

MODULE 4

WINDOWS

To understand the purpose and usage of different kinds of windows for respective tasks.

- To identify proper input devices for implementing to the user based on their characteristics
- To identify proper screen based controls for implementing to the user based on their characteristics

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.

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

- 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:** 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.
- Presentation of Multiple Kinds of Information:** Variable information needed to complete a task can be displayed simultaneously in adjacent windows. For example in one window billing can be done and in one window stock

Maintenance can be done at the same time using windows. Significant windows could remain displayed so that details may be modified as needed prior

- Sequential Presentation of Levels or Kinds of Information:** Steps to accomplish a task can be sequentially presented through windows. 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.
- Access to Different Sources of Information:** Independent sources of information may have to be accessed at the same time. Independent sources of information may have to be accessed at the same time
- 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:** While waiting for a long, complex procedure to finish, another can be performed. Tasks of higher priority can interrupt less important ones and then the interrupted tasks can be preceded.
- Reminding:** It can be used to provide remainder through messages or popup or menus.
- Monitoring:** Data in one window can be modified and its effect on data in another window can be studied.
- Multiple Representations of the Same Task:** the same task can be represented in two different ways in two windows. For example a report can be given as table in one window and as a chart in another window.

Constraints in Window System Design

- Historically, system developers have been much more interested in solving hardware problems than in user considerations.
- This lack of guidelines makes it difficult to develop acceptable and agreeable window standards.
- 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

- 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.
- Poor screen resolution and graphics capability may also deter effective use of windows by not permitting sharp and realistic drawings and shapes

Human Limitations

- 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.
- 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

Components of a Window

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.

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.

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 16 X 16 version of the icon of the object being viewed.

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—is 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, since the window can no longer be increased in size.
- 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.

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.

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's 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*.

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

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*

Toolbar

- Toolbars are permanently displayed 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*.

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.

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.

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 also be referred to as the *client area*.

Table 5.1 Microsoft Windows Components

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

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.

Tiled Windows

- Tiled windows derive their name from 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.



□ Advantages:

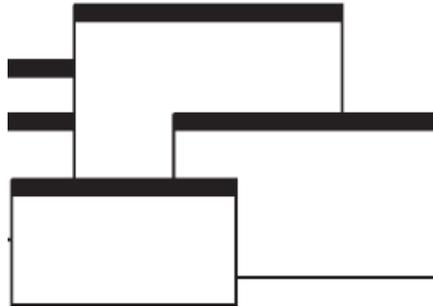
- The system usually allocates and positions windows for the user, eliminating the necessity to make positioning decisions.
- Open windows are always visible, eliminating the possibility of them being lost and forgotten.
- Every window is always completely visible, eliminating the possibility of information being hidden.
- They are perceived as fewer complexes 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.

□ Disadvantages

- 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.

Overlapping Windows

- Overlapping windows 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.

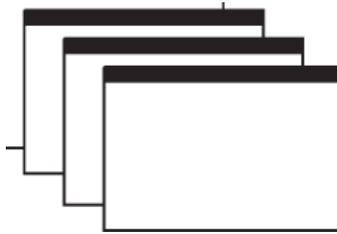


- Advantages:
 - Visually, their look is three-dimensional, resembling the desktop that is familiar to the user.
 - Greater control allows the user to organize the windows to meet his or her needs.
 - Windows can maintain larger sizes.
 - Windows can maintain consistent sizes.
 - Windows can maintain 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
 - 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



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

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.

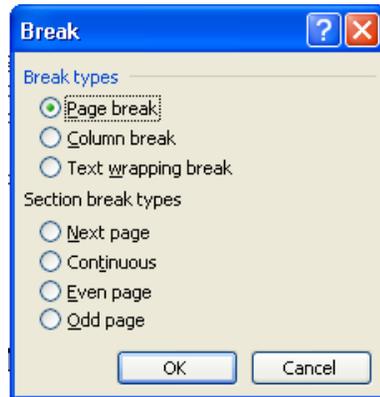
Types of Windows

Primary Window



- Proper usage:
 - Should 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.
 - For example, 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.
- 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

Secondary Windows



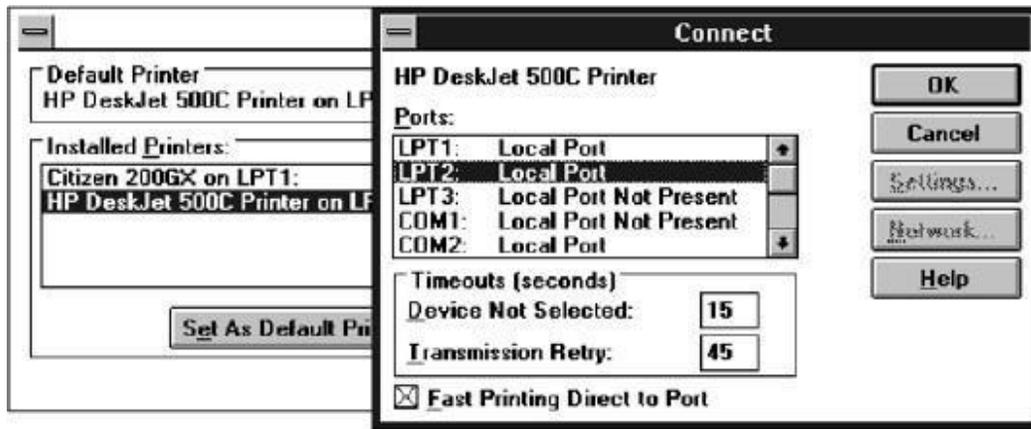
- Proper usage:
 - For performing subordinate, supplemental, or ancillary actions that are:
 - Extended or more complex in nature.
 - Related to objects in the primary window.
 - For presenting frequently or occasionally used window components.
- Important guidelines:
 - Should typically not appear as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

Modal and Modeless

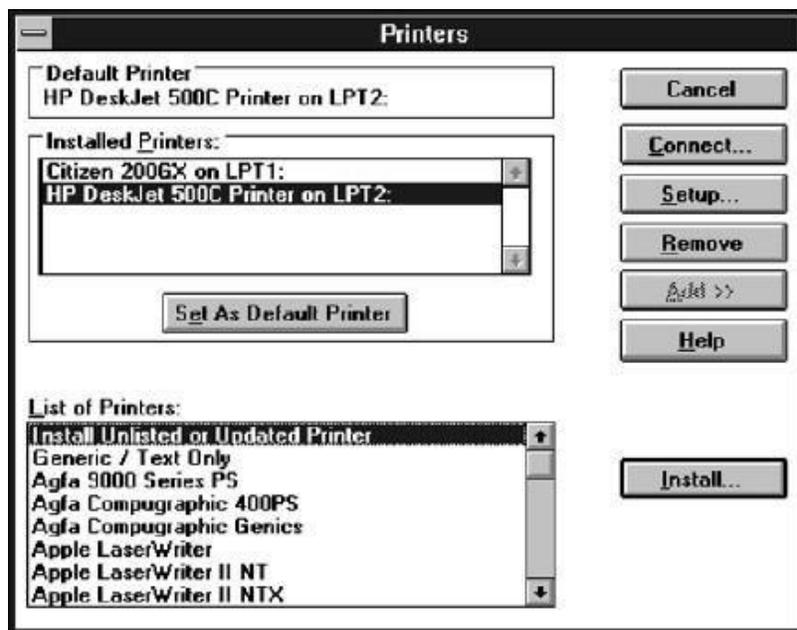
- Modal:
 - Use when interaction with any other window must not be permitted.
 - Use for:
 - Presenting information.
 - For example, messages (sometimes called a message box).
 - Receiving user input.
 - For example, data or information (sometimes called a prompt box).
 - Asking questions.
 - For example,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

Cascading and Unfolding

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

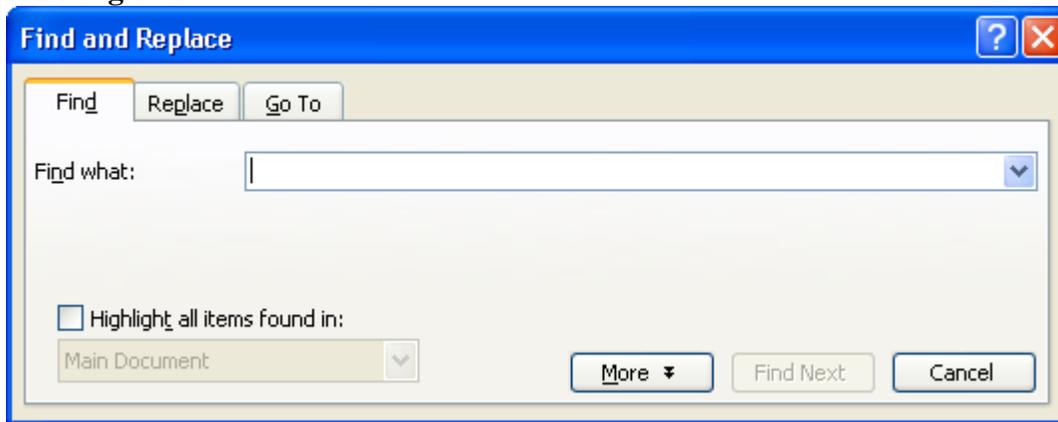


Cascaded Window



Unfolded Window

Dialog Boxes

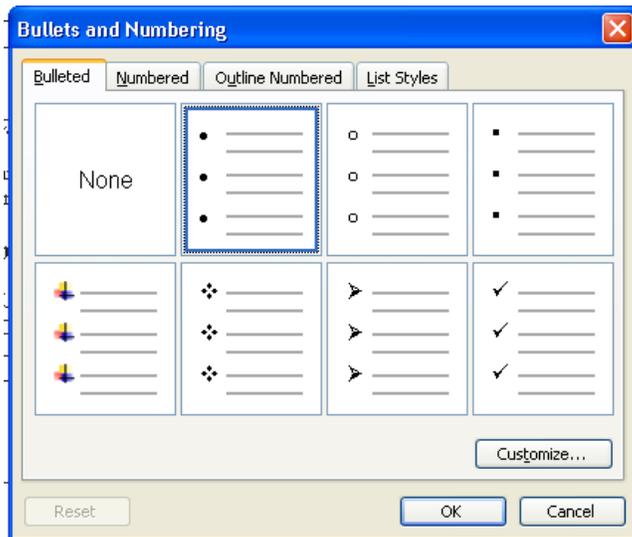


- Use for presenting brief messages.
- Use for requesting specific, transient actions.
- Use for performing actions that:
 - Take a short time to complete.
 - Are not frequently changed.
- Command buttons to include:
 - OK.
 - Cancel.
 - Others as necessary.

Property Sheets and Property Inspectors

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

Property Sheets



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

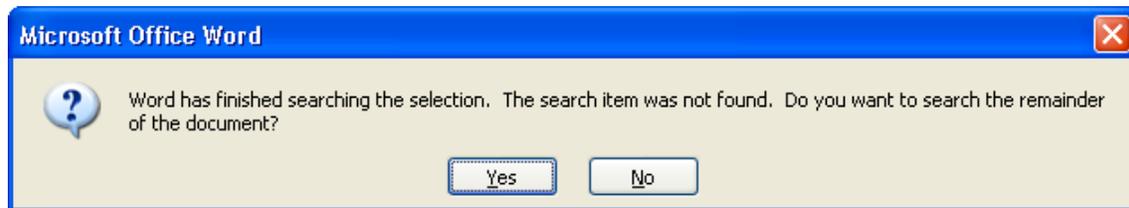
- The recommended sizes for property sheets are:
 - 252 DLU's wide x 218 DLU's high
 - 227 DLU's wide x 215 DLU's high
 - 212 DLU's wide x 188 DLU's high
- Command buttons to include:
 - OK.
 - Cancel.
 - Apply.
 - Reset.
 - Others as necessary.
- For single property sheets, place the commands on the sheet.
- For tabbed property pages, place the commands outside the tabbed pages.

Property Inspectors



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

Message Boxes



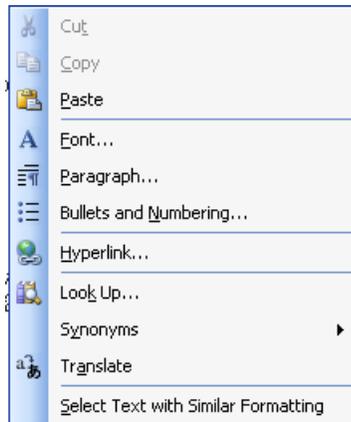
- Use for displaying a message about a particular situation or condition.
- Command buttons to include:
 - OK.
 - Cancel.
 - Help.
 - Yes and No.
 - Stop.
 - Buttons to correct the action that caused the message box to be displayed.
- Enable the title bar close box only if the message includes a cancel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

Palette Windows



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

Pop-up Windows



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

Window Management

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

Single-Document Interface

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

- Most common usage.
- Window manipulation is easier and less confusing.
- Data-centered approach.
- Disadvantage:
 - Information is displayed or edited in separate windows.

Multiple-Document Interface

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

Workbooks

- Description:

- A window or task management technique that consists of a set of views organized like a tabbed notebook.
- It is based upon the metaphor of a book or notebook.
- Views of objects are presented as sections within the workbook's primary windows; child windows do not exist.
- Each section represents a view of data.
- Tabs can be included and used to navigate between sections.
- Otherwise, its characteristics and behavior are similar to those of the multiple document interface with all child windows maximized.
- Proper usage:
 - To manage a set of views of an object.
 - To optimize quick navigation of multiple views.
 - For content where the order of the sections is significant.
- Advantages:
 - Provides a grouping and focus for a set of activities within the larger environment of the desktop.
 - Conserves screen real estate.
 - Provides the greater simplicity of the single-document window interface.
 - Provides greater simplicity by eliminating child window management.
 - Preserves some management capabilities of the multiple-document interface.
- Disadvantage:
 - Cannot present simultaneous views.

Projects

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

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

Organizing Window Functions

Window Organization

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

Number of Windows

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

Window Operations

Active Window

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

General Guidelines

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

Opening a Window

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

Sizing Windows

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

- They facilitate learning: Data relationships and groupings are more obvious.
- Less window manipulation requirements exist.
- Breadth is preferred to depth (based on menu research).
- More efficient data validation and data correction can be performed.
- Disadvantages include:
 - Longer pointer movements are required.
 - Windows are more crowded.
 - More visual scanning is required.
 - Other windows more easily obscure parts of the window.
 - It is not as easy to hide inappropriate data.

Window Placement

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

Window Separation

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

Moving a Window

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

Resizing a Window

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

Other Operations

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

Window Shuffling

Window shuffling must be easy to accomplish.

Keyboard Control/Mouse less Operation

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

Closing a Window

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

Select the Proper Device-Based Controls

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

Characteristics of Device-Based Controls

Several specific tasks are performed using graphical systems.

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

Direct and Indirect Devices

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

Trackball

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

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

Joystick

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

Graphic Tablet

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

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

Touch Screen

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

Light Pen

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

- Stands up well in high-use environments.
- More accurate than finger touching.
- Disadvantages:
 - Hand may obscure part of screen.
 - Requires picking it up to use.
 - Requires moving the hand far from the keyboard to use.
 - Very fatiguing to use for extended period of time.

Voice

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

Mouse

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

Keyboard

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

Selecting the Proper Device-Based Controls

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

- 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.

Keyboard versus 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,

Control Research

- 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 due to 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. Using a mouse, graphic tablet, and trackball for this task, as well as pointing, was studied by MacKenzie, Sellen, and Buxton (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.

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 other 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.

Choose the Proper Screen-Based Controls

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

Work the way it looks.

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