in meeting these objectives, a broken underscore and an outlined box were the best delimiters for screen entry fields. The older text-based screens traditionally have used the underscore as the delimiter; graphical screens, the outlined box. Interestingly, both resemble the coding areas most frequently found on paper forms. To visually indicate that it is an enterable field, present the box in a recessed manner, as is done by Microsoft Windows.

Present display/read-only text boxes on the window background. To make text boxes more readable, it is desirable to break them up into logical pieces. Slashes, dashes, and spaces should be inserted into the entry fields as illustrated.

Segment long text boxes by including slashes, dashes, or other common delimiters between their logical groups. For fixed-length data, provide multiple logical boxes using auto-skip to move between them.

Size. The size of a field must give an approximate indication of the data length. Text boxes for fixed-length data must be long enough to contain the entry. Variable-length text boxes should be large enough to contain the majority of the entries. The size of a variable-length text box will be dependent on field alignment, space utilization, and aesthetics. If a text box is not large enough to key or view the entire entry, it must be scrollable. Scrolling, however, should be avoided whenever possible.

Highlighting. Text box data (as opposed to captions) is the most important part of a screen. Call attention to it through highlighting techniques. With monochrome screens, display it bright or in high intensity. With color, use the brightest colors. If a box is the delimiter, choose a background color that complements the screen body background and provides good contrast with the color chosen for the data.

Temporarily unavailable. For fields temporarily unavailable for entry, gray-out the box and its associated label. This temporary graying out implies, however, that the user can perform some action that will again make the field enterable.

Rich-text boxes. Most text boxes typically support only the standard system font. A Microsoft Windows rich-text box is similar to a text box but provides, in addition, font properties, such as typeface, size, color, bold, and italics. It also supports character and paragraph alignment, tabs, indents, and numbering, as well as printing.

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. Some selection controls present all the alternatives together and visible on a screen; others may require an action to retrieve the entire listing and/or scrolling to view all the alternatives. Selection controls include radio buttons, check boxes, list boxes, drop-down/pop-up list boxes, and palettes. Radio buttons and check boxes are similar in structure and use. The distinguishing conceptual difference is that radio buttons permit selecting only choice from the options presented, whereas check boxes permit the selecting of multiple choices.

Radio Buttons

- Description:
  - — A two-part control consisting of the following:
    - Small circles, diamonds, or rectangles.
    - Choice descriptions.
  - — When a choice is selected
    - The option is highlighted.
    - Any existing choice is automatically unhighlighted and deselected.
- Purpose:
  - — To set one item from a small set of mutually exclusive options (2 to 8).
- Advantages:
  - — Easy-to-access choices.
  - — Easy-to-compare choices.
  - — Preferred by users.
- Disadvantages:
  - — Consume screen space.
  - — Limited number of choices.
- Proper usage:
  - — For setting attributes, properties, or values.
  - — For mutually exclusive choices (that is, only one can be selected).
  - — Where adequate screen space is available.

— Most useful for data and choices that are
- Discrete.
- Small and fixed in number.
- Not easily remembered.
- In need of a textual description to meaningfully describe the alternatives.
- Most easily understood when the alternatives can be seen together and compared to one another.
- Never changed in content.

— Do not use
- For commands.
- Singly to indicate the presence or absence of a state.

Structure. Controls of this type take several different physical forms. 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. When an alternative is selected, the center of the circle is partially or fully filled in to provide a visual indication that it is the active choice. Other styles of radio buttons have also been implemented. 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. Another method for presenting exclusive choices is the butted box or button where the alternatives are inscribed in horizontally arrayed adjoining rectangles resembling command buttons. The selected alternative is highlighted in some way. Examples of radio buttons are illustrated in Figures 7.24 and 7.25. Deciding on which style to use seems to be more a matter of preference than performance. No published comparison studies are available for guidance. However, the dominance of Microsoft products suggests that most users are familiar with their presentation style.

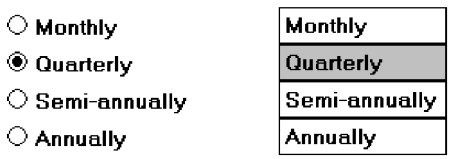


Figure 7.24: Radio buttons.

Figure 7.25: Radio buttons.

Purpose. Radio buttons are used to designate one of a small set of options, no more than about eight. As with a radio, the choices are mutually exclusive, only one frequency or setting is permitted at one time in the presented array.

Advantages/disadvantages. With radio buttons, all alternatives are always visible. Therefore, it is easy to access and compare choices. Two studies (Johnsgard et al., 1995; Tullis and Kodimer, 1992) have found radio buttons a preferred and very effective control for presenting mutually exclusive choices. These studies will be described later in this chapter. On the negative side, radio buttons do consume a certain amount of screen real estate, limiting the number of alternatives that can reasonably be displayed.

Proper usage. Radio buttons are useful for setting attributes, properties, or values where adequate screen space is available. The choices should be discrete, small in number, and in need of a textual description to identify them meaningfully. Radio buttons are helpful in situations where the alternatives cannot always be easily remembered or where displaying the alternatives together facilitates understanding and selecting the proper choice. The radio button choices displayed should be stable, never changing in content. Never use radio buttons for implementing commands, such as causing a dialog box to immediately appear based upon a button setting. Commands to the system should result from direct user command actions, such as pressing a command button. This allows control to remain with the user. Unfortunately, use of a radio button to perform an action is a common Web page design problem. Also, do not use one radio button by itself to indicate the presence or absence of a state. A single check box is recommended for this purpose.

Choice Descriptions

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

Choice descriptions must be clear, meaningful, fully spelled out, and displayed in a mixed-case text. For multiword descriptions, capitalize the first letter of the description and use the sentence style for the remainder of the description. Small button-type indicators should be located to the left of the choice description; rectangular boxes that resemble a command button will find the description within the box. Small buttons associated with text are advantageous when the choice description must be lengthy. Descriptions in boxes impose restrictions on the number of words that can be inscribed within them. When a choice is unavailable for selection in the present condition,

display the choice selection grayed out or dimmed. Where a "None" alternative clarifies the alternatives presented, provide it in the listing.

Size

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

Generally, selection fields of this style should not present more than eight choices. 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 drop-down list box. Johnsgard et al. (1995) have found, however, that even for as many as thirty choices, radio buttons were preferred by users, and performed better, than these other controls.

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.

Provide a default setting for a radio button whenever possible. In some situations, however, a default setting may be difficult to predetermine, or inappropriate to predetermine (sex: male or female?). Microsoft Windows provides for additional settings called the indeterminate or mixed value states. When a default setting cannot be preestablished because of the nature of the information, leave all the buttons blank or not filled in. If a multiple selection on an object is performed and the values in the selection are mixed or differ, display the applicable radio buttons in another distinctive manner, such as a gray shadow.

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.

Figure 7.27

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

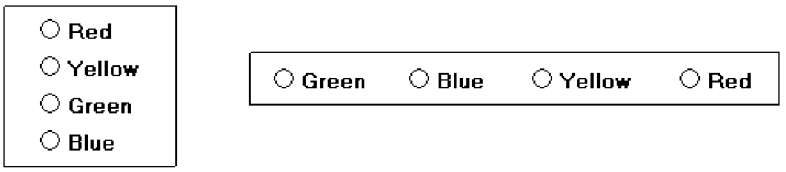


Figure 7.28

The preferred orientation of radio buttons is columnar. This aids visual scanning and choice comparison. Controls with small button indicators usually fit best in this manner because choice descriptions do not have to be restricted in size. Left-align the buttons and choice descriptions. Provide adequate separation-about three spaces-between choices if they must be presented horizontally. Enclose the buttons in a border. Rectangular boxes should be of equal height and/or width and be butted up against one another. This will distinguish them from nonexclusive choice fields (check boxes) that will be separated from one another. Figure 7.29 illustrates the best ways to, and not to, present radio buttons.

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.

Selection choices should be organized logically. If the alternatives have an expected order, follow it. Other ordering schemes such as frequency of use, sequence of use, or importance may also be considered. Always begin ordering at the top or left. If, under certain conditions, a choice is not available, display the nonselectable choice subdued or less brightly than the available choices.

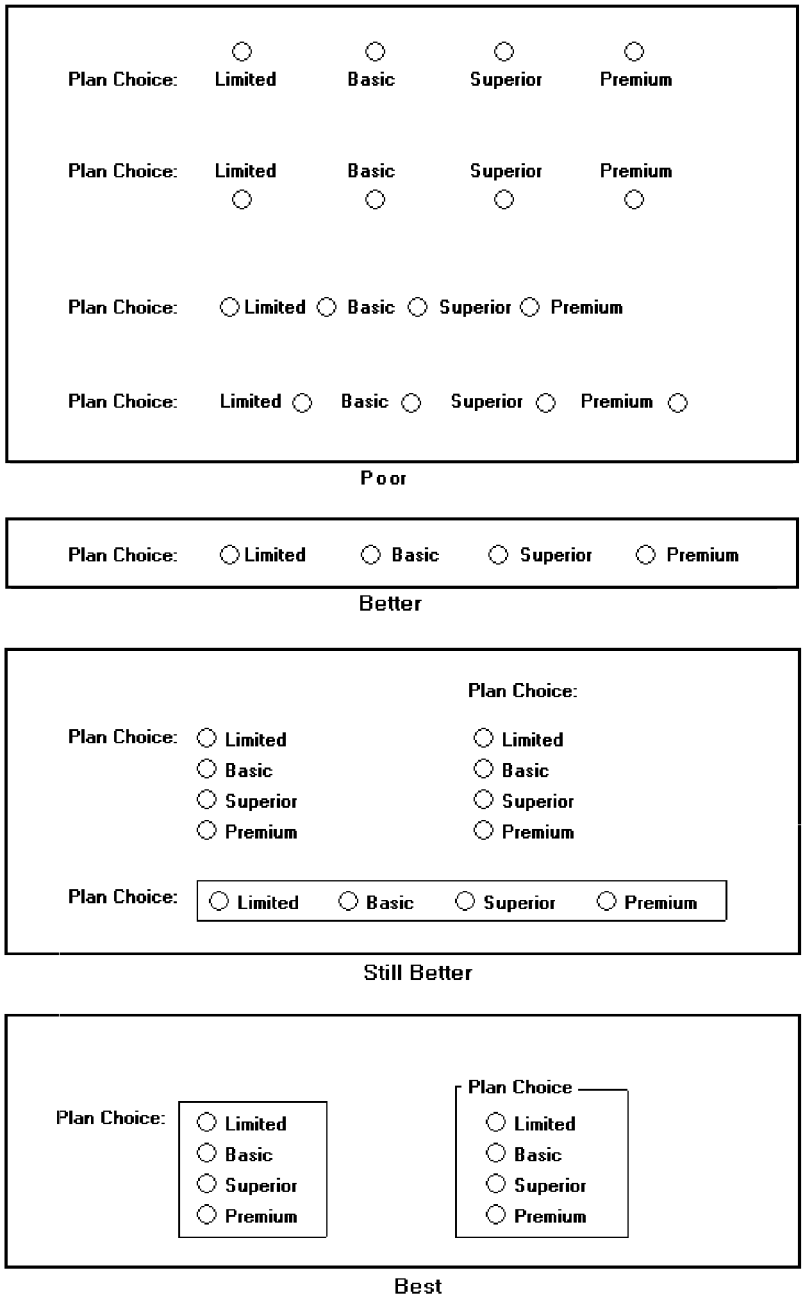


Figure 7.29: Ways to, and not to, present radio buttons.

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

Position any control related to a check box immediately to the right of the choice description. If a the check box label also acts as the label for the control that follows it, present it in mixed case, sentence style text, and end the label with an arrow (>) to relate the choice description to the control.

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.

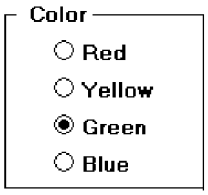


Figure 7.31

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

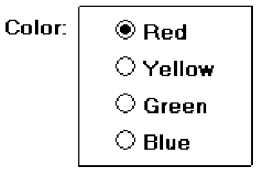


Figure 7.32

— Without an enclosing control border, position the caption

- Left-justified above the choice descriptions, separated by one space line.

Figure 7.33

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

Figure 7.34

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

Figure 7.35

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

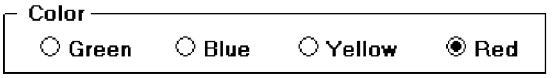


Figure 7.36

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

Structure. Using a static text or group box control field, display the caption fully spelled out, using mixed-case letters in the headline style. Some occasional common abbreviations may be used, however, to achieve the alignment goals to be specified.

Columnar orientation. The preferred location of a radio button control caption within a screen can vary. Ideally, the caption is placed upper-left-justified within a line border or group box that surrounds columnar-oriented radio buttons, as shown in the example in the preceding guideline summary. If other controls on a screen possess captions positioned to the left, and the radio button control is aligned with these controls, position the caption to the left of the control. This will help achieve screen efficiency, minimize viewer eye movements, and provide caption and choice distinctiveness. Without an enclosing control border, position the caption left-justified above the choice descriptions, or to the left of the topmost choice description.

Horizontal orientation. In a horizontal orientation, position the caption to the left of the choice descriptions, or left-justified within an enclosing control border. If the screen contains only one radio button control, the screen title may serve as the control caption. 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.

Pointing. The selection target area should be as large as possible in order to make it easy to move to. If a small button is the selection indication method used, the target area should include the button and the choice description text. If the rectangular box selection method is used, the entire box should be the target. Highlight the selection choice in some visually distinctive way when the pointer is resting on it and the choice is available for selection. If a small button is the selection indication method used, a distinctive reverse video, reverse color, or dotted or dashed box selection cursor or bar may be used to surround the selected choice description. This cursor should be as long as the longest description plus one space at each end. The cursor should not cover the small button.

Activation. When a choice is selected, distinguish it visually from the unselected choices. A radio button should be filled in with a solid dark dot or other similar marking (for example, making the button look depressed or higher than the others through the use of a drop shadow). A rectangular box can be highlighted in a manner different from when it is pointed at, or a bolder border can be drawn around it. When a choice is selected, any other selected choice must be deselected or made inactive.

Defaults. If a selection field is displayed with a choice previously selected or a default choice, display the currently active choice in the same manner as when it is selected.

MYTH **Our software is highly usable — it includes all the latest interface widgets.**

Check Boxes

- Description:
  - — A two-part control consisting of a square box and choice description.
  - — Each option acts as a switch and can be either "on" or "off."
    - When an option is selected (on), a mark such as an "X" or "check" appears within the square box, or the box is highlighted in some other manner.
    - Otherwise the square box is unselected or empty (off).
  - — Each box can be
    - Switched on or off independently.
    - Used alone or grouped in sets.
- Purpose:
  - — To set one or more options as either on or off.
- Advantages:
  - — Easy-to-access choices.
  - — Easy-to-compare choices.
  - — Preferred by users.
- Disadvantages:
  - — Consume screen space.
  - — Limited number of choices.
  - — Single check boxes difficult to align with other screen controls.
- Proper usage:
  - — For setting attributes, properties, or values.
  - — For nonexclusive choices (that is, more than one can be selected).
  - — Where adequate screen space is available.
  - — Most useful for data and choices that are
    - Discrete.
    - Small and fixed in number.
    - Not easily remembered.
    - In need of a textual description to describe meaningfully.
    - Most easily understood when the alternatives can be seen together and compared to one another.
    - Never changed in content.
  - — Can be used to affect other controls.
  - — Use only when both states of a choice are clearly opposite and unambiguous.

Description. *Check box* controls, also referred to as *tick box* or *ballot 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), an X or check appears within the square box, or it is highlighted in some other manner. When not selected, the square box is unselected or empty (off). Each box can be switched on or off independently. Check boxes may be used alone or grouped in sets.

Check boxes, too, may take different physical forms and be called by different names. The most common name is *check boxes*, the name used by Microsoft

Windows. Others names include *toggle buttons*, *switches*, and *two state nonexclusive settings*. Not only their names differ; differences also exist in the way these fields are presented on screens. One very common display method is a check box, which, resembling its namesake, consists of a square placed adjacent to each alternative. When the choice is selected, some systems place an X in the square to provide a visual indication that it is active. Others, including Microsoft Windows, place a check mark in the square, or fill in the selected square or make it look depressed when selected.

Interestingly, in the past decade, Microsoft Windows and others have switched from Xs to checks as the "on" mark in a check box. This has occurred because of possible confusion concerning Xs that have existed in some using communities. In an engineering environment, for example, an X marked in a box means not applicable or not set, while a check mark customarily means active or set. Internationally, also, an X is not universally recognized. (This control is called a check box, isn't it?)

Another style for this type of field is a button or box with the choice description inscribed inside. When selected, the alternative is highlighted in some way. To distinguish these fields visually from similarly constructed fields presenting mutually exclusive choices, the buttons are not adjacent to, or butted up against, one another. Check boxes are illustrated in Figures 7.39 and 7.40. Again, deciding on which style to use seems to be more a matter of preference than performance (other than for the possible confusion of Xs and checks). No published comparison studies are available for guidance.

Purpose. The purpose of a check box, then, is to set one or more options either on or off.

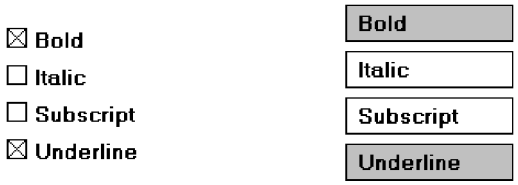


Figure 7.39: Check boxes.

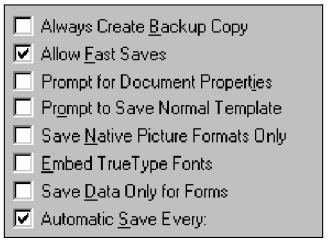


Figure 7.40: Check boxes.

Advantages/disadvantages. With check boxes, all alternatives are always visible. Therefore, it is easy to access and compare choices. Like radio buttons, check boxes were the preferred, and fastest to use, controls in the Johnsgard et al. (1995) study. One disadvantage is the large amount of screen real estate they consume, limiting the number of alternatives that can be efficiently displayed. Another potential disadvantage is that it can be difficult to align a single check box with other arrayed screen controls because they often possess long descriptions for clarity purposes.

Proper usage. Check boxes are useful for setting attributes, properties, or values where adequate screen space is available. The alternatives should be discrete, small in number, and in need of a textual description to identify meaningfully. Check boxes are helpful in situations where the alternatives cannot always be easily remembered and the displaying of the alternatives together aids in understanding and selecting the proper choice. The choices displayed should be stable, never changing in content.

Check boxes can be used to affect other controls. The contents of a list can, for example, be filtered by setting check boxes. Use a check box only when both states of a choice are clearly opposite and unambiguous. If opposite states are not clear, use two radio buttons, clearly stating the opposite states.

Choice Descriptions

- Provide meaningful, fully spelled-out choice descriptions clearly describing the values or effects set by the check boxes.
  — Do not use negatives in the description.
- Display them in a single line of text.
- Display them using mixed-case letters in sentence style.
- Position descriptions to the right of the check box. Separate by at least one space from the box.
- When a choice is unavailable for selection under a certain condition, display the choice description visually dimmed.

Choice descriptions must be clear, meaningful, fully spelled out, and displayed in a mixed-case text. Do not use negatives (such as *dis-*) in the description because the chance for a confusing double negative exists. Always use a positive statement such as *enable*, not *disable*. For multiword descriptions, capitalize the first letter and present the description in the sentence style. Small box-type indicators should be located to the left of the choice description; rectangular boxes that resemble a command button will find the description within the box. Small boxes associated with text are advantageous when the choice description must be lengthy. Descriptions in boxes impose restrictions on the number of words that can be inscribed within them. When a choice is conditionally unavailable for selection, display it grayed out or dimmed.

Size

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

Generally, selection fields of this style should not offer more than eight choices. 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. Johnsgard et al. (1995) have found, however, that even for as many as 30 choices, check boxes were preferred by users and performed better than other nonexclusive controls.

Defaults

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

Provide a default setting for a check box whenever possible. If a multiple selection is performed and the values in the selection differ, display the applicable check boxes in the mixed value state, or in another distinctive manner such as with a gray shadow.

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.

Figure 7.41

- 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.

Figure 7.42

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

Figure 7.43

Provide groupings of related check boxes. The preferred check box orientation is columnar. This aids scanning and choice comparison. Controls with small box indicators usually fit best in this manner because choice descriptions are not restricted in size. Left-align the check boxes and choice descriptions. Rectangular boxes should be of equal width and separated from one another by small and equidistant spaces. This will distinguish them from mutually exclusive choices that will be butted up against one another. If the boxes must be horizontally oriented, provide adequate separation between them. Enclose the boxes in a border to emphasize their relationship. Figure 7.44 illustrates ways to, and not to, present groupings of check boxes.

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.

Selection choices should be organized logically. If the alternatives have an expected order, follow it. Other ordering schemes such as frequency of use, sequence of use, or importance may also be considered. Always begin the ordering at the top or left. If, under certain conditions, a choice is not available, display the unavailable choice subdued or less brightly than the available choices.

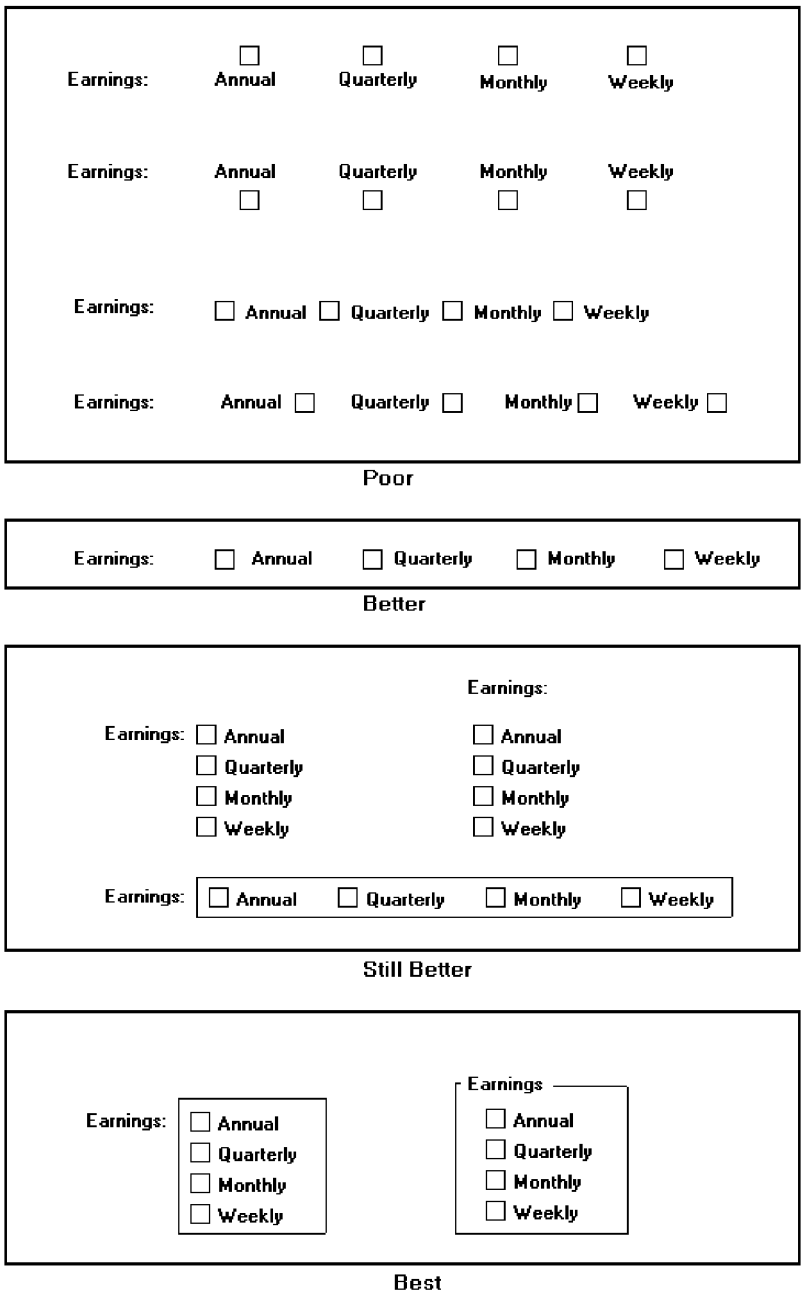


Figure 7.44: Ways to, and not to, present check boxes.

Related Control

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

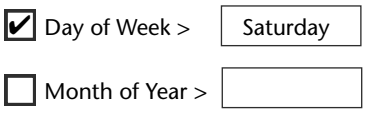


Figure 7.45

Position any control related to a check box immediately to the right of the choice description. If a the check box label also acts as the label for the control that follows it, present it mixed case, sentence style text and end the label with an arrow (>) to relate the choice description to the control.

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.
    - 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.

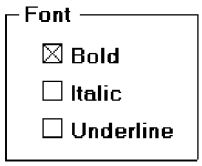


Figure 7.46

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

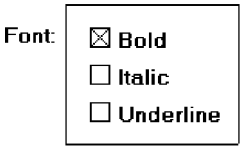


Figure 7.47

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

Figure 7.48

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

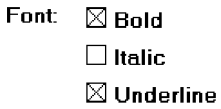


Figure 7.49

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

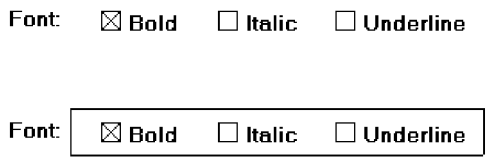


Figure 7.50

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

Figure 7.51

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

Structure. Using a static text or group box control, provide a caption for each grouping of related check boxes. Display the caption fully spelled out using mixed-case letters. Some occasional common abbreviations may be used, however, to achieve the alignment goals to be specified.

Columnar orientation. The preferred location of a check box control caption within a screen can vary. Ideally, the caption is placed upper-left-justified within a line border, or group box, surrounding columnar-oriented check boxes as shown in the first example in the preceding guideline summary. If other controls on a screen possess captions positioned to the left, and the check box control is aligned

with these controls, position the caption to the left of the control. This is the second example previously illustrated . This will help achieve screen efficiency, minimize viewer eye movements, and provide caption and choice distinctiveness. Without an enclosing border, position the caption left-justified above the choice descriptions, or to the left of the topmost choice.

Horizontal orientation. If horizontal orientation is necessary, position the caption to the left of the choice descriptions, or left-justified within an enclosing control border. If the screen contains only one related grouping of check boxes, the screen title may serve as the control caption. 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.

Figure 7.52

Assign unique keyboard mnemonics for each check box 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 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.

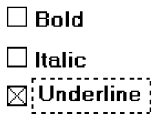


Figure 7.53

- 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.

Figure 7.54

  — 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.

Pointing. The selection target area should be as large as possible in order to make it easy to move to. If a small check box is the selection indication method used, the target area should include the box and the choice description text. If the rectangular box selection method is used, the entire box should be the target. Highlight the selection choice in some visually distinctive way when the pointer is resting on it and the choice is available for selection. If a check box is the selection indication method used, a distinctive reverse video, reverse color, or dotted or dashed box selection cursor or bar may be used to surround the selected choice description. This cursor should be as long as the longest description plus one space at each end. The cursor should not cover the check box.

Activation. When a choice is selected, distinguish it visually from the unselected choices. A check box may be marked with an X or check or filled in. Other methods include making the button look depressed or raised through appropriate use of drop shadows. A rectangular box can be highlighted in a manner different from when it is pointed at, or a bolder border can be drawn around it. The style chosen must be consistently applied throughout an application or system.

Defaults. If a selection field is displayed with a choice previously selected or a default choice, display the currently active choice in the same manner as when it is selected.

Select/deselect all. Do not use Select All and Deselect All check boxes. If this option is necessary, consider using a multiple selection list box with command buttons to accomplish these actions.

Mixed-value state. A check box can have three states.

- Checked — the associated property or value is set.
- Cleared — the associated value or property is not set.
- Mixed value — the associated value is set for some, but not all elements of the selection.

An example of the mixed value state would be when a sentence is selected and the selected text is partly bold and partly normal. So, when a check box represents a value, and a selection encompasses multiple value occurrences set in both the on and off state, display the check box in the mixed value state. Fill the check box with another easily differentiable symbol or pattern. Toggle the check box through clicking as described in the preceding guidelines.

Palettes

- Description:
  - — A control consisting of a series of graphical alternatives. The choices themselves are descriptive, being composed of colors, patterns, or images.
  - — In addition to being a standard screen control, a palette may also be presented on a pull-down or pop-up menu or a toolbar.
- Purpose:
  - — To set one of a series of mutually exclusive options presented graphically or pictorially.
- Advantages:
  - — Pictures aid comprehension.
  - — Easy-to-compare choices.
  - — Usually consume less screen space than textual equivalents.
- Disadvantages:
  - — A limited number of choices can be displayed.
  - — Difficult to organize for scanning efficiency.
  - — Requires skill and time to design meaningful and attractive graphical representations.
- Proper usage:
  - — For setting attributes, properties, or values.
  - — For mutually exclusive choices (that is, only one can be selected).
  - — Where adequate screen space is available.
  - — Most useful for data and choices that are
    - Discrete.
    - Frequently selected.
    - Limited in number.
    - Variable in number.
    - Not easily remembered.
    - Most easily understood when the alternatives may be seen together and compared to one another.
    - Most meaningfully represented pictorially or by example.

- Can be clearly represented pictorially.
- Rarely changed in content.

— Do not use

- Where the alternatives cannot be meaningfully and clearly represented pictorially.
- Where words are clearer than images.
- Where the choices are going to change.

Description. Like radio buttons, palettes can also be used to present two or more mutually exclusive alternatives. The choices presented, however, are visually descriptive within themselves, no choice descriptions being needed to identify them. Examples of palettes might be fill-in colors, patterns, or different shades of a color. A palette may also be referred to as *value set* or *well*. A palette is illustrated in Figure 7.55. In addition to being a standard screen control, a palette may also be presented on a pull-down or pop-up menu, included in a toolbar, or be contained in a Palette window.

Purpose. A palette's purpose is to set one of a series of mutually exclusive options presented in a graphic or pictorial form.

Advantages/disadvantages. Palettes are preferable to radio buttons in that they take up less space and allow the viewer to focus on the visual characteristics of the choice itself, instead of having to read the choice text and cross-reference it to a radio button. Some qualities, such as colors, patterns, and shades, are much more easily comprehended when they are actually seen. While a larger number of choices can be presented than with radio buttons, there is still a limit to how many can be practically displayed. Because of their larger size, palettes are also more difficult to organize for scanning efficiency. Palettes also require skill and time to design meaningful and attractive graphical representations.

Proper usage. Palettes are used for setting attributes, properties, or values of mutually exclusive choices where adequate screen space is available. Consider using a palette when the choices have qualities that can be best described by actual illustration. They are useful for data and choices that are discrete and limited in number. They are most useful when the choices' being seen together and compared to one another aids identification and selection of the proper alternative. They are also most useful when the alternatives can be meaningfully and clearly represented pictorially or by example. Palettes should rarely change in content.

Palettes should be displayed in the proper manner. If the attributes on a palette must be available at all times, place them on a standard control or fixed palette. If the attributes on the palette are sometimes used frequently and other times used infrequently, place them on a pop-up or tear-off menu. Do not place frequently used palettes on pull-down menus.

Also, from a presentation perspective, do not use a palette if the alternatives cannot be meaningfully and clearly represented pictorially. In addition, do not use one where words are clearer than images, or in situations where the choices are going to change.

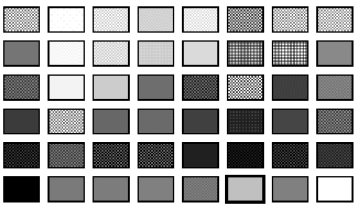


Figure 7.55: Palette.

Graphical Representations

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

Provide meaningful, accurate, and clear illustrations of alternative choices. Create equal size images large enough to illustrate clearly the available alternatives and permit ease in pointing and selecting. Always test illustrations with users before implementing them, to ensure that they will work satisfactorily. While most palettes will not possess textual choice descriptions, under certain circumstances textual descriptions may be needed. For example, a choice might require selection of a style of font. The palette may contain the names of the available styles (such as Roman) with the text displayed as the font style would actually appear.

Size

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

Because palettes will consume less screen space, more choices can be displayed in a given area of a screen than can be displayed when using textual descriptions. Present all available alternatives within the limits imposed by how big the graphical representations must be and the capabilities of the display hardware in creating clear illustrations. Limitations in people's ability to accurately differentiate the kinds of graphical representations being presented must also be considered.

Layout

- Create boxes large enough to
  — Effectively illustrate the available alternatives.
  — Permit ease in pointing and selecting.
- Create boxes of equal size.
- Position the boxes adjacent to, or butted up against, one another.
- A columnar orientation is the preferred manner.
- If vertical space on the screen is limited, orient the choices horizontally.

Palette boxes must be large enough to illustrate effectively the available alternatives and to maximize ease in selecting. Created boxes should be of equal size and positioned adjacent to, or butted up against one another, because they are mutually exclusive choices. Columns are preferred, but horizontal rows can be used if space constraints exist on the screen.

Organization

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

Palettes should be organized logically. If the alternatives have an expected order, follow it. Colors, for example, should be ordered from the right or top by their spectral position: red, orange, yellow, green, blue, indigo, and violet. If an expected or normal order does not exist, arrange choices by frequency of occurrence, sequence of use, or importance. Palettes with text may be arranged alphabetically. If, under certain conditions, a choice is not available, display the unavailable choice subdued or less brightly than the available 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.

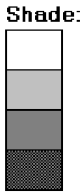


Figure 7.56

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

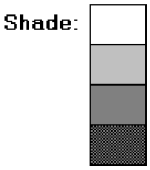


Figure 7.57

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

Figure 7.58

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

Figure 7.59

Use a static text control to provide a caption for each palette. In screens containing only one palette, the screen title may serve as the caption. Display the caption fully spelled out using mixed-case letters, although some abbreviations may be used to achieve the alignment goals to be specified. Captions may be located above or to the left of the palette, as previously shown. With a horizontal orientation, the caption may be positioned above the palette or to the left of the alternatives. Positioning on any one screen will be dependent on other caption-control relationships within the screen.

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.

Pointing. The selection target should be as large as possible in order to make it easy to move to. 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. 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 bolder border around it.

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

List Boxes

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

- Proper usage:
  - — For selecting values or setting attributes.
  - — For choices that are
    - Mutually exclusive (only one can be selected).
    - Non-mutually exclusive (one or more may be selected).
  - — Where screen space is available.
  - — For data and choices that are
    - Best represented textually.
    - Not frequently selected.
    - Not well known, easily learned, or remembered.
    - Ordered in an unpredictable fashion.
    - Frequently changed.
    - Large in number.
    - Fixed or variable in list length.
  - — When screen space or layout considerations make radio buttons or check boxes impractical.

Description. A *list box* is a permanently displayed rectangular box control that contains a list of values or attributes from which single or multiple selections are made. It can also be referred to as a *fixed list box* because it is fixed on the screen. In Java, they are called *lists*, and in HTML, *selection lists/scrolling lists*. The choices are usually text, but they may be pictorial representations or graphics as well. A list box may be scrollable to view large lists, and the user uses a mouse to point and click to make selections. No text entry field exists in which to type text, but a single-selection list box may be associated with a text box where the selected choice is displayed or an item may be added to the list. Examples of single-selection list boxes are illustrated in Figure 7.60.

Purpose. The purpose of a list box is display a collection of items. The choices may be mutually exclusive (single-selection) or not mutually exclusive (multiple-selection).

Advantages/disadvantages. List boxes are always visible, reminding users of the choices available. They permit an unlimited number of options to be displayed. Among their disadvantages are the excessive screen space they consume and the possible necessity for time-consuming scrolling to see all items. Because the list content can change, and items can be ordered in an unpredictable way, it can be hard to find items.

Proper usage. List boxes are used for selecting objects or values or setting attributes, either mutually exclusive or not, where sufficient screen space is available to display six to eight choices. Their best use is for data and choices that are textual; large in number; fixed or variable in list length; not well known, easily learned or remembered; and ordered in an unpredictable fashion. List box items should not have to be selected or changed frequently. List boxes may be used when screen space, list size, and data volatility considerations make use of radio buttons and check boxes impractical. The content of a list box is easier to change than that of radio buttons and check boxes.

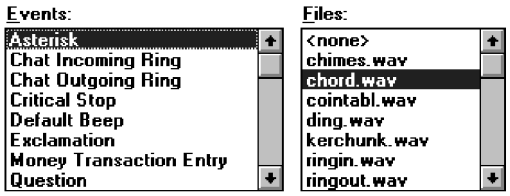


Figure 7.60: List boxes.

List Box General Guidelines

First, general list box guidelines will be presented. Then, specific guidelines for single- and multiple-selection lists will be reviewed.

Selection Descriptions

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

Selection descriptions will reflect the selection alternatives available. They should be meaningful, fully spelled out, and follow the sentence style of presentation. Array the descriptions into columns. Meaningful ordering schemes include logical order, frequency of use, sequence of use, and importance. If no such pattern exists, arrange the list alphabetically. Display the list of choices using mixed-case letters.

List Size

- No 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.

A list being displayed in a fixed list box has no actual size limit. All available alternatives should be capable of being displayed. Searching a very long list, however, can be very time-consuming. A list should not require more than 40 page-downs to completely search it. If more are necessary, provide a method for using search criteria or scoping the options, perhaps through a first-letter search.

Box Size

- Show as many items as possible, considering screen space constraints.
- Minimally, must be long enough to display six to eight items 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.

Figure 7.61

- — 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.

The exact size of a fixed list box will depend on its function and screen space constraints. Generally, boxes should be restricted to displaying no more than eight choices at one time. Larger boxes that eliminate the need for scrolling, however, are preferable to list boxes that require scrolling. If screen space constraints exist, the box may be reduced in size to display at least three items. If scrolling is necessary, include a scroll bar on the right side of the box.

The list box should be wide enough to display fully all item wording. 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 the long entries with an ellipsis in the middle. If breaking long entries, preserve the important characteristics needed to distinguish them. When shortening an item's name in this way, include a ToolTip that displays the item's full name.

As a last resort, provide horizontal scrolling and a scroll bar at the bottom of the list box. Many people dislike horizontal scrolling, however.

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.

Choices should be organized logically to permit easy browsing. If the alternatives have an expected order, follow it. Other ordering schemes such as frequency of use, sequence of use, or importance may also be considered. When no obvious scheme exists, use alphabetic order. You can provide 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, it should be omitted from the list. If it is important that the existence and unavailability of a list item be communicated, however, 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.

Enclose the box in a solid border in the color of the choice descriptions. To provide adequate legibility, leave one space between the choice descriptions and the left border, and one space between the longest choice description and the right border.

Captions

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

Figure 7.62

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

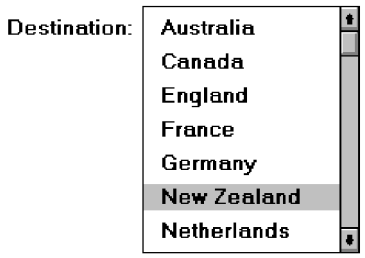


Figure 7.63

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

To identify the list box, a field caption in mixed-case letters, with each significant word capitalized is necessary. A list box does not have a caption, so create one using a static text control. Place this caption either above the upper-left corner of the box or to the left of the first choice description. The caption style chosen will again be dependent upon caption-control relationships in other controls included within the screen. It should be consistently oriented with the other control captions.

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.

Figure 7.64

— Mark the selected choice in a distinguishing way.

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

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. One method used for this is to place a bold border around the choice.

Selection. Indicate the selected choice through use of a reverse video or reverse color bar, as wide as the box itself. Visually differentiate multiple-choice (nonexclusive) from single-choice (mutually exclusive) fixed list boxes, as described in the following sections.

Activation. Require the pressing of a command button when an item, or items, is selected. Double-clicking is difficult for many people. Always provide for a single click followed by a button press.

Single-Selection List Boxes

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

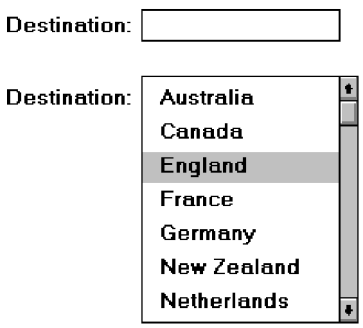


Figure 7.65

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

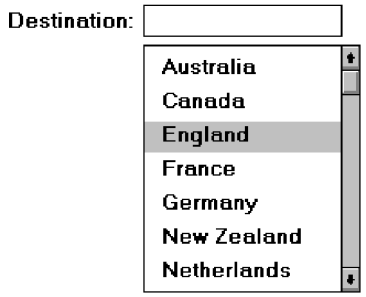


Figure 7.66

— 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.

— Other:

- Follow other relevant list box guidelines.

Purpose. A *single-selection* list box is used for selecting only one item in a list. It provides a mutually exclusive operation similar to a group of radio or option buttons. This kind of list box, however, can handle a large number of items more efficiently.

Related text box. If the list box is associated with a text field, position the list box below and as close as possible to the related text box. If this cannot be accomplished, position the text box to the left. Captions of related text boxes and list boxes must be worded similarly. If, however, the text box and the list box are located in close physical proximity, the caption may be omitted from the list box. Visually relate a list box to a text box by using the same background color for both boxes.

For single-selection fixed list boxes, indicate an active choice by highlighting it or marking it with a circle or diamond to the left of the choice description. If the list represents mixed values for a multiple selection, do not highlight any entry.

Extended and Multiple-Selection List Boxes

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

Figure 7.67

- 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.

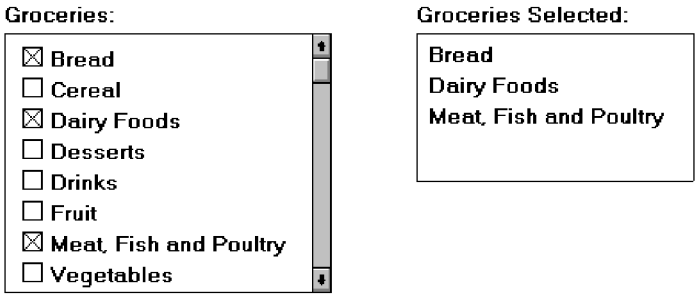


Figure 7.68

  — 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.

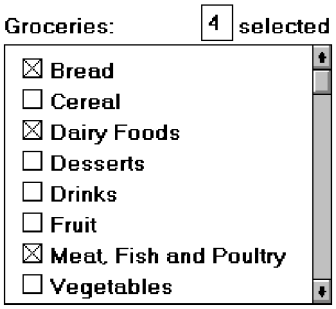


Figure 7.69

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

— Defaults:
- When the list box is first displayed:
  — Display the currently active choices highlighted.
  — Mark with an X or check mark to the left of the entry.
  — If the list represents mixed values for a multiple selection, do not highlight an entry.

— Other:
- Follow other relevant list box guidelines.

Purpose. *Multiple-selection* list boxes permit selection of multiple items from a long listing. They provide a nonexclusive operation similar to a group of check boxes. This kind of list box, however, can handle a large number of items more efficiently. *Extended list* boxes are optimized for individual item or range selection. Multiple-selection list boxes are optimized for independent item selection.

Selection indication. For choice selections, mark them with an X or check mark to the left of the entry. Also consider providing a *summary list* box, another list box containing a compilation of the active selections from the multiple-selection list box. This will permit quick scanning and comparison of these active choices and greatly reduce the need for scrolling if the selectable list is lengthy. The summary list box can be made scrollable, if necessary. Position the summary list adjacent to, and to the right of, the multiple-selection list box. Use the same colors for the summary list box and the multiple-selection list box. Include 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.

Also consider providing a *display-only text box* control indicating how many choices have been selected in the multiple-selection list box. This text box can be associated with either the multiple-selection or summary list box. It is useful in situations where the multiple selections may be numerous and all the choices cannot be seen without scrolling. It is also useful when the user must know exactly how many choices have been selected. Position this text box justified upper-right above the list box.

Select All and Deselect All buttons. Provide command buttons to accomplish fast "select all" and "deselect all" actions, when these actions must be frequently or quickly performed.

Defaults. When the list box is first displayed, the active selection will depend on previous activities. If a choice has been previously selected, display the currently active choice in the same manner used when it was selected. If the list represents mixed values for a multiple selection, do not highlight any list entries.

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.

Description. A *list view* control is a special extended-selection list box that displays a collection of items, each consisting of an icon and a label. List view controls can display content in four different views: large icon, small icon, list, and report. The control also supports options for aligning, selecting, and sorting icons, and for editing icon labels.

Purpose and usage. Use list views when the representation of objects as icons is appropriate, or to represent items with multiple columns of information.

Drop-Down/Pop-Up List Boxes

- Description:
  - — A single rectangular control that shows one item with a small button to the right side.
    - The button provides a visual cue that an associated selection box is available but hidden.
  - — When the button is selected, a larger associated box appears, containing a list of choices from which one may be selected.

— Selections are made by using the mouse to point and click.
— Text may not be typed into the control.

- Purpose:
  — To select one item from a large list of mutually exclusive options when screen space is limited.
- Advantages:
  — Unlimited number of choices.
  — Reminds users of available options.
  — Conserves screen space.
- Disadvantages:
  — Requires an extra action to display the list of choices.
  — When displayed, all choices may not always be visible, requiring scrolling.
  — The list may be ordered in an unpredictable way, making it hard to find items.
- Proper usage:
  — For selecting values or setting attributes.
  — For choices that are mutually exclusive (only one can be selected).
  — Where screen space is limited.
  — For data and choices that are
    - Best represented textually.
    - Infrequently selected.
    - Not well known, easily learned, or remembered.
    - Ordered in a unpredictable fashion.
    - Large in number.
    - Variable or fixed in list length.
  — Use drop-down/pop-up lists when
    - Screen space or layout considerations make radio buttons or single-selection list boxes impractical.
    - The first, or displayed, item will be selected most of the time.
  — Do not use a drop-down list if it important that all options be seen together.

Description. A *drop-down/pop-up list box* is a single rectangular field with a small button to the side and an associated hidden list of options. In Java, they are called *choice/pop-up lists*, in HTML, *selection lists/pop-up menus*. The button provides a visual cue to the user that an associated selection box of choices is hidden but available on demand. When requested, a larger associated rectangular box appears containing a scrollable list of choices from which one is selected. Selections are made by using the mouse to point and click. No text field exists in which to type text.

Fields of this nature go by many different names. 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*. Other list boxes of this type seem to pop up on the screen, either next to or over the selection field. Even a Microsoft Windows "drop-down" will "pop up" if it is opened near the bottom of the screen. In this discussion, these variously named controls will be given the generic name of drop-down/pop-up list boxes. A drop-down list is illustrated in Figures 7.70 and 7.71. Figure 7.72 shows a pop-up list.

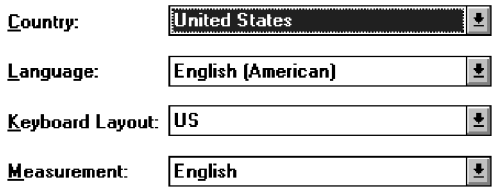


Figure 7.70: Drop-down list boxes. There are four unopened boxes: Country, Language, Keyboard Layout, and Measurement.

Figure 7.71: Drop-down list box opened for Country.

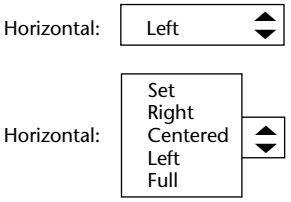


Figure 7.72: Pop-up list box, closed and opened.

Purpose. The purpose of these list boxes is to permit selection from a large set of mutually exclusive choices when screen space is scarce.

Advantages/disadvantages. The most useful feature of drop-down/pop-up list boxes is that they conserve screen space. They may be retrieved on demand, reminding users of the choices available. They permit an unlimited number of options to be displayed.

A significant disadvantage of these lists is that they necessitate an extra step to display the available options. Scrolling may also be necessary to see all items. Because items can be ordered in an unpredictable way, they can be hard to find occasionally. Generally, drop-down/pop-up list boxes require more work on the part of the user than many other screen controls because of the activation step and the possible need for scrolling. A study comparing drop-down/pop-up lists to other similar controls will be described later in this step.

Proper usage. Drop-down/pop-up list boxes are used much like regular list boxes, except that the choices are not visible at all times. They are used for selecting values or setting attributes when sufficient screen space is not available to display the choices permanently. Their best use is for data and choices that are textual; large in number; fixed or variable in list length; not well known, easily learned or

remembered; and ordered in an unpredictable fashion. Items should not have to be selected frequently. Drop-down/ pop-up lists are most useful when screen space or layout considerations make radio buttons or single-selection list boxes impractical. It is also desirable that most users select the first, or displayed, item in the listing so the box does not have to be opened. Never use a drop-down/pop-up list if it is important that all options be seen together at one time.

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.

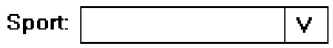


Figure 7.73

Most systems indicate the presence of a drop-down or pop-up list by associating a meaningful icon with the applicable field. This icon can be seen positioned to the left of the selection field, within the selection field as is done by Microsoft Windows, or to the right of the selection field. Other platforms do not provide any visual indication that a hidden list is available. An indication to the user that a drop-down or pop-up list is available should be indicated on the screen. This is especially critical if not all fields have associated hidden lists. The best location is to the right of the selection field where it is out of the way until needed. To visually differentiate it from another control (the drop-down/pop-up combination box), position the button abutting or within the selection field. (A drop-down/pop-up combination box button will be separated by a space.) The indicator should be large enough to provide a good pointing target.

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.

Selection descriptions will reflect what choices exist in the control. They should be meaningful, fully spelled out, and organized in columns. Display the list of choices using mixed-case letters. Box descriptions should be displayed in the same color as the selection field text. If a particular choice is not available in the current context, it should be omitted from the list.

List Size

- Not limited in size.
- Present all available alternatives.

A list being displayed in a drop-down/pop-up list box has no size limit. All available alternatives should be capable of being displayed. It would seem practical that for large scrollable lists, the same rules as presented for list boxes should also be applied. Restrict page-downs to no more than 40 and provide a method to scope actions.

Box Size

- Long enough to display six to eight 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.

Drop-down/pop-up list boxes should be restricted to eight or fewer choices. If more must be displayed, permit scrolling and include a scroll bar on the right side of the box. The list box should be wide enough to fully display all selection choice wording. 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 the long entries, inserting an ellipsis in the middle. If breaking entries, preserve the important characteristics needed to distinguish them. When shortening an item's name in this way, include a ToolTip that displays the item's full name. As a last resort, provide horizontal scrolling and a scroll bar at the bottom of the list box. Avoid horizontal scrolling whenever possible, however.

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.

Selection choices should be organized logically. If the alternatives have an expected order, follow it. Other ordering schemes such as frequency of use, sequence of use, or importance, may also be considered. Always begin ordering at the top or left. If, under certain conditions, a choice is not available, display the unavailable choice subdued or less brightly than the available choices.

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.

To provide adequate legibility, leave one space between the choice descriptions and the left border, and one space between the longest choice description and the right border. Extending the listing box to the right edge of the prompt button allows the user to move easily from the button to the list. To set off the box from the screen body background, use the same color background for the box as is used in the entry field. Also incorporate a solid line border around the box in the same color as the choice descriptions.

Captions

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

To identify the drop-down/pop-up list box, a field caption in mixed-case letters, with each significant word capitalized is necessary. Use a static text control to create the caption. The recommended position is to the box's left. Select a positioning consistent with other controls presented on the window.

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.

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. The list must be opened to change the choice.

Disabling

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

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.

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. Close the listing when an item is selected.

Combination Entry/Selection Controls

It is possible for a control to possess the characteristics of both a text field and a selection field. In this type of control, information may either be keyed into the field or selected and placed within it. The types of combination entry/selection fields are spin boxes, attached combination boxes, and drop-down/pop-up combination boxes.

Spin Boxes

- Description:
  — A single-line field followed by two small, vertically arranged buttons.
    - The top button has an arrow pointing up.
    - The bottom button has an arrow pointing down.
  — Selection/entry is made by
    - Using the mouse to point at one of the directional buttons and clicking. Items will change by one unit or step with each click.
    - Keying a value directly into the field itself.
- Purpose:
  — To make a selection by either scrolling through a small set of meaningful predefined choices or typing text.
- Advantages:
  — Consumes little screen space.
  — Flexible, permitting selection or typed entry.

- Disadvantages:
  - — Difficult to compare choices.
  - — Can be awkward to operate.
  - — Useful only for certain kinds of data.
- Proper usage:
  - — For setting attributes, properties, or values.
  - — For mutually exclusive choices (only one can be selected).
  - — When the task requires the option of either key entry or selection from a list.
  - — When the user prefers the option of either key entry or selection from a list.
  - — Where screen space is limited.
  - — Most useful for data and choices that are
    - Discrete.
    - Infrequently selected.
    - Well known, easily learned or remembered, and meaningful.
    - Ordered in a predictable, customary, or consecutive fashion.
    - Infrequently changed.
    - Small in number.
    - Fixed or variable in list length.

Description. A *spin box*, also called a *spin button* or an *up-down control*, is a single-line field followed by two small, vertically arranged buttons inscribed with up and down arrows. These buttons may also be referred to as up-down buttons. Selection of an item is accomplished using the mouse to point at one of the buttons and clicking. Items in a listing in the display field will change by one unit or step in the direction selected with each click. The list is searched as the ring or circle of alternatives "spins" by. Keying a value directly into the field itself may also complete a spin box. A spin box is illustrated in Figure 7.74.

Advantages/disadvantages. Spin boxes are flexible, permitting either selection or typed entry. They also consume little screen space. On the other hand, spin boxes are useful only for certain kinds of data, that which is predictable or consecutive. Because only one item is displayed at a time, it is difficult to compare choices. Spin boxes may also be awkward to operate, often requiring several back and forth iterations to bring the desired value into view.

Proper usage. Spin boxes are used for setting attributes, properties, or values that are mutually exclusive. They are useful when the task requires, or the user prefers, the option of either key entry or selection from a list. Spin boxes are useful for data and choices that are discrete and small in number. The choices themselves should be well known, easily learned or remembered, and meaningful. Choices should be ordered in a predictable, customary, or consecutive fashion so people can anticipate the next not-yet-visible choice. Items in spin boxes should not require frequent selection, and the array of items listed should be stable.

Figure 7.74: Spin boxes.

List Size

- Keep the list of items relatively short.
- To reduce the size of potentially long lists, break the listing into subcomponents, if possible.

Because the list must be manipulated to display its contents, it should be as short as feasible. To reduce the size of potentially long lists, break the listing into subcomponents whenever possible. A date, for example, may be broken into its components of month, day, and year.

List Organization

- Order the list in the customary, consecutive, or expected order of the information contained within it.
  — Ensure that the user can always anticipate the next (not-yet-visible) choice.
- When first displayed, present a default choice in the box.

Spin boxes are most effective when the values they contain have a customary or consecutive order that is predictable. Information can be letters or numbers. Examples are days of the week, months of the year, shoe sizes, and so on. The user must always be able to anticipate the next choice before it is displayed. The control should always contain a default value when first displayed.

Other Spin Box Guidelines

- Box size:
  — The spin box should be wide enough to display the longest entry or choice.
- Caption:
  — Display it using mixed-case letters.
  — Position the caption to the left of the box.
    - Alternately, it may be positioned left-justified above the box.
- Entry and selection methods:
  — Permit completion by
    - Typing directly into the box.
    - Scrolling and selecting with a mouse.
    - Scrolling and selecting with the up/down arrow keys.

— For alphabetical values,
  - Move down the order using the down arrow.
  - Move up the order using the up arrow.

— For numeric values or magnitudes,
  - Show a larger value using the up arrow.
  - Show a smaller value using the down arrow.

Box size. Fully display all alternatives within the spin box. The box should be wide enough to display the longest entry or choice.

Caption. To identify the spin box, use a static text field to provide a field caption in mixed-case letters, with each significant word capitalized. The recommended position is to the box's left. Select a positioning consistent with other controls presented on the window.

Entry and selection. Spin box completion should be possible by typing directly into the field or by scrolling and selecting options with a mouse or keyboard keys. When spinning alphabetical values, move down the order using the down arrow and up the order using the up arrow. For numeric values or magnitudes, display a larger value using the up arrow and a smaller value using the down arrow.

Combo Boxes

- Description:
  — A single rectangular text box entry field, beneath which is a larger rectangular list box (resembling a drop-down list box) displaying a list of options.
  — The text box permits a choice to be keyed within it.
  — The larger box contains a list of mutually exclusive choices from which one may be selected for placement in the entry field.
    - Selections are made by using a mouse to point and click.
  — As text is typed into the text box, the list scrolls to the nearest match.
  — When an item in the list box is selected, it is placed into the text box, replacing the existing content.
  — Information keyed may not necessarily have to match the list items.
- Purpose:
  — To allow either typed entry in a text box or selection from a list of options in a permanently displayed list box attached to the text box.
- Advantages:
  — Unlimited number of entries and choices.
  — Reminds users of available options.
  — Flexible, permitting selection or typed entry.
  — Entries not necessarily restricted to items selectable from list box.
  — List box always visible.
- Disadvantages:
  — Consumes some screen space.
  — All list box choices not always visible, requiring scrolling.

— Users may have difficulty recalling sufficient information to type entry, making text box unusable.
— The list may be ordered in an unpredictable way, making it hard to find items.

■ Proper usage:
— For entering or selecting objects or values or setting attributes.
— For information that is mutually exclusive (only one can be entered or selected).
— When users may find it practical to, or prefer to, type information rather than selecting it from a list.
— When users can recall and type information faster than selecting it from a list.
— When it is useful to provide the users a reminder of the choices available.
— Where data must be entered that is not contained in the selection list.
— Where screen space is available.
— For data and choices that are
- Best represented textually.
- Somewhat familiar or known.
- Ordered in an unpredictable fashion.
- Frequently changed.
- Large in number.
- Variable or fixed in list length.

Description. A *combo box*, also known as an *attached combination box,* is a single rectangular entry field, beneath which is a larger rectangular box (resembling a drop-down list box) displaying a list of options. In Java, combo boxes are called *editable choice pop-up lists*. The entry field permits a choice to be keyed within it, while the larger box contains a list of mutually exclusive choices, from which one may be selected for placement in the entry field. A combo box combines the capabilities of both a text box and a list box. It visually resembles a drop-down list box or drop-down combo box (to be described). The text box and its associated list box have a dependent relationship. As text is typed into the text box, the list scrolls to the nearest match. Also, when an item in the list box is selected, that item is placed within the text box, replacing the existing content. When typing into the field, the information keyed does not have to match the list items. Combo boxes are illustrated in Figure 7.75.

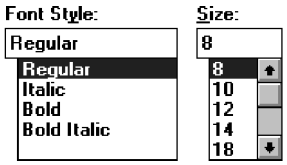


Figure 7.75: Combo boxes.

Advantages/disadvantages. Combo boxes are flexible, permitting selection or typed entry. Alternatives are always visible, or retrievable, reminding people of the available options. An unlimited variety of entries and choices are possible. Entries are not necessarily restricted to items selectable from a box. Combo boxes do consume quite a bit of screen space. Because all box choices may not be visible, some scrolling may be required. It is always possible that people may have difficulty recalling sufficient information to type, making the text box unusable. The list may also be ordered in an unpredictable way, making it hard to find items. Additional work is required of the user if selection scrolling must be performed.

Proper usage. Combo boxes are useful for entering or selecting objects or values or setting attributes that are mutually exclusive. They are most valuable when users may find it practical to, or prefer to, type information rather than selecting it from a list, but where reminders of alternatives available must occasionally be provided. They are also useful when the listings are dynamic and changeable, permitting the user to key items not contained on the list in the box. They do require that screen space be available to display them, but they eliminate the extra steps involved in retrieving drop-down lists. Combo boxes are useful for textual data and choices that are frequently changed and somewhat familiar or known, fostering keyed entry. The lists may be long, variable, and ordered in an unpredictable fashion.

Combo Box Guidelines

For the text box entry field, see "Text Box/Single Line" guidelines. For the list box, see "Drop-Down/Pop-Up List Boxes" guidelines.

Drop-Down/Pop-Up Combo Boxes

- Description:
  - — A single rectangular text box with a small button to the side and an associated hidden list of options.
    - The button provides a visual cue that an associated selection box is available but hidden.
  - — When requested, a larger associated rectangular box appears, containing a scrollable list of choices from which one is selected.
  - — Selections are made by using the mouse to point and click.
  - — Information may also be keyed into the field.
  - — As text is typed into the text box, the list scrolls to the nearest match.
  - — When an item in the list box is selected, it is placed into the text box, replacing the existing content.
  - — The information keyed does not necessarily have to match list items.
  - — Combines the capabilities of both a text box and a drop-down/pop-up list box.
- Purpose:
  - — To allow either typed entry or selection from a list of options in a list box that may be closed and retrieved as needed.

- Advantages:
  — Unlimited number of entries and choices.
  — Reminds users of available options.
  — Flexible, permitting selection or typed entry.
  — Entries not restricted to items selectable from list box.
  — Conserves screen space.
- Disadvantages:
  — Requires an extra step to display the list of choices.
  — When displayed, all box choices may not always be visible, requiring scrolling.
  — User may have difficulty in recalling what to type.
  — 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 entering or selecting objects or values or setting attributes.
  — For information that is mutually exclusive (only one can be entered or selected).
  — When users may find it practical to, or prefer to, type information rather than selecting it from a list.
  — When users can recall and type information faster than selecting from a list.
  — When it is useful to provide the users with an occasional reminder of the choices available.
  — Where data must be entered that is not contained in the selection list.
  — Where screen space is limited.
  — For data and choices that are
    - Best represented textually.
    - Somewhat familiar or known.
    - Ordered in an unpredictable fashion.
    - Frequently changed.
    - Large in number.
    - Variable or fixed in list length.

Description. A *drop-down/pop-up combo box* is a single rectangular field with a small button to the side and an associated hidden list of options. The button provides a visual cue to the user that an associated selection box of choices is available but hidden. When requested, a larger associated rectangular box appears, containing a scrollable list of choices from which one can be selected. One makes selections by using the mouse to point and click. The text box and its associated list box have a dependent relationship. As text is typed into the text box, the list scrolls to the nearest match. Also, when an item in the list box is selected, that item is placed within the text box, replacing the existing content. It closely resembles a drop-down/pop-up list box. Information, however, may also be keyed into the field. The information keyed does not necessarily have to match items in the list. A drop-down/pop-up combination box, therefore, combines the capabilities of both a text box and a selection field. A drop-down combo box is illustrated in Figures 7.76 and 7.77.

Figure 7.76: Windows 3.1 Drop-down combo box, closed.

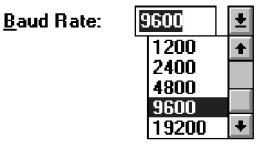


Figure 7.77: Windows 3.1 Drop-down combo box, opened.

Advantages/disadvantages. Drop-down/pop-up combo boxes are flexible, permitting selection or typed entry. They conserve screen space, but alternatives are always retrievable, reminding people of the available options. An unlimited variety of entries and choices are possible. Entries are not restricted to items selectable from a box.

In terms of disadvantages, they necessitate an extra step to display the available options. Scrolling may also be necessary to see all items. Because the list content can change, and items can be ordered in an unpredictable way, it can be hard to find items. It is always possible also that people may have difficulty recalling sufficient information to type, making the entry field unusable. Generally drop-down/pop-up combination boxes require more work on the part of the user than many other screen controls.

Proper usage. Drop-down/pop-up combo boxes are useful for entering or selecting objects or values or setting attributes that are mutually exclusive. They are most valuable when users may find it practical to, or prefer to, type information rather than selecting from a list but where reminders of alternatives available must occasionally be provided. The box may only be retrieved as needed, thereby conserving screen space. They are also useful when the listings are dynamic and changeable, permitting the user to key items not contained on the list in the box. Drop-down/pop-up combo boxes are useful for textual data and choices that are frequently changed and somewhat familiar or known, fostering keyed entry. The list may be long, variable, and ordered in an unpredictable fashion.

Prompt Button

- Provide a visual cue that a list box is hidden by including a downward-pointing arrow to the right of the text box.
- Separate the button from the text box by a small space.

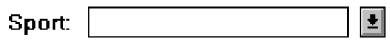


Figure 7.78

Provide a visual cue that a list box is hidden by including a downward-pointing arrow to the right of the entry field. Unfortunately, Microsoft Windows has provided drop-down list boxes and combo boxes that are visually almost identical. The only way to differentiate the two types is to click and see whether a box can be typed into. This is extremely poor design. Each unique control should be identifiable by the way it looks so trial and error behavior can be avoided. Microsoft, in Windows 3.1, did provide this distinction. The prompt button was slightly separated from the text box, as shown in Figures 7.76 and 7.77. Ideally, then, position the prompt button separated by a space from the associated text box.

Other Guidelines

For the text box entry field, see the "Text Box/Single Line" guidelines. For the box and selection components, see the "Drop-Down/Pop-Up List Boxes" guidelines.

Other Operable Controls

Other more specialized operable controls also exist. Among them are sliders, tabs, date-pickers, and scroll bars.

Slider

- Description:
  - — A scale exhibiting degrees of a quality on a continuum.
  - — Includes the following:
    - A shaft or bar.
    - A range of values with appropriate labels.
    - An arm indicating relative setting through its location on the shaft.
    - Optionally, a pair of buttons to permit incremental movement of the slider arm.
    - Optionally, a text box for typing or displaying an exact value.
    - Optionally, a detent position for special values.
  - — May be oriented vertically or horizontally.
  - — Selected by using the mouse to
    - Drag a slider across the scale until the desired value is reached.
    - Point at the buttons at one end of the scale and clicking to change the value.
    - Keying a value in the associated text box.
- Purpose:
  - — To make a setting when a continuous qualitative adjustment is acceptable, it is useful to see the current value relative to the range of possible values.
- Advantages:
  - — Spatial representation of relative setting.
  - — Visually distinctive.

- Disadvantages:
  - — Not as precise as an alphanumeric indication.
  - — Consumes screen space.
  - — Usually more complex than other controls.
- Proper usage:
  - — To set an attribute.
  - — For mutually exclusive choices.
  - — When an object has a limited range of possible settings.
  - — When the range of values is continuous.
  - — When graduations are relatively fine.
  - — When the choices can increase or decrease in some well-known, predictable, and easily understood way.
  - — When a spatial representation enhances comprehension and interpretation.
  - — When using a slider provides sufficient accuracy.

Description. A *slider* is a scale that exhibits the amount or degree of a quantity or quality on a continuum (see Figure 7.79). It is sometimes called a track bar control. A slider incorporates the range of possible values and includes a shaft or bar representing the range, the values themselves with appropriate labels, and a visual indication of the relative setting through the location of a sliding arm. Optionally, sliders also may include a pair of buttons to permit incremental movement of the slider arm, an entry/display text box for typing and displaying an exact value, and a detent position for special values. A slider may be oriented vertically or horizontally. Slider values can be set by using the mouse to drag a slider across the scale until the desired value is reached. A visual indication of the relative setting is seen as the setting movement is made. In addition, some sliders may also be set by pointing at slider buttons located at one end of the scale and incrementally moving the arm through button clicks. Finally, keying a value in an associated text box may also set some sliders. A slider is used to make a setting when a continuous qualitative adjustment is acceptable, and it is advantageous to see the current value relative to all possible values.

Advantages/disadvantages. A slider displays a spatial representation of a relative setting, providing an excellent indication of where a value exists within a range of values. They are also visually distinctive and very recognizable. Sliders, however, are not as precise as an alphanumeric indication, unless a display field is provided. They also consume more screen space than other kinds of fields, and they can be more complex to operate.

Proper usage. Sliders are used to set an attribute when a limited range of continuous, relatively fine, possible settings exist. The choices must increase or decrease in some well-known, predictable, and easily understood way. Spatial representation of the attribute should enhance comprehension and interpretation and be sufficiently accurate.

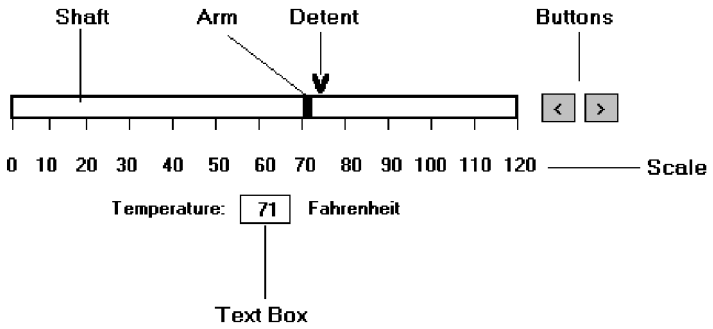


Figure 7.79: Slider.

General

- Use standard sliders whenever available.

Use of standard system sliders will speed learning.

Caption and Labels

- Caption:
  — Provide meaningful, clear, and consistent captions.
  — Display them using mixed-case letters.
  — Position the caption to the left of the box.
    - Alternately, it may be positioned left-justified above the slider.
- Labels:
  — Provide meaningful and descriptive labels to aid in interpreting the scale.

Caption. The slider caption must clearly reflect the quality being displayed. Use a static text or group box control to provide a field caption in mixed-case letters with each significant word capitalized. Use a static text field; the recommended position is to the box's left. Alternately, captions may be positioned above the slider, aligned with the left edge. Select a positioning consistent with other controls presented in the window.

Labels. Provide meaningful and descriptive labels to aid in interpreting the scale. A temperature slider will necessitate the inclusion of numeric temperature values. A volume slider may be labeled *low* and *high*. Create the labels using static text fields.

Scale

- Show a complete range of choices.
- Mark the low, intermediate, and high ends of the scale.
- Provide scale interval markings, where possible.
- Provide consistent increments.
- Permit the user to change the units of measure.
- If the precise value of a quantity represented is important, display the value set in an adjacent text box.

Provide a complete range of choices on the scale. Mark the low, intermediate, and high ends of the scale. For example, volumes may be indicated by low and high. Provide scale interval markings at consistent increments. Allow the user to change the units of measure, for example, changing a temperature from Fahrenheit to centigrade. If the precise value of a quantity represented is critical, display the set value in an adjacent entry/display text box control. This will also permit typed entry of the desired value.

Slider Arm

- If the user cannot change the value shown in a slider, do not display a slider arm.

Do not display a slider arm if the user cannot change the value shown in a slider. Fill in the shaft in a distinctive way to indicate the relative setting, as illustrated in the guideline for proportions.

Slider Buttons

- Provide slider buttons to permit movement by the smallest increment.
- If the user cannot change the value shown in a slider, do not display slider buttons.

Provide slider buttons to permit movement by the smallest increment. Movement is achieved by pointing and clicking. If the user cannot change the value shown in a slider, do not display slider buttons.

Detents

- Provide detents to set values that have special meaning.
- Permit the user to change the detent value.

For values that have special meaning, provide detents that can be changed by the user.

Proportions

- To indicate the proportion of a value being displayed, fill the slider shaft in some visually distinctive way.
  - — Fill horizontal sliders from left to right.
  - — Fill vertical sliders from bottom to top.

When the proportion of a value is also important, provide proportional indicators by filling in the slider shaft in a distinctive way. Fill it from left to right and bottom to top.

Tabs

- Description:
  - — A window containing tabbed dividers that create pages or sections.
  - — Navigation is permitted between the pages or sections.
- Purpose:
  - — To present information that can be logically organized into pages or sections within the same window.
- Advantages:
  - — Resemble their paper-based cousins.
  - — Visually distinctive.
  - — Effectively organize repetitive, related information.
- Disadvantages:
  - — Visually complex.
- Proper usage:
  - — To present discrete, logically structured, related, information.
  - — To present the setting choices that can be applied to an object.
  - — When a short tab label can meaningfully describe the tab's contents.
  - — When the order of information use varies.
  - — Do not use for sequential steps.

Description. A *tab* control is a window containing tabbed dividers that create pages or sections. Also referred to as a *notebook*, the tabs are analogous to dividers in a file cabinet or notebook. Navigation is permitted between the tabbed pages or sections. Microsoft Windows has a window organization scheme called a *workbook* (see Step 4) that is similar to the notebook control. Tabs from Microsoft Windows are illustrated in Figure 7.80.

Advantages/disadvantages. Tabs resemble their paper-based cousins, entities that are familiar to almost everyone. They are very meaningful electronic metaphors. Tabs are visually distinctive, and they permit effective organization of repetitive and related information. One drawback: They may result in a visually more complex screen.

Proper usage. Tabs can be used to present independent data that can be logically structured into discrete and meaningful pages or sections. They are most useful for presenting the choices that can be applied to an object, for example, a person and the person's descriptive data such as address, employment, family, and so forth. When the information on one tab is heavily dependent on information found on another tab, tabs are a poor choice because the user will have to keep flipping between tabs. Tabs are useful only if a brief tab label can identify their contents. Tabs are also useful when the order in which information is used varies.

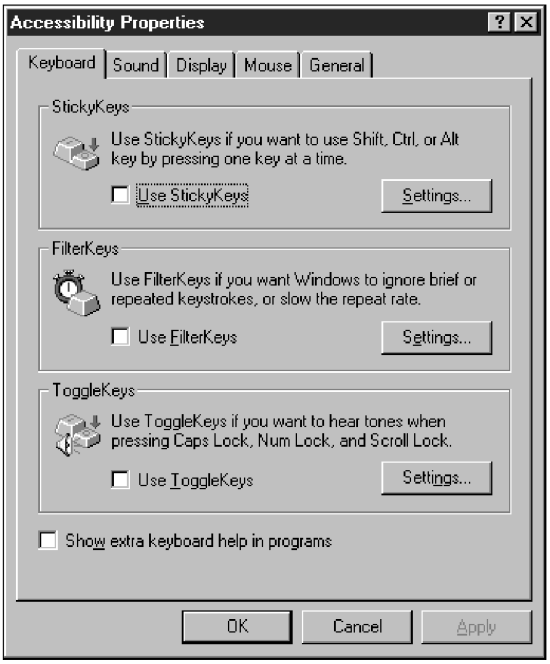


Figure 7.80: Tabs from Microsoft Windows.

Sections and Pages

- Place related information within a page or section.
- Order them meaningfully.
- Arrange pages so they appear to go deeper, left to right and top to bottom.
- Provide pages of equal size.

Place related information within a page or section. Order the pages in a meaningful way, based upon the window's content. Arrange the pages so they appear to go deeper, left to right and top to bottom. Provide pages of equal size.

Location, Size, and Labels

- Place the tabs at the top of the page or section.
- Assure the tabs look like real-world tabs.
- Provide fixed-width tabs for pages or sections of related information.
- Provide labels clearly descriptive of their function or destination.
  - — Use system fonts.
  - — Keep information brief and the same general length.
    - Nouns are usually better than verbs.
  - — Use mixed case, capitalizing each significant word.
  - — Assign a keyboard equivalent for keyboard access.
- Center the labels within the tabs.
- Restrict tabs to only one row.
- Arrange tabs so that they appear to go deeper, left to right and top to bottom.

Place tabs at the top of the page or section. This is the most customary location. Tabs may be capable of being located at all four points of the compass but these positions are not as common. Left and right tabs will provide label readability problems. Bottom tabs do not reflect the file cabinet analogy and may be confusing. Assure the tabs look like real-world tabs. A study found that people are more likely to click on tabs that look like paper-based tabs. Fixed-width tabs are preferred but variable-width tabs may be used if screen space constraints exist. For labels, use the standard system fonts in mixed case, capitalizing each significant word. Keep tab text labels brief and of the same general length. Nouns are usually better than verbs. If tab labels cannot be made clear because of tab size constraints, do not use tabs. Assign each tab a keyboard equivalent to facilitate keyboard access. Center the labels within the tabs and restrict them to only one row.

Avoid multiple rows of tab or scrolling a single row of tabs. This adds complexity to the interface and makes it harder for the user to read and access a particular tab. As an alternative, consider separating the tabbed pages into sets and using another control to move between sets, or use subordinate dialog boxes. Arrange tabs so that they appear to go deeper, left to right and top to bottom.

Command Buttons

- If they affect only a page or section, locate the buttons on the page or section.
- If they affect the entire tabbed control, locate the buttons outside the tabbed pages.

For command buttons that affect only the tabbed page being displayed, locate the buttons on that page. If they affect the entire tabbed control, position the buttons outside the pages but within the window holding the pages. Tab users often have trouble understanding that command buttons actions within a tab page only affect that page, but that command buttons outside all the pages affect the entire tab window. The users may not always be aware that when the window is closed, actions taken within a window are not "activated" until the *window's* OK button is pressed. Guidance may have to be provided to the users to ensure that all expected user actions are actually performed. Alternatives include providing instructions on the window or providing a confirming message box if the window is closed.

Date-Picker

- Description:
  - — A drop-down list box that displays a one-month calendar in the drop-down list box.
  - — The displayed month can be changed through pressing command buttons with left- and right-pointing arrows.
    - The left arrow moves backward through the monthly calendars.
    - The right arrow moves forward through the monthly calendars.
  - — A date for the list box can be selected from the drop-down calendar.
- Purpose:
  - — To select a date for inscribing in a drop-down list box.
- Advantages:
  - — Provides a representation of a physical calendar, a meaningful entity.
  - — The calendar listing is ordered in a predictable way.
  - — Visually distinctive.
- Disadvantages:
  - — Requires an extra step to display the calendar.
  - — When displayed, all month choices are not visible, requiring a form of scrolling to access the desired choice.
- Proper usage:
  - — To select and display a single date in close monthly proximity to the default month presented on the drop-down list box.

A *date-picker*, illustrated in Figure 7.81, is a drop-down list box that displays a one-month calendar in the drop-down. The displayed month can be changed by pressing command buttons with left- and right-pointing arrows. A relevant date to be

entered in the list box is selected from the calendar drop-down list box. Advantages and disadvantages are similar to drop-down list boxes. Its structure as a calendar is a meaningful representation for most users. Like drop-down list boxes, the date-picker requires exposing the list before the date can be chosen. If the date desired as not within the default month's calendar presented, the calendar must be scrolled to the proper month. If the date is far away in time, excessive scrolling may be required to obtain it. Therefore, it is most useful for dates close in time to the default month presented. A wide range of dates would best be collected through a text box.

Tree View

- Description:
  - — A special list box control that displays a set of objects as an indented outline, based on the objects' logical hierarchical relationship.
  - — Includes, optionally, buttons that expand and collapse the outline.
    - A button inscribed with a plus (+) expands the outline.
    - A button inscribed with a minus (-) collapses the outline.
  - — Elements that can optionally be displayed are
    - Icons.
    - Graphics, such as a check box.
    - Lines to illustrate hierarchical relationships.
- Purpose and proper usage:
  - — To display a set of objects as an indented outline to illustrate their logical hierarchical relationship.

A tree view control, as illustrated in Figure 7.82, is a special list box control that displays a set of objects as an indented outline, based on their logical hierarchical relationship. The control is used to display the relationship between a set of containers or other hierarchical elements, and, optionally, includes buttons to expand or collapse the hierarchy. Icons can be included with the text label for each item in the tree. Different icons can be displayed when the tree expands or collapses. A graphic, such as a check box, can be used to reflect state information about the item. The tree view control also supports an optional display of lines to illustrate the hierarchical relationship of the items in the list.

Figure 7.81: A date-picker control.

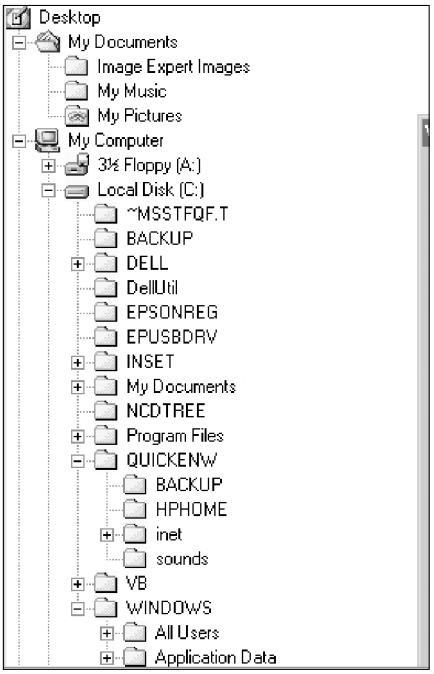


Figure 7.82: A tree view control.

Scroll Bars

- Description:
  - — An elongated rectangular container consisting of
    - A scroll area.
    - A slider box or elevator inside.
    - Arrows or anchors at either end.
  - — Available, if needed, in primary and secondary windows, some controls, and Web pages.
  - — May be oriented vertically or horizontally at the window or page edge.
- Purpose:
  - — To find and view information that takes more space than the allotted display space.
- Advantages:
  - — Permits viewing data of unlimited size.
- Disadvantages:
  - — Consumes screen space.
  - — Can be cumbersome to operate.
- Proper use:
  - — When more information is available than the window space for displaying it.
  - — Do not use to set values.

Figure 7.83: Scroll bar.

Description. A *scroll bar* is an elongated rectangular container consisting of a scroll area, a slider box, or elevator inside the scroll bar, and arrows or anchors at either end. They may be placed, if needed, in windows, in some controls, and in Web pages. They may be oriented vertically or horizontally at the right or bottom of a screen. Historically, scroll bars have been designed in a variety of styles; a typical one is illustrated in Figure 7.83.

Purpose. Scroll bars are used to find and view information that occupies more space than the allotted display space.

Advantages/disadvantages. While they permit viewing data of unlimited size, they do consume screen space and can be cumbersome to operate.

Proper usage. Use a scroll bar, or bars, when more information is available than the window space for displaying it. Do not use scroll bars to set values. If a value must be set or adjusted, use a slider or another control such as a spin box. Because scroll bars are designed for scrolling through information, using a scroll bar to set values may confuse the user about the purpose or interaction of the control.

Scroll Bar Design Guidelines

- General:
  — Provide a scroll bar when invisible information must be seen.
- Scroll area or container:
  — To indicate that scrolling is available, a scroll area or container should be provided.
    - Construct it of a filled-in bar displayed in a technique that visually contrasts with the window and screen body background.
- Scroll slider box or handle:
  — To indicate the location and amount of information being viewed, provide a slider box or handle.
    - Constructed of a movable and sizable open area of the scroll area, displayed in a technique that contrasts with the scroll area.
    - By its position, spatially indicate the relative location in the file of the information being viewed.
    - By its size, indicate, proportionately, the percentage of the available information in the file being viewed.

- Scroll directional arrows:
  - — To indicate the direction in which scrolling may be performed, directional arrows should be provided.
    - Construct them as arrows in small boxes, with backgrounds that contrast with the scroll area.
- Selection:
  - — When the slider box/handle has been selected, highlight it in some visually distinctive way.
- Location:
  - — Position a vertical (top-to-bottom) scroll bar to the right of the window.
  - — Position a horizontal (left-to-right) scroll bar at the bottom of the window.
- Size:
  - — A vertical scroll bar should be the height of the scrollable portion of the window body.
  - — A horizontal scroll bar should be at least one-half the width of the scrollable portion of the window body.
- Current state indication:
  - — Whenever the window's size or the position of the information changes, the scroll bar components must also change, reflecting the current state.
  - — Include scroll bars in all sizable windows.
    - If no information is currently available by scrolling in a particular direction, the relevant directional arrow should be subdued or grayed out.

General. A *scroll bar* provides a method to permit the displaying of information that may not always fit within a window on a screen. A scroll bar should only be included when scrolling may be necessary. In today's systems, scroll bars come in a variety of styles. Scroll bars consist of three elements: a scroll area or container, a slider box or handle that moves within a track made by the scroll area/container, and directional or scroll arrows.

Scroll area or container. The scroll area or, as it is sometimes called, the scroll container, is an elongated rectangular-shaped bar. Its presence indicates scrolling is available. It usually is constructed of a filled-in area displayed in a technique that visually contrasts with the window and screen body background. The chosen display technique should be of moderate intensity, neither too powerful nor too subtle. A powerful technique will be distracting; a subtle technique may be overlooked.

Slider box or handle. To indicate the location and amount of information being viewed, a slider box or, as it is sometimes called, a scroll handle, is included within the scroll area/container. It is constructed of a movable and sizable portion of the scroll area displayed in a technique that contrasts with the scroll area. It should indicate, by its position, the relative spatial location in the file of the information being viewed. It should indicate by its size, proportionately, the percentage of the available information in the file being viewed. Displaying, within a scrollable screen, the page number of page-organized material being viewed can further enhance the usability of the slider box or handle.

Directional or scroll arrows. To indicate the direction in which scrolling may be performed, directional or scroll arrows are also included. They are constructed of variously shaped arrows in small boxes with backgrounds that contrast with the scroll area/container. They are most often located at each end of the scroll bar, but some systems locate them adjacent to one another within the scroll area/container itself.

Placing directional arrows at opposite ends of the scroll bar is conceptually the clearest. The mouse pointer is moved in the same direction, away from the current position, when either the scroll arrow or scroll handle is manipulated. The distance that the directional arrows are separated by, however, causes increased effort when a window's contents must be adjusted by scrolling in opposite directions.

One platform solved the direction-switching problem by positioning the directional arrows adjacent to one another at one end of the scroll bar. While the forward-backward scrolling is made more efficient, the spatial correspondence between the beginning, middle, and end of the data is lost.

Another platform took another approach, placing the directional arrows at opposite ends of the slider box/handle to maintain the desirable spatial correspondence while at the same time minimizing their separation. Since during a continuous scrolling operation the directional arrows move as the slider box/handle moves, this platform automatically moves the mouse pointer to keep it aligned with the scroll arrow. This eliminates the need for the user to move the pointer during the continuous scrolling operation, but it requires that the user relinquish control of the mouse operation, and may be disorienting.

Using a scroll bar, the scrolling movement can be performed in several ways. The most common actions involve grabbing the slider box/handle and moving it in the desired direction, or selecting the proper directional arrow. Clicking a mouse button while selecting a directional arrow moves the contents of a window one line. Pressing the mouse button scrolls the window's contents continuously until the button is released. One platform provides another more efficient process. A region of the scroll area/container can also be selected, automatically moving the slider box/handle to that point and displaying the proper window contents.

Based upon early scrolling research (Bury et al., 1982), movement of the window data usually follows the window-up or telescope approach, whereby the window moves around over data that appears fixed in location. This causes the data in a window to move in the direction opposite the one indicated by the directional arrow or the direction of movement of the scroll container/handle. Scrolling using window systems, however, seems to be especially mistake-prone, users often assuming the arrows will move the data in the same direction as the directional arrow or scroll container/handle. In other words, it is sensed that the data moves under the window, not the window over the data (Billingsley, 1988). Why this happens is open to conjecture. Billingsley speculates that, because windows seem to move on screens, when data scrolls or moves in a window, people may conclude the data must be moving because the window remains still during the scrolling operation. Or, because of the close physical proximity of the directional arrows in scroll bars to the data, people may feel that the arrows are acting

on the data, not the window. The implication is that the scrolling procedure should be rethought and restudied. Some recent applications have devised scrolling methods through actually point at the window data.

Selection. When the slider box/handle has been selected, highlight it in some visually distinctive way. Most systems do provide some visual feedback of this kind.

Location. While, again, no universal agreement exists, the majority of systems locate the vertical (top-to-bottom) scroll bar to the right of the window and the horizontal (left-to-right) scroll bar at the bottom of the window.

Size. A vertical scroll bar should be the height of the scrollable portion of the window body. A horizontal scroll bar should be at least one-half the width of the scrollable portion of the window body.

Current state indication. Whenever the window's size or the position of information changes, the scroll bar components must also change, reflecting the current state of the scrolling process. Providing accurate information about the scrolling location facilitates user navigation and makes it easier to reposition the slider box/container. Include scroll bars in all sizable windows.

If scrolling cannot be performed in a particular direction, the relevant arrow box should be reduced in contrast or grayed out. If all the information in a window is displayed and no information is available for scrolling, both directional arrows should be reduced in contrast or grayed out.

Scroll Bar Usage Guidelines

- Scroll bar style:
  — Stick with standard, proven design styles.
- Directional preference:
  — Use vertical (top-to-bottom) scrolling.
  — Avoid horizontal (left-to-right) scrolling.

Style. The standard, well-known, proven design style used in graphical systems works best. A scroll bar is complex enough that presenting a new style to the user will focus attention away from the screen content as the user struggles to learn how to deal with the new style. This is a form of "senseless" learning and must be avoided.

Directional preferences. Where the choice exists, people prefer and deal better with vertical (top-to-bottom) scrolling rather than horizontal (left-to-right) scrolling. Avoid horizontal scrolling whenever possible.

The usability aspects of scrolling, and paging are thoroughly discussed in Step 3.

Media Controls

- For all playable files provide the following controls.
  — Play.
  — Pause/Resume.
  — Stop.
  — Rewind.
  — Fast Forward.
  — Volume.
  — Provide the content and size of all media objects.

Some media products provide their own controls. For others, controls may have to be designed. Always provide the above standard media controls. Instructions for downloading should include media type, file size, and a description of the subject matter.

Custom Controls

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

Many toolkits and interface builders provide the ability to create custom controls; implement them with caution. The user is currently presented with a multitude of controls whose usage and operation must be learned and remembered. The addition of custom controls adds to this learning and increases system complexity. If custom controls must be developed and implemented, make their look and behavior as different as possible from the standard controls. This will avoid confusion between the various controls.

MAXIM **Fewer is usually better.**

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.

Static Text Fields

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

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

Purpose/proper usage. Use static text fields to create and present read-only information, including all control captions. Also using static text fields clarify screen usage by providing prompting or instructional information. Other descriptive screen information can also be provided through static text fields. Examples are headings, subheadings, slider scales, progress indicator text, and so on. In Microsoft Windows, static text cannot be selected, so avoid using it for any text the user might want to copy to the clipboard.

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.

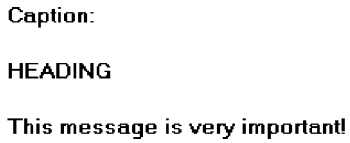


Figure 7.84: Static text field.

Captions. Always include a colon as part of the caption. The colon immediately identifies the element as a caption. In Microsoft Windows the colon is also used by screen review utilities. Include a keyboard equivalent (mnemonic) for all captions to provide keyboard access to its associated control. Captions may also provide a means of indicating that their associated controls are disabled. Follow all the rules for caption display presented throughout these guidelines.

Instructional or prompting information. Display in a unique and consistent font style for easy recognition and differentiation. Follow all other presented guidelines for prompting and instructional information. Guidelines for writing instructional or prompting information are discussed in Step 8.

Descriptive information. Other descriptive information includes headings, subheadings, slider scales, progress indicator text, and so forth. Also, follow all the rules for these other kinds of screen information presented throughout these guidelines.

Group Boxes

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

Description. A *group box* is a standardized 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 Figure 7.85.

Purpose/proper usage. The purpose of a group box is to visually relate the elements of a single control or a grouping of related controls. Group boxes should be used to provide a border around a radio button control, a grouping of related check boxes, or two or more functionally related controls.

Guidelines. Typically, use a noun or noun phrase for group box labels or headings. Keep the text brief, one or two words. Consider the group box content and relate the control captions inside the group box to the label or heading being created. Use headline-style capitalization, but do not include any ending colon. Refer to all the guidelines presented for control and section borders, in designing group boxes.

Figure 7.85: Group boxes.

Column Headings

- Description:
  - — Read-only textual information that serves as a heading above columns of text or numbers.
  - — Can be divided into two or more parts.
- Purpose:
  - — To identify a column of information contained in a table.
- Proper usage:
  - — To display a heading above a column of information contained in a table.
- Guidelines:
  - — Heading:
    - Provide a brief heading.
    - Can include text and a graphic image.
    - Capitalize the first letter of each significant word.
    - Do not include an ending colon (:).
  - — The width of the column should fit the average size of the column entries.
  - — Does not support keyboard access.

Description. A *column heading control*, also known as a *header control*, is used to display 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.

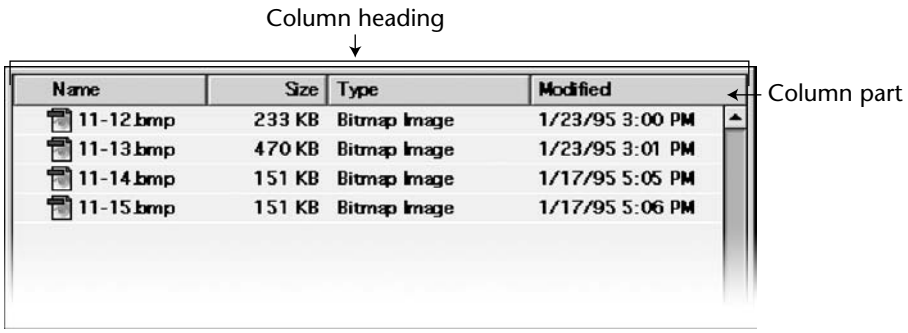


Figure 7.86: Column heading control.

Purpose/proper usage. To identify and display a heading above a column of information contained in a table.

Guidelines. Provide a brief heading. Headings can include text as well as a graphic image. Use the headline style of capitalization, without an ending colon. The width of each column should fit the average size of the column entries. Column heading controls do not support keyboard access.

ToolTips

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

Description. 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 Figure 7.87. A *Gloss* on a Web page is essentially a ToolTip.

Figure 7.87: ToolTip.

Purpose/proper usage. The purpose of a ToolTip is to simply provide thorough descriptive information about a control when its function must be quickly identified. It is also used to provide additional descriptive or status information about a screen element.

Advantages/disadvantages. A ToolTip provides an easy way to identify an otherwise unidentifiable control, or a control with a cryptic caption. It reduces possible screen clutter caused by control captions, enabling the control size to be reduced. A ToolTip, however, is not obvious and must be discovered. Its appearance when the pointer is positioned incorrectly, or is slowly passing over it, can be distracting to the screen viewer.

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.

Display the ToolTip on the screen after a short pause, ideally three-quarters of a second. This avoids its appearing briefly as the pointer is just being moved over a control or element that possesses a ToolTip. When used to provide descriptions of toolbar buttons, keep ToolTips brief, usually one or two words that identify the button's action. Use the headline style of capitalization with no ending punctuation. For other elements, descriptive or status information may be provided. In this case, use a short phrase, in sentence-style capitalization, which briefly describes the item or status. Position the ToolTip to the lower right of the pointer, but fully on the screen. Always adjust the location for a full fit. For text boxes, present the ToolTip centered under the control it relates to. Display it in the system's standard ToolTip colors so it will be immediately recognized as a ToolTip. Remove the ToolTip when the control is clicked or the pointer is moved away.

Don't substitute ToolTips for good design. Presented screen elements should always be designed for maximum comprehension. ToolTips are supplements.

Balloon Tips

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

Description. A *balloon tip*, illustrated in Figure 7.88, is a small pop-up window that contains information presented in a word balloon. Its components can include a title, body text, and informational, warning, or critical icons. (These icons are described in the Step 8.) Custom icons are not supported. Balloon tips appear adjacent to the item to which they apply, generally, above or to the left of the item. However, the system automatically adjusts their position so they remain on-screen. Although a tip can be posted at any time, only one, the last posted, will be visible at any time. Balloon tips used for the taskbar are presented for a specified time, within minimum and maximum limits.

Purpose. To provide additional descriptive or status information about a screen element.

Advantages/disadvantages. Balloon tips can provide useful reminder and status information to the user. Their sudden appearance can at some times be distracting, and perhaps aggravating, however, especially if overused in situations the user considers not very important. If overused the also lose their attention-getting value.

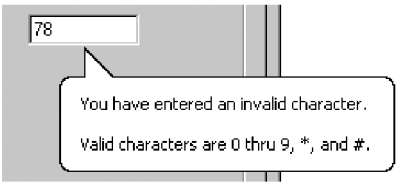


Figure 7.88: Balloon tip.

Proper usage. For noncritical reminder or notification information, special conditions, or status information that would otherwise require a message box. They are very useful for informing the user of unexpected system behaviors. Because of their brief nature and frequent out-of-the-way location, never rely on balloon tips to display critical information.

Balloon Tip Guidelines

- General:
  - — Use a notification tip to inform the user about state changes.
  - — Use a reminder tip for state changes that the user might not usually notice.
  - — Point the tip of the balloon to the item it references.
  - — Do not use them to replace ToolTips.
  - — Do not overuse balloon tips.
- Content:
  - — Restrict them to a length of 100 characters, including title and body text.
  - — Title text should
    - If the tip refers to an icon or other image representing a specific object, include
      - — The object's name, using its normal capitalization.
      - — The object's status, using sentence-style presentation without ending punctuation.
    - Be presented in bold.
  - — Body text should:
    - Include a description of the situation in one or two brief sentences.
    - Include a brief suggestion for correcting the situation.
    - Be presented using mixed-case in the sentence style.

General. Balloon tips can provide either notifications or reminders. The notification balloon is displayed and then times out. This tip style should be used to notify the user about state changes. The reminder balloon appears at regular intervals. The default interval is 60 minutes. Use the reminder balloon only for state changes that the user might not usually notice. Notification and reminder styles are supported for taskbar components. Other screen elements are only supported by the notification style. Tips are automatically removed when the user clicks it or clicks elsewhere.

Always point the tip of the balloon to the item it references. Balloon tips are not intended to replace standard ToolTips. ToolTips and balloon tips are mutually exclusive; if a ToolTip is currently displayed and a balloon tip is presented, the balloon tip will automatically cause the ToolTip to be removed. ToolTips will not appear until the balloon tip is dismissed. Finally, be careful not to overuse balloon tips. The user may ignore them if they appear too frequently.

Content. The notification balloon tip has a maximum length of 100 characters, including the title and body text. Title text automatically appears as bold text. Body text uses the text style and size of standard ToolTips. For the title text, if the balloon tip refers to an icon or other image representing a specific object, include the object's name using its normal capitalization and its status using sentence-style capitalization without ending punctuation. Otherwise, just display the status text. The body text should include a statement of the problem in one or two brief sentences, followed by a brief suggestion for correcting the problem. Use sentence-style capitalization and appropriate punctuation.

Progress Indicators

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

A *progress indicator* is a rectangular bar that fills as a process is being performed. The filled-in area indicates the percentage of a process that has been completed. A progress indicator, sometimes called a *progress bar*, is illustrated in Figure 7.89.

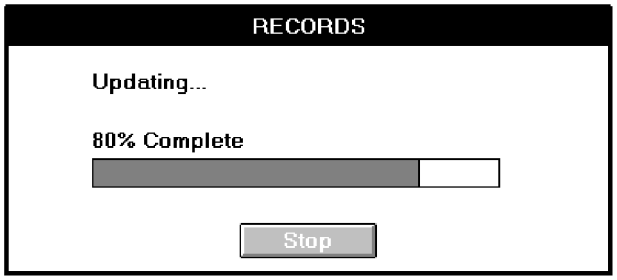


Figure 7.89: Progress indicator.

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.

Fill horizontally arrayed progress indicators from left to right; fill vertically arrayed progress indicators from bottom to top. Fill them with a color or a shade of gray. Create necessary descriptive text using a static text control. Position the text outside of the control. Progress indicators are also discussed in the Step 9.

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.
- 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.

Description. A *sample box* is a box illustrating what will show up on the screen based upon the parameter or parameters currently selected. A common example, shown in Figure 7.90, illustrates a font selected for display on a screen. Sample boxes may include text, graphics, or both.

Purpose. To provide a representation of actual screen content based upon the parameter or parameters selected so that the choice may actually be seen.

Guidelines. Include a brief label using mixed case in the headline style. Position the sample box immediately adjacent to the control or controls affecting its content.

Figure 7.90: Sample box.

SOME THOUGHTS ON POOR SCREEN DESIGN

Following is a collection of control design inadequacies whose most endearing qualities are that they *really, really aggravate* the author.

Why do interface designers insist on collecting state codes through drop-down list boxes, when all other name and address details are collected through text boxes?

I'm merrily typing away my "mentally programmed" name and address when I encounter the state field and come to a screeching halt. Do I reach for the mouse? Open the drop-down list box with the keyboard? I stumble, grumble, open the drop-down list box, find my home state, Arizona, and select it, thinking all the while how glad I am I live in a state whose name is near the top of the alphabetized list. My condolences go out to friends in places like Illinois and Texas who'll have to scroll to their state name. I return my hands to the keyboard and forge ahead to zip code. Why, I'm now thinking, do they need my zip code? They now have enough information to determine it — but that's another story.

Why am I often penalized by American systems for living in the United States?

Back to the name and address again; now I must indicate country. Can I type USA? Nope, back to a drop-down list box. Opening it, I stare at an alphabetized list of countries within which, I quickly realize, the United States of America lies well submerged. I look at the system's owner (a large hotel chain) and, because 95 percent of their hotels are in the USA, I estimate that the overwhelming majority of guests must be American. Have these people ever thought of another sequence like, perhaps, frequency of use? Or a combination of frequency and alphabetical with the several most frequent possibilities at the top? I guess not, I think, wondering if I should move to Australia to avoid this nonsense. "They're getting even with you for living in Arizona," you may be thinking, and this may be true. My friends in Illinois and Texas are double-whammed, however.

Why do they clear one critical piece of information when they re-present a screen?

With another hotel system, I encounter a "sorry, no room at this inn" for the days I'd like to be there. I select a different hotel and glance at the booking screen. The dates I previously entered are still there as well as the room type. I click "send." You didn't tell us whether your trip was for business or pleasure, their error message sternly advises. (Can't I also have pleasure on a business trip, I wonder?) I glance at the top of the screen. The radio buttons designating business or pleasure have magically been cleared, but nothing else on the reservation screen has changed. If I'm looking for another hotel on the same dates as just a moment ago, would I now be doing it for business instead of pleasure (or vice versa), I wonder. Perhaps some hotels are only for business, others only for pleasure — a worthwhile subject for usability research...

Why do they give me a small command button and surround it by acres of white space?

Are they afraid it might be too easy to find? Don't they really want it pressed? Are they testing my perceptual-motors skills? Is an anorexic button beautiful? Have they ever heard of Fitts' Law? Undoubtedly not!

Scrolling Tickers

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

Description. A *scrolling ticker* is a window that contains text scrolling horizontally.

Advantages/disadvantages. The biggest advantage of a scrolling ticker is its efficient use of screen space. Disadvantages include scrolling text being hard to read and time-consuming to interpret. Human memory being what it is, information scrolled out of sight is difficult to remember, and longer messages may not be understood. Scrolling screen elements are also visually distracting.

Guideline. The most prudent guideline at the moment is: Do not use scrolling tickers.

Selecting the Proper Controls

Providing the proper control, or mix of controls, is critical to a system's success. The proper control will enable a person to make needed selections, entries, and changes quickly, efficiently, and with fewer mistakes. Improper selection most often leads to the opposite result. In general, when selecting controls do the following:

Choose familiar controls. Never assume that all people are familiar with all types of controls. Unfamiliar controls will slow people down, and may even prevent people from using them at all. A study (Koyani et al., 2004) showed that some people, particularly those older, do not know how to use a drop-down list box. Be extremely cautious in using custom controls.

Consider the task. Will the choices to be made involve selecting one of many alternatives, or several items at once?

Reduce the number of "clicks." Design a data entry application so that people can stay with one entry method as long as possible. A study (Zavod and Fulop, 2001) comparing entry speeds using an entry field and drop-down lists found experienced users are much faster using entry fields. Requiring any user to continually shift between the keyboard and a mouse can substantially slow their entry speed.

Display as many control choices as possible. Available screen space is important in choosing the proper control. Can all, or most, of the control's options be dis-

played together on the screen, or do screen space constraints require using a control that hides most of the options, necessitating scrolling to see the entire list? A study by Couper et al. 2004, reached an interesting conclusion — what people see is what they select from! The options in this study were in a survey, so the alternatives to choose from required comparing and was essentially discretionary in nature. (Which of the following nutrients is most important in selecting a breakfast cereal? versus What is your home state? is hardly a discretionary choice.)

The study found that people tended to select from the entire list of options that they were first presented and saw. Rarely was an effort made to find additional options through scrolling. If eleven items were presented, the choice was from these eleven, if five, the choice was from these five (replicating what is known from paper survey research). Study design enabled the researchers to postulate why this happened:

- People spend more time thinking about the first presented options than the additional ones, so they pick them more often.
- People's thinking about the first options is clearer because they have fewer choices to consider.
- Initial options are harder to dislodge as option experience accumulates.

The conclusion: When options must be compared with one another, controls presenting all the options together will yield the best results. Controls initially presenting only a few of many options will not be the best choice.

This section will begin with a survey of several other research studies addressing control selection. Studies such as these, while few in number, appear in the research literature. The results of these studies have already been incorporated within the control usage guidelines just discussed. Next, the criteria that must be considered in control selection will be summarized. Finally, some selection guidelines will be presented.

Entry versus Selection — A Comparison

The first studies to be described are a series performed by IBM. These studies (Gould et al., 1988; Greene et al., 1988; Greene et al., 1992) looked at the advantages and disadvantages of using either entry fields or selection fields for data collection. Entry involved keying text; selection was performed by pointing at a choice through the keyboard using the cursor control keys (not a mouse). The information compared was of three kinds: dates, text, and data. The first conclusion follows:

Choosing a Type of Control

- For familiar, meaningful data, choose the technique that, in theory, requires the fewest number of keystrokes to complete.
- If the data is unfamiliar or prone to typing errors, choose a selection technique.

The studies found that if the data to be entered was familiar, the technique that required the minimum theoretical number of keystrokes to complete the task was the fastest. Theoretical keystrokes are the minimum number possible, excluding mis-keys, and erroneous cursor or selection movements. However, as the information became less familiar or became subject to spelling or typing errors, the minimum keystroke principle broke down. Selection techniques, and the reminders and structure they provide for unfamiliar items, hard-to-spell words, and items prone to typing errors becomes advantageous. The point at which the changeover occurs is not known. It would be influenced by the nature of the task and the nature of the user.

These studies point out the advantages of the techniques that permit both typed entry and selection to enter the data (spin box, drop-down/pop-up combination box, and attached combination box).

Aided versus Unaided Entry

- Provide aided entry whenever possible.
  - — Absorb any extra and unnecessary keystrokes.
  - — Provide an auditory signal that autocompletion has been performed.

The studies also compared unaided typed entry (the entire field had to be keyed) with aided entry (the system automatically and immediately completed the field when enough characters were keyed to make the desired data known). They found that aided entry, also known as autocompletion, was preferred over unaided entry methods, and it was also the fastest. Autocompletion was also preferable to, and faster than, many selection methods. Greene et al. (1992) found that, for keying of difficult to spell words, aided entry was much faster, and significantly reduced errors, when compared to unaided entry.

The result is that, when possible, autocompletion of text entry fields should be provided. Autocompletion will minimize the user's effort by reducing input time and keystrokes. It should also enhance the user's opinion of the system. If aided entry is provided, extra keystrokes must be absorbed by the system. The software will finish spelling a word much faster than a person's fingers are capable of stopping movement. Also, provide some kind of auditory signal that autocompletion has begun. A person may not be looking directly at the control when the completion is performed.

Comparison of GUI Controls

Tullis and Kodimer (1992) compared seven controls used for direct manipulation, selection, and data entry. The task was to reorder four items in a table (Filename, Number, Size, and Date). The controls studied were the following. Complete descriptions of control usage methods are summarized in Table 7.1.

- Direct Manipulation
  1. Drag and drop on
  2. Drag and drop between

- Selection
  3. Icons
  4. Radio buttons
  5. Menus (drop-down list boxes)
- Entry
  6. One entry area
  7. Four entry areas

The direct manipulation methods reflected the perceived strength of graphical systems, namely, manipulation of objects on the screen. The selection methods utilized indirect manipulation and illustrated the types of controls available in graphical screen design. The entry methods are a carryover from text-based screens, the only way the task could be accomplished for many years. Study participants were experienced Microsoft Windows users. No instructions were provided on how to carry out the item reorganization tasks. Users had to rely on their experience.

The two fastest methods were radio buttons and the one entry field. The methods most preferred by participants were radio buttons, drop-down list boxes, and one entry field. The direct manipulation methods fared rather poorly, ending midlist in the speed and preference rankings. The surprise, perhaps, was the good showing of an old control: the one entry field, or text box.

Tullis (1993) performed a follow-up to this study by asking a group of programmers to predict the study results (without, of course, being privy to its results). For both reordering speeds and subjective preferences, their predictions were way off the mark. They anticipated that the direct manipulation methods would be the fastest and most preferred. This, of course, was not at all the case. They predicted that radio buttons would be midway in the speed and preference ordering and that one entry field would be near the bottom. Again, they were quite mistaken. The correlation between their predictions and actual reordering speed was a dismal .07. They did slightly better on predicting preferences, the correlation being .31.

Based on these studies, Tullis concludes that control selection decisions made according to convention and intuition may not necessarily yield the best results. This conclusion might be modified to say, with a great deal of justification, that such decisions made using *common sense* may not even yield *good* results. Just because a control or process is new does not necessarily make it better. Just because the control has been around a long time does not necessarily make it poorer. Controls should be selected on the basis of the objectives they are to achieve, and they should be subjected to the same rigorous testing as all other parts of the system.

Another control comparison study was conducted by Johnsgard et al. (1995). They evaluated a variety of controls including check boxes, drop-down list boxes, drop-down combination boxes, text boxes, list boxes, radio buttons, sliders, and spin boxes. Speeds, errors, and preferences were obtained for the various controls under various conditions.

Table 7.1: Controls Evaluated by Tullis and Kodimer (1992)

DIRECT MANIPULATION
1. Drag and Drop On • The items are arrayed horizontally. An item is dragged to a new location above another item and released. The item in that position moves to the old location of the arriving item.
2. Drag and Drop Between • The items are arrayed horizontally. An item is dragged to a new location between two other items and released. The items are readjusted into new positions, including, when necessary, automatic wrap-around for items located at the end of the line.
SELECTION
3. Icons • The items are arrayed horizontally. Icons are positioned between each pair of items. Selecting an icon switches the positions of each adjacent item.
4. Radio Buttons • The items are presented in a matrix, item name along the left side, item position numbers across the top. Radio buttons in the matrix are selected to represent each item's position.
5. Menus (Drop-Down List Boxes) • Items are positioned horizontally. A drop-down listing is activated, and the item for that location selected.
ENTRY
6. One Entry Area • A single text entry field is provided. A one-character mnemonic (F,N,S,D) is provided for each choice. The mnemonics are keyed in the order in which the items are to be arrayed.
7. Four Entry Areas • Four text entry fields, labeled with the item names are arranged vertically. A number (1-4) is keyed into each field, indicating the manner in which the items are to be ordered.

Mutually Exclusive Choice Controls

For a small set of options (5), a medium set (12), and a large set (30), radio buttons were significantly faster than the other mutually exclusive controls. They were also the most accurate and most preferred by the study participants. This result is consistent with the results of the Tullis and Kodimer (1992) study. Among other findings: As control set sizes increased, control activation speeds significantly increased (took longer), and sets organized in a meaningful way were searched significantly faster than those in random orders.

The medium and large set sizes (12 and 30) are larger than generally recommended for radio buttons (8 or less). The results indicate that radio buttons may effectively be used for these larger quantities, if sufficient screen space exists for their presentation. Controls requiring scrolling to see all the choices, or requiring an action to display a listing of choices (drop-downs), seem to significantly impede selection speeds.

Nonexclusive Choice Controls

For a small set of options (5) with two selected choices, a medium set (12) with three selected choices, and a large set (30) with eight selected choices, check boxes were significantly faster than the other nonexclusive controls. Check boxes were also the most preferred by the study participants. Among other findings: Like radio buttons, as control set sizes increased, control activation speeds increased (took longer), and sets organized in a meaningful way were searched significantly faster than those in random order.

The medium and large set sizes (12 and 30) are also larger than generally recommended for check boxes (8 or less). The results also seem to indicate that check boxes may effectively be used for these larger quantities, if sufficient screen space exists for their presentation. Again, scrolling and retrieving lists slow one down.

Combination Selection and Entry Controls

Two controls were compared: a drop-down combination box and an array of radio buttons, including an "other" choice with an associated text entry field for keying the "other" value. The fastest, most accurate, and preferred: radio buttons with the text entry field.

Controls for Selecting a Value within a Range

Setting range values included indicating a time, a percentage, or the transmission frequency of a radio station. Controls evaluated were the spin button, text entry field, and the slider. The spin button was the most accurate, and the text entry field fastest and most preferred. The slider finished last in all three measurement categories.

The study's general conclusions are

- Making all options always visible will enhance performance.
- Requiring additional actions to make further options visible slows performance.
- For longer lists, scrolling tends to degrade performance more than the action associated with retrieving a hidden list.

As set size increases, performance times increase more for controls that require scrolling than for those that do not. For a large set size (30 options) scrolling slowed performance more than the action to retrieve a list.

Control Selection Criteria

Selection of the proper control, then, depends on several factors. The first is the structure and characteristics of the property or data itself. Other considerations include the nature of the task, the nature of the user, and the limitations of the display screen itself.

Property or data considerations reflect the characteristics of the data itself. Some kinds of controls are very restrictive in that they permit only specific kinds of information with specific qualities to be presented within them. Other kinds of controls may not be as restrictive concerning a data's qualities, but they are not well suited to the kind of data being used. Data considerations include the following:

- Is the property or data *mutually exclusive* or *nonexclusive*? Does entry/selection require single or multiple items?
- Is the property or data *discrete* or *continuous*? Discrete data can be meaningfully specified and categorized, while continuous data cannot.
- Is the property or data *limited* or *unlimited* in scope? If limited, how many items will the data normally not exceed?
- Is the property or data *fixed* or *variable* in list length? Are there always a fixed number of items, or will it vary?
- Is the property or data ordered in a *predictable* or *unpredictable* fashion? If predictable, will the user be able to anticipate the next, unseen, item?
- Can the property or data be *represented pictorially*? Will a picture or graphic be as meaningful as a textual description?

Task considerations reflect the nature of the job. Considerations include the following:

- How *often* is an item *entered* or *selected*?
- How *often* is an item *changed*?
- How *precisely* must the item be entered or selected?

User considerations reflect the characteristics of the user. Important considerations:

- How much *training* in control operation will be provided?
- How *meaningful* or *known* is the property or data to the user?
- How *easily remembered or learned* by the user is the property or data?
- How *frequently used* will the system be?
- Is the user an *experienced typist*?

Display considerations reflect the characteristics of the screen and hardware.

- How much *screen space* is available to display the various controls?

Choosing a Control Form

In light of the above research and considerations, and the known characteristics of the various controls, some guidance in control selection can be presented.

When to Permit Text Entry

- Permit text entry if any of the following questions can be answered *Yes*:
  - — Is the data unlimited in size and scope?
  - — Is the data familiar?
  - — Is the data *not* conducive to typing errors?
  - — Will typing be faster than choice selection?
  - — Is the user an experienced typist?

Permit text entry when any of the above conditions exist. The use of combination controls is almost always the best alternative, permitting the user to *choose* when to type and when to point and select.

What Kind of Control to Choose

Next are two tables providing some control recommendations based upon a control's known advantages, disadvantages, and proper usage characteristics. Table 7.2 is a simple decision chart for small listings, based upon Johnsgard et al. (1995). Table 7.3 is more thorough and is based upon the known characteristics of the controls described in this chapter.

The recommendations presented by Johnsgard et al. in these tables are based upon their research study. The names of some controls have been modified to reflect the control classification scheme found in this text. It would seem worth considering for controls containing a small number of choices. All controls in their study, except *setting a value within a range*, were limited to 30 options. For more than 30 choices, the use of radio buttons or check boxes still seems inappropriate at this time. More research is needed in this area.

Table 7.2: Best Controls for Certain Tasks and Screen Conditions

TASK	BEST CONTROL	IF SCREEN SPACE CONSTRAINTS EXIST
Mutually Exclusive	Radio Buttons	Drop-Down/Pop-Up List Box
Not Mutually Exclusive	Check Boxes	Multiple-Selection List Box
Select or Type a Value Text Entry Field	Radio Buttons with "Other"	Drop-DownComboBox
Setting a Value within a Range	Spin Button	TextBox

From Johnsgard et al. (1995).

Table 7.3: Suggested Uses for Graphical Controls

1. *IF*:	*USE:*
• Mutually exclusive alternatives. • Discrete data. • Best represented verbally. • Very limited in number (2 to 8).	
AND:	
• Typed entry is never necessary. • Content can never change. • Adequate screen space is available.	RadioButtons

(continued)

Table 7.3 *(continued)*

OR:	
• Typed entry is never necessary. • Content can never change. • Adequate screen space is not available.	Drop-Down/Pop-Up List Box
OR:	
• Typed entry may be necessary. • Content can change. • Adequate screen space is available.	Combo box
OR:	
• Typed entry may be necessary. • Content can change. • Adequate screen space is not available.	Drop-Down/Pop-Up Combo Box
2. IF:	*USE:*
• Mutually exclusive alternatives. • Discrete data. • Best represented verbally. • Potentially large in number (9 or more).	
AND:	
• Typed entry is never necessary. • Content can never change. • Adequate screen space is available.	Single-Selection List Box
OR:	
• Typed entry is never necessary. • Content can never change. • Adequate screen space is not available.	Drop-Down/Pop-Up List Box
OR:	
• Typed entry may be necessary. • Content can change. • Adequate screen space is available.	Combo Box
OR:	
• Typed entry may be necessary. • Content can change. • Adequate screen space is not available.	Drop-down/Pop-up Combo Box
3. IF:	*USE:*
• Mutually exclusive alternatives. • Discrete data. • Best represented graphically. • Content rarely changes. • Small or large number of items.	Palette

Table 7.3 *(continued)*

4. IF:	*USE:*
• Mutually exclusive alternatives. • Not frequently selected. • Content does not change. • Well-known, easily remembered data. • Predictable, consecutive data. • Typed entry sometimes desirable.	
AND:	
• Adequate screen space is not available.	Spin Box
OR:	
• Adequate screen space is available.	Combo Box
5. IF:	*USE:*
• Mutually exclusive alternatives. • Continuous data with a limited range of settings. • Value increases/decreases in a well-known, predictable way. • Spatial representation enhances comprehension.	Slider
6. IF:	*USE:*
• Nonexclusive alternatives. • Discrete data. • Best represented verbally. • Typed entry is never necessary. • Content can never change. • Adequate screen space is available.	
AND:	
• Very limited in number (2 to 8).	Check Boxes
OR:	
• Potentially large in number (9 or more).	Multiple-Selection List Box

Choosing between Buttons and Menus for Commands

Determining the proper way to present a command also depends on several factors. The following considerations are involved in choosing the correct command form:

- Whether or not the command is part of a *standard tool set*.
- The total *number* of commands in the application.
- The *complexity* of the commands.
- The *frequency* with which commands are used.
- Whether or not the command is used in association *with another control*.

Guidelines for choosing the proper command form are presented in Table 7.4.

Table 7.4: Choosing a Command Form

IF THE COMMANDS:	USE:
Are standard commands provided by a tool set.	Commands provided by the tool set.
Total seven or more, and can be arranged hierarchically into groups.	Menu bar and pull-downs
Total six or fewer, are selected frequently, and affect an entire window.	Buttons in a window
Total seven or more, are selected frequently, and affect an entire window.	Buttons in a toolbar
Are used with other controls, or are complicated commands and need to be simplified.	Buttons in a dialog box
Are sometimes used frequently and are sometimes used infrequently.	Buttons in a dialog box
Are frequently accessed and have only two conditions.	Toggled menu item

Examples

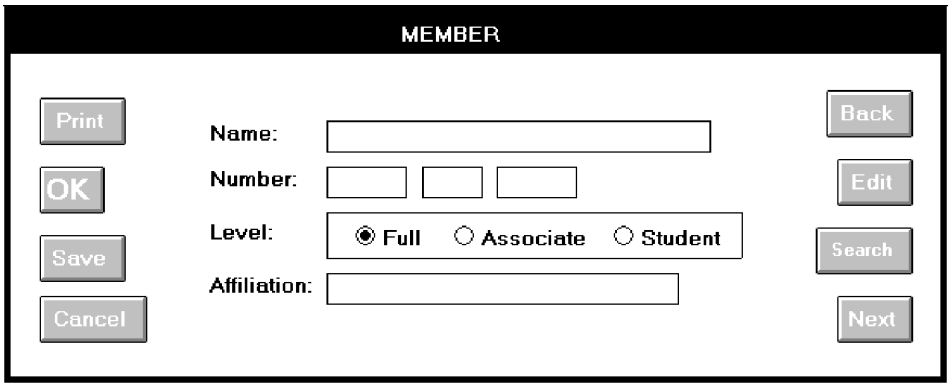
Improper and proper usage of several controls are illustrated and discussed.

Example 1

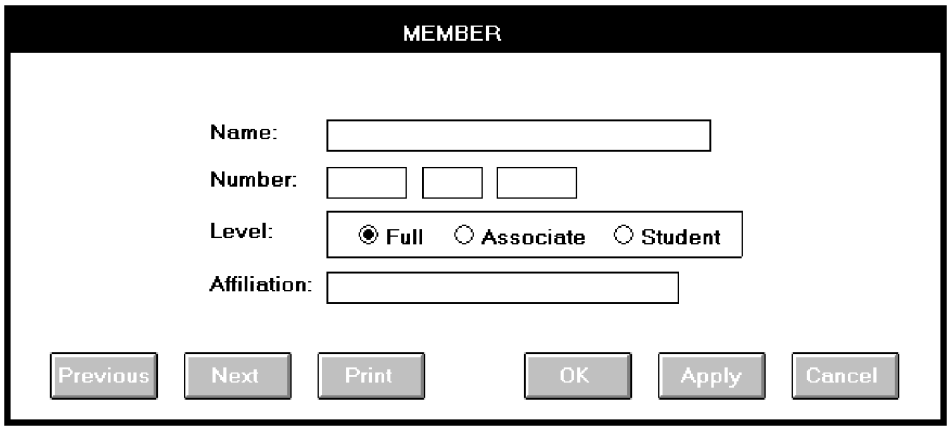
This is an instance of improper and proper presentation of command buttons.

Screen 1.1

Here the design and display of buttons is poor. Problems include: (1) The buttons are split between the left and right side of the screen, causing a wide separation. Positioning buttons to the left, from a screen usage and flow standpoint, is illogical. (2) Differences in sizes exist between buttons. OK, a very frequently used button, is the smallest, slowing down selection speed if a pointer is used. (3) Different size labels also exist (OK and Search). (4) There appears to be redundancy in button use and purpose. How does OK differ from Save? What does Edit do? (5) From an organization standpoint, standard and application buttons appear to be intermixed. (6) The Back and Next actions are widely separated, making fast reversal of actions more difficult.



Screen 1.1



Screen 1.2

Screen 1.2

This shows a much better button design and presentation. Enhancements include: (1) The buttons are located at the bottom of the screen, in a position following the screen usage flow. (2) Button size is standardized, presenting generally larger targets. (3) Button label size is standardized. (4) The seemingly redundant buttons are eliminated. (5) Function-specific buttons are grouped separately from standard buttons, and button groupings are created through a larger spacing between Print and OK. (6) Back, now called Previous, and Next are positioned together for fast paging reversal.

Example 2

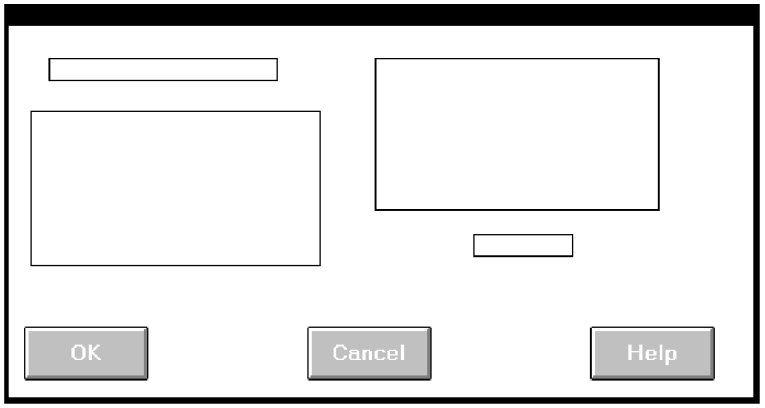
Here inconsistent locations of command buttons are reviewed.

Screens 2.1 to 2.4

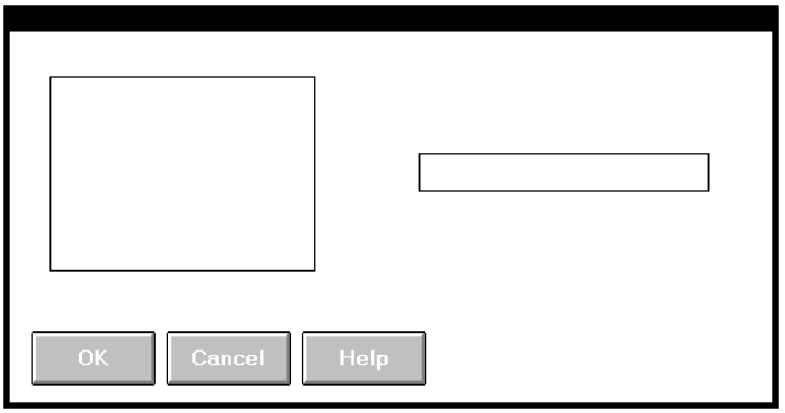
These are the button locations found on four windows within a graphical system. Positions include spread out across the window's bottom (Screen 2.1), left-justified at the bottom (Screen 2.2), centered along the right side (Screen 2.3), and top-justified along the right side (Screen 2.4). Memorization and prediction of button location will be very difficult, slowing down the experienced user.

Screen 2.5

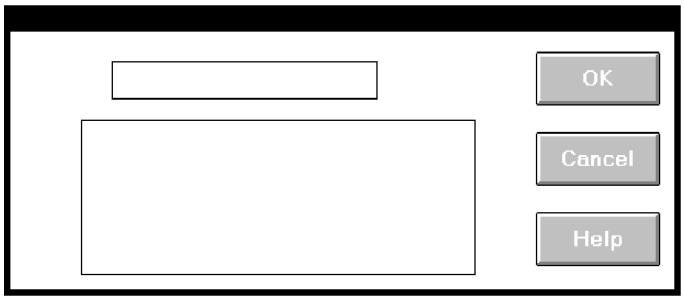
Proper positioning would have found *all* the command buttons consistently positioned, as at the bottom center, illustrated in Screen 2.5.



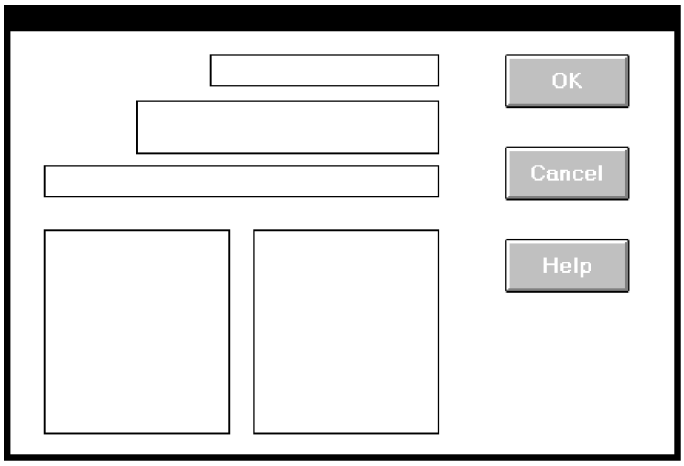
Screen 2.1



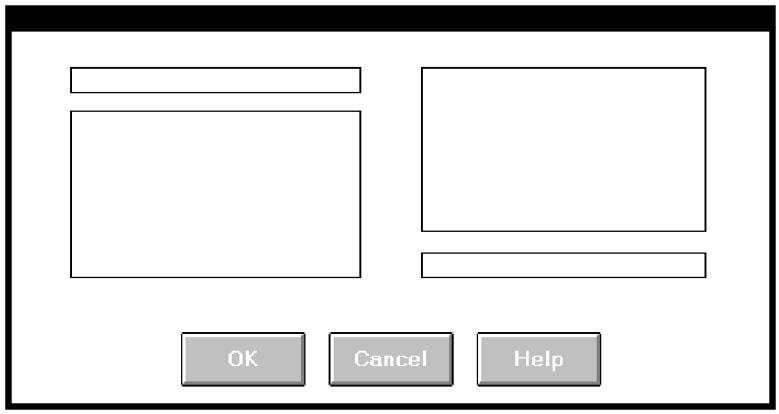
Screen 2.2



Screen 2.3



Screen 2.4



Screen 2.5

Example 3

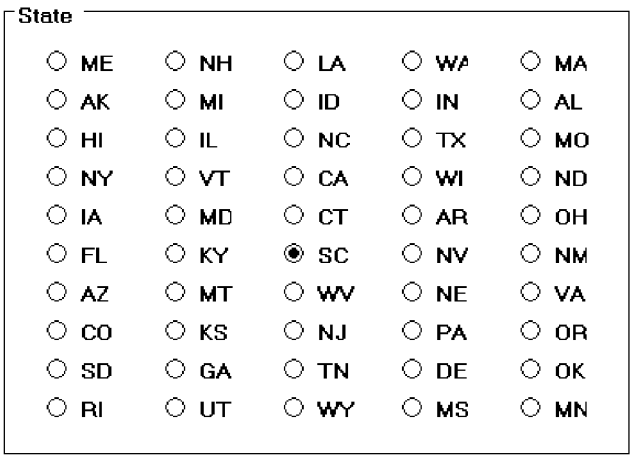
This is an example of improper and proper use of a control.

Control 3.1

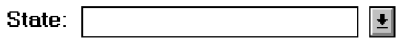
The names of states must be selected using radio buttons. Problems include: (1) The large number of choices presented makes scanning very difficult. (2) Are all the state abbreviations familiar to you, and all users? (3) The organization of states must have been established by a lottery. The name of the state I want is Mississippi. How do I find it in the array?

Control 3.2

This shows a much better alternative, a drop-down/pop-up combination box. If the state name is known, it can be typed in the field. Ideally, typing the state code, if known, will also be acceptable. If the name of a particular state is unknown, or if it's spelling is unclear, the drop-down/pop-up can be retrieved and the state name selected from the list presented. Ideally, also, a misspelled keyed state name will still be correctly identified by the system and displayed properly.



Control 3.1



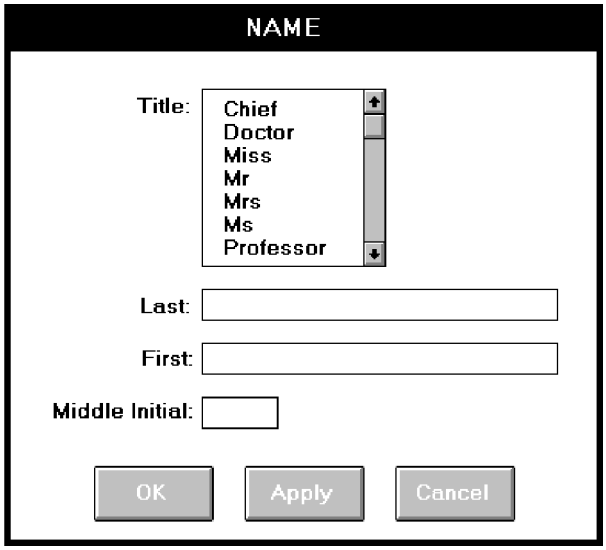
Control 3.2

Example 4

Here is another example of improper and proper use of a control.

Screen 4.1

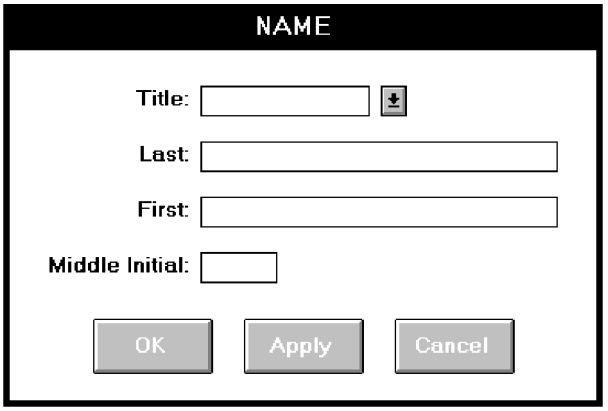
A listing of names is being collected. A courtesy title is selected through list box; last name, first name, and middle initial are typed. The problem: The task is heavily keyboard intensive. To select a title requires shifting to an alternative device control, such as a mouse, or tabbing through the list box listing to find the proper title. This slows down the keying process and may be awkward. The list box also consumes a great deal of space on the screen.



Screen 4.1

Screen 4.2

A solution: Collect the courtesy title using a pop-up/drop-down combination box. Familiar titles may be quickly typed, along with the remainder of the name data. Rare or unusual titles may be identified by selecting, displaying, and searching the listing of all alternatives. The title may then be entered in the field by selecting from the list or keying it into the field.



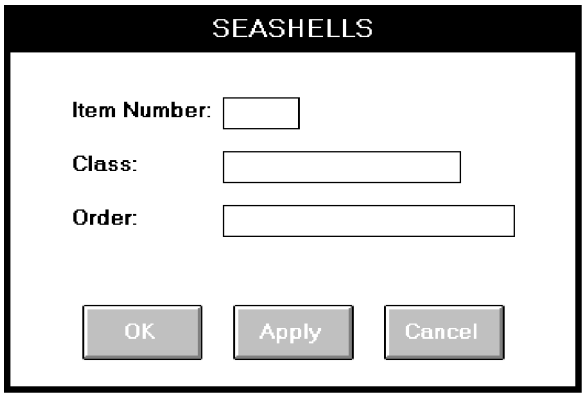
Screen 4.2

Example 5

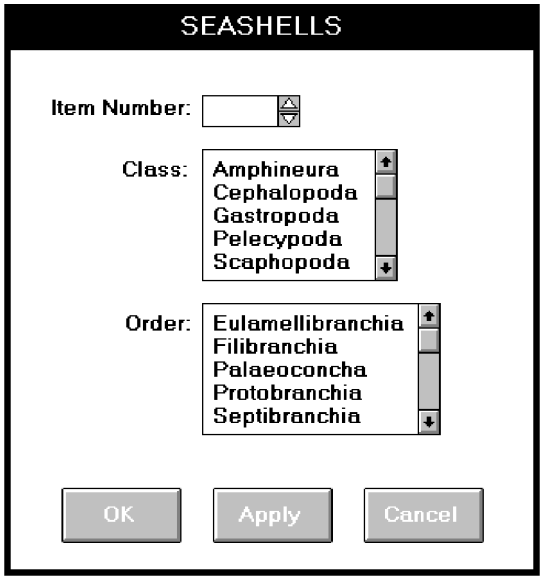
Again, here is an example of improper and proper use of controls.

Screen 5.1

A collection of seashells is being cataloged by class and order. Text boxes are provided for the task. The catalog process includes typing words such as "Cephalopoda" and "Eulamellibranchia." The process is slow and conducive to spelling errors.



Screen 5.1



Screen 5.2

Screen 5.2

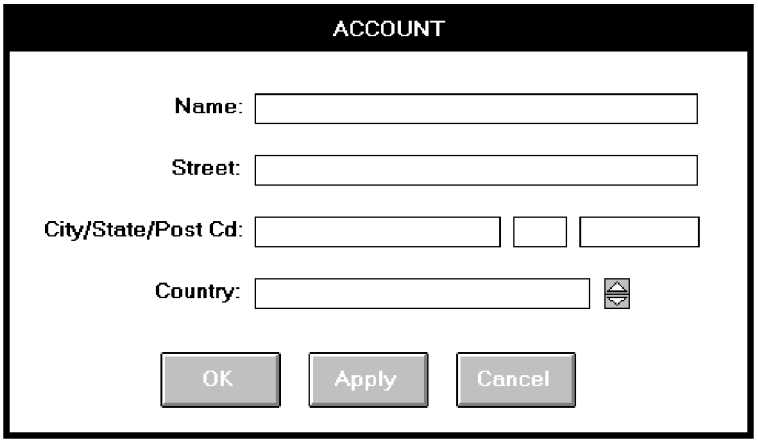
A solution: Present Class and Order in list boxes from which the proper varieties are selected. This will speed up the cataloguing process and eliminate the possibility of spelling errors. To make the entire procedure a selection task, also make Item Number a selectable and incrementable spin box.

Example 6

Again, here is an example of improper and proper use of a control.

Screen 6.1

An international corporation is setting up a worldwide account database. Names from dozens of different countries are added each day. Country is collected though a spin box. Is this proper usage for a spin box?



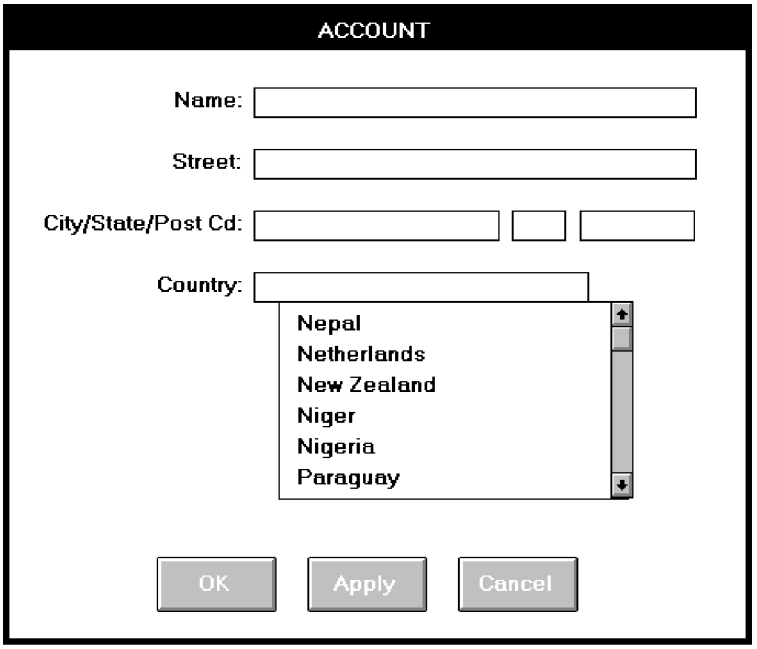
Screen 6.1

Screens 6.2 and 6.3

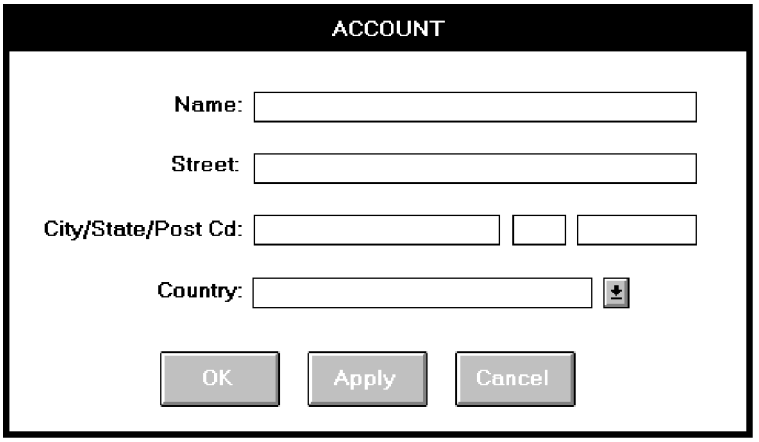
With a spin box, the next unseen choice must be capable of being anticipated. If not, tedious clicking and searching to find the correct choice might have to be performed. (What country follows Greece in the worldwide alphabetical listing of countries today? Guatemala, at least at this writing.) The data in spin boxes should be stable, not often changing. This quality does not accurately reflect the state of countries in the world today.

The best choice would really depend on the variability of the information being collected. If the account information being collected tended to be quite variable in flow, that is, successive account entries were usually from different countries, a better choice would be a combo box (Screen 6.2). Well-known country names can be typed and less-well-known ones found in the listing. Because of the dynamic nature of country names, frequent reference to the list can be expected. Permanently displaying the list avoids the step of retrieving it when needed. The attached listing also permits scanning several names at one time, alleviating the predictability problem. Names can also be easily added or changed as needed. The combo box is also at the bottom of the window where it tends to be out of the way.

If successive account entries tended to be from the same country, that is, the information is batched, a pop-up/drop-down combination would be more appropriate (Screen 6.3). The box can remain closed through successive same country entries and only needs to be opened occasionally.



Screen 6.2



Screen 6.3

Step 7 Exercise

An exercise for Step 7 can be found on this book's companion Web site, www.wiley.com/college/galitz.

FUTURE VISION BIE

By K B Hemanth Raj

Visit : https://hemanthrajhemu.github.io

Quick Links for Faster Access.

CSE 8th Semester - https://hemanthrajhemu.github.io/CSE8/

ISE 8th Semester - https://hemanthrajhemu.github.io/ISE8/

ECE 8th Semester - https://hemanthrajhemu.github.io/ECE8/

8th Semester CSE - TEXTBOOK - NOTES - QP - SCANNER & MORE

17CS81 IOT - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS81/

17CS82 BDA - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS82/

17CS832 UID - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS832/

17CS834 SMS - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS834/

8th Semester Computer Science & Engineering (CSE)

8th Semester CSE Text Books: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Text-Book.html

8th Semester CSE Notes: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Notes.html

8th Semester CSE Question Paper: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Question-Paper.html

8th Semester CSE Scanner: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Scanner.html

8th Semester CSE Question Bank: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Question-Bank.html

8th Semester CSE Answer Script: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Answer-Script.html

Contribution Link:

https://hemanthrajhemu.github.io/Contribution/

Stay Connected... get Updated... ask your queries...

Join Telegram to get Instant Updates:
https://telegram.me/joinchat/AAAAAFTtp8kuvCHALxuMaQ

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/futurevisionbie/