

## UNIT III

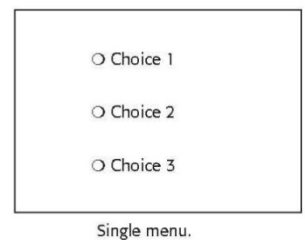
**System menus: Structures of Menus, Functions of Menus, Content of Menus, Kinds of Graphical menus**  
**Windows: Window characteristics, Components of a window, Window presentation styles, Types of windows, Window management**

### Structures of Menus

- Menus vary in form from very simple to very complex. They may range from small dialog boxes requesting the user to choose between one of two alternatives, to hierarchical tree schemes with many branches and level of depth.
- A menu's structure defines the amount of control given to the user in performing a task. The most common structures are the following.

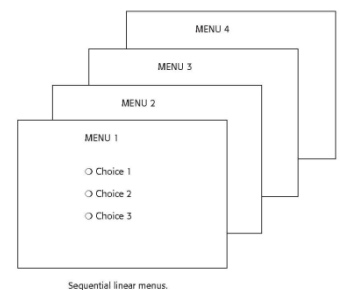
#### Single Menu

- In this simplest form of menu, a single screen or window is presented to seek the user's input or request an action to be performed, as illustrated in Figure.
- Single menus conceptually require choices from this single menu only, and no other menus will follow necessitating additional user choices.
- The user need only consider the immediate consequences of the item being chosen and need not be concerned with any other additional system menus. A single menu may be iterative if it requires data to be entered into it and this data input is subject to a validity check that fails. The menu will then be represented to the user with a message requesting reentry of valid data.



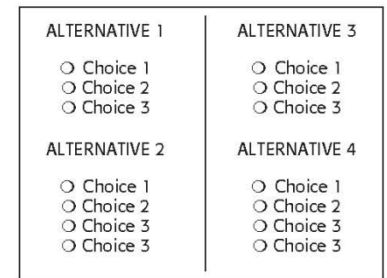
#### Sequential Linear Menus

- Sequential linear menus are presented on a series of screens possessing only one path. The menu screens are presented in a preset order, and, generally, their objective is for specifying parameters or for entering data.
- The length of the path may be short, or long, depending upon the nature of the information being collected.
- Sequential path menus have several shortcomings. A long sequence may become tedious as menu after menu is presented. The user may not remember an answer to a previous question, a question important to the currently presented choices.
- The user may also want to return to a previous menu to change an answer or look at an answer, an awkward process that must be allowed.
- Finally, the user may, conceptually, want to complete the menus in a different order than which they are being presented.



### Simultaneous Menus

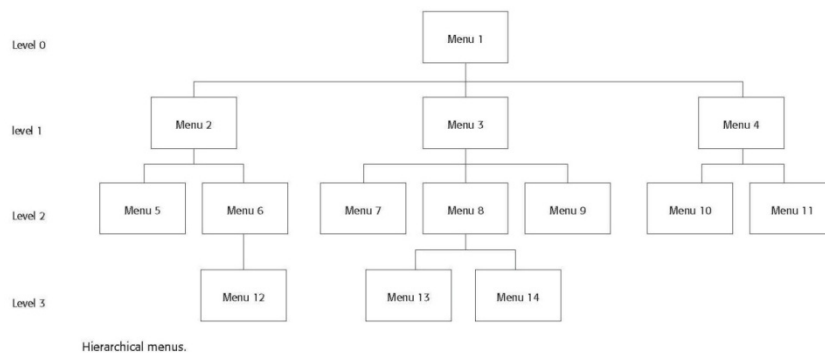
- Instead of being presented on separate screens, all menu options are available simultaneously, as illustrated in Figure.
- The menu may be completed in the order desired by the user, choices being skipped and returned to later. All alternatives are visible for reminding of choices, comparing choices, and changing answers. The tedium associated with a long series of sequential menus is greatly reduced.
- Problems with simultaneous menus are that for large collections of menu alternatives screen clutter can easily occur, and screen paging or scrolling may still be necessary to view all the choices.
- Presenting many menu dependencies and relationships on a screen, especially if poorly indicated, can also be very confusing for a novice user.



Simultaneous menus.

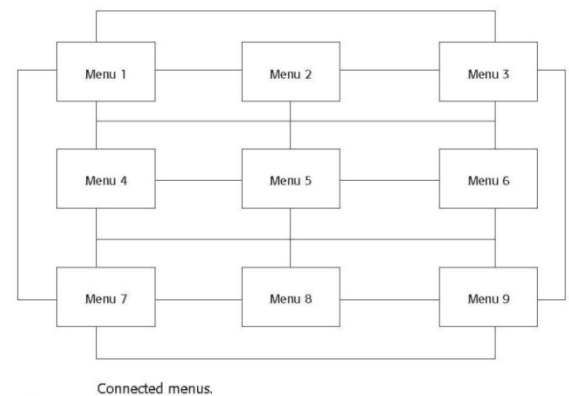
### Hierarchical Menus

- When many relationships exist between menu alternatives, and some menu options are only appropriate depending upon a previous menu selection, a hierarchical structure is the best solution.
- A hierarchical structure results in an increasing refinement of choice as menus are stepped through, for example, from options, to suboptions, from categories to subcategories, from pages to sections to subsections, and so on.
- A hierarchical structure can best be represented as an inverse tree, leading to more and more branches as one moves downward through it. Hierarchical structures are characterized by depth and breadth, depth being the number of choice levels one must traverse to reach the destination, breadth being the number of alternatives found at each level.
- Common examples of hierarchical design today are found in menu bars with their associated pull-downs, and in Web sites with their navigation links. The order and structure of branching in a hierarchy is preset and the normal order of flow one-way, top down.
- A disadvantage of a hierarchical scheme is that the defined branching order may not fit the users conception of the task flow. If users are not familiar with the hierarchical menu, or are unable to predict what suboptions lie below a particular choice, they may go down wrong paths and find it necessary to go back up the tree to change a choice, or perhaps even return to the top-level menu.
- Hierarchies be consistent with user expectations, and that choice uncertainties be reduced as much as possible. It must also be easy to back upward through the tree to facilitate exploration of the tree. A hierarchical menu is illustrated in Figure



### Connected Menus

- Connected menus are networks of menus all interconnected in some manner. Movement through a structure of menus is not restricted to a hierarchical tree, but is permitted between most or all menus in the network.
- From the user's perspective there is no top-down traversal of the menu system but an almost unhindered wandering between any two menus of interest.
- A connected menu system may be cyclical, with movement permitted in either direction between menus, or acyclical, with movement permitted in only one direction.
- These menus also vary in connectivity, the extent to which menus are linked by multiple paths.
- The biggest advantage of a connected menu network is that it gives the user full control over the navigation flow.
- Its disadvantage is its complexity, and its navigation may be daunting for an inexperienced user.



### Event-Trapping Menus

- Event Trapping menus provide an ever-present background of control over the system's state and parameters while the user is working on a foreground task.
- Event-trapping menus generally serve one of three functions.
  - (1) They may immediately change some parameter in the current environment (bold a piece of text)
  - (2) They may take the user out of the current environment to perform a function without leaving the current environment (perform a spell check), or
  - (3) They may exit the current environment and allow the user to move to a totally new environment (Exit).
- These menus can also change content based upon the system state, or an event, existing at that moment. A Paste option in a word-processing application, for example, will only function if there is something in a clipboard to paste.

## Functions of Menus

- From the user's perspective, a menu can be used to perform several functions, to *navigate* to a new menu, to *execute* an action or procedure, to *display* information, or to *input* data or parameters.

### Navigation to a New Menu

- Each user selection causes another menu in a hierarchical menu tree to be displayed. The purpose of each selection is to steer the user toward an objective or goal. Selection errors may lead the user down wrong paths, and cost time and, perhaps, aggravation, but these errors are nondestructive and usually undoable.

### Execute an Action or Procedure

- A user selection directs the computer to implement an action or perform a procedure. The action may be something like opening or closing a file, copying text, or sending a message.
- In some cases execution may only occur after a hierarchical menu tree is navigated. In other cases actions may be performed as successive hierarchical menus are encountered and traversed.
- Selection errors may or may not have serious consequences, depending upon the nature of the action. Accidental selection of critical irreversible actions must be prevented in interface design.

### Displaying Information

- The main purpose of selecting a menu choice may simply be to display information. The user may be searching for specific information in a database or browsing the Web.
- The user's focus is primarily on the information desired and less on the selection function.
- In many cases, information retrieval may occur only after a hierarchical menu tree is navigated. The content material and the user's interests will determine the paths followed.
- Users may spend considerable time and effort understanding and processing uncovered information in order to evaluate subsequently displayed menu choices.
- Wrong turns in the process will again cost time and perhaps aggravation, but these errors are nondestructive and usually undoable.

### Data or Parameter Input

- Each selection specifies a piece of input data for the system or provides a parameter value. Data or values may be input on a single menu or spread over a hierarchy of menus. The user's focus is primarily on the information being provided and, again, less on the selection function. Selection errors can easily be corrected if detected by the system.

## **Content of Menus**

- A menu consists of four elements, its *context*, its *title*, its *choice descriptions*, and its *completion instructions*

### **Menu Context**

- A menu's context provides information to keep the user oriented. This kind of information is critical in complex or hierarchical menu systems, where loss of position or disorientation can easily occur.
- Feedback is necessary that tells users where they are in a process, what their past choices were, and possibly how much farther they still have to navigate.
- Verbal linkage, spatial linkage, or both may be used to provide navigation feedback.
- *Verbal linkage* involves providing, on the current menu screen, a listing of choices made on previous menus that have led to this position. It also involves assuring the user that the displayed menu is the menu desired. Its title should mirror the option selected on the previous menu, and its content should reflect its title.
- *Spatial linkage* can be accomplished by graphic methods. Each succeeding menu screen can be displayed overlapping the previous menu screen so a succession of choices can be seen in a single view. A sense of progress and distance can then be easily ascertained.

### **Menu Title**

- A menu's title provides the context for the current set of choices. The title must reflect the choice selected on the previously displayed menu.

### **Choice Descriptions**

- Choice descriptions are the alternatives available to the user. These descriptions can range from a mnemonic, numeric, or alphabetized listing of choices to single words or phrases to full sentences or more.
- The style chosen will reflect the experience of the user (novice or expert), the nature of the choices (well-learned alternatives or not), the nature of the selection mechanism (keyboard or mouse), and the nature of the system (business system application or Web page).

### **Completion Instructions**

- Completion instructions tell users how to indicate their choices. They may include the rationale for why the user is being asked to make this choice and the impact the choice will have on subsequent processes.
- Explicit instructions may be needed for first time or casual users of a system.
- Experienced users will find overly verbose instructions unnecessary.
- The needs of all system users, and the nature of the system, must again be considered in creating this kind of on-screen guidance.

## Kinds of Graphical Menus

- Providing the proper kinds of graphical menus to perform system tasks is also critical to system success. The best kind of menu to use in each situation depends on several factors.
- The following must be considered:
  - ✓ The number of items to be presented in the menu.
  - ✓ How often the menu is used.
  - ✓ How often the menu contents may change.

### Menu Bar

- The highest-level graphical system menu is commonly called the menu bar. A menu bar consists of a collection of descriptions that serve as headings or titles for a series of actions on an associated pull-down menu. A menu bar choice by itself should not initiate an action.
- Menu bars often consist of a series of textual words. Some products have placed the choices within buttons.



Menu bar composed of text.



Menu bar composed of buttons.

- Each menu bar item is the top level of a hierarchical menu. It will have a pull-down menu associated with it, detailing the specific actions that may be performed.
- Menu bars are used to present application alternatives or choices to the screen user. Each system provides a default set of menu bar commands (for ex: File, Edit, View, Window, Help).
- The *advantages* of menu bars are that they:
  - ✓ Are always visible, reminding the user of their existence.
  - ✓ Are easy to browse through.
  - ✓ Are easy to locate consistently on the screen.
  - ✓ Usually do not obscure the screen working area.
  - ✓ Usually are not obscured by windows and dialog boxes.
  - ✓ Allow for use of keyboard equivalents.
- The *disadvantages* of menu bars are that:
  - ✓ They consume a full row of screen space.
  - ✓ They require looking away from the main working area to find.
  - ✓ They require moving pointer from the main working area to select.
  - ✓ The menu options are smaller than full-size buttons, slowing selection time.
  - ✓ Their horizontal orientation is less efficient for scanning.
  - ✓ Their horizontal orientation limits number of choices that can be displayed.

**Display**

- All primary windows must have a menu bar.
- All menu bars must have an associated pull-down menu containing at least two choices.
- Do not allow the user to turn off the display of the menu bar.
- If all the items in its associated pull-down menu are disabled, then disable the menu bar item.
  - ✓ Display the disabled item in a visually subdued manner.
  - ✓ However, the disabled pull-down menu must always be capable of being pulled down so that the choices may be seen.

**Location**

- Position choices horizontally over the entire row at the top of the screen, just below the screen title.
- A large number of choices may necessitate display over two rows.

**Title**

- The window title will be the menu bar title.

**Item Descriptions**

- The menu item descriptions must clearly reflect the kinds of choices available in the associated pull-down menus.
- Menu item descriptions will be the “titles” for pull-down menus associated with them.
- Use mixed-case letters to describe choices.
- Use single-word choices whenever possible.
- Do not display choices that are never available to the user.

**Organization**

- Follow standard platform ordering schemes where they exist.
  - Place application-specific choices where they fit best.
- Order choices left-to-right with:
  - — Most frequent choices to the left.
  - — Related information grouped together.
- Choices found on more than one menu bar should be consistently positioned.
- Left-justify choices within the line.
- When choices can be logically grouped, provide visual logical groupings, if possible.
- Help, when included, should be located at the right side of the bar.



**Layout**

- Indent the first choice one space from the left margin.
- Leave at least three spaces between each of the succeeding choices (except for Help which will be right-justified).
- Leave one space between the final choice and the right margin.



File Edit Options Window Help

**Separation**


- Separate the bar from the remainder of the screen by:
  - A different background, or
  - Solid lines above and below.

**Other Components**

- Keyboard equivalent mnemonics should be included on menu bars.
- Keyboard accelerators, to a window indicators, and cascade indicators need not be included.

**Selection Indication**

- Keyboard cursor:
  - Use a reverse video, or reverse color, selection cursor to surround the choice.
  - Cover the entire choice, including one blank space before and after the choice word.



File Edit Options Window Help

- Pointer:
  - Use reverse video, or reverse color, to highlight the selected choice.

**Pull-Down Menu**

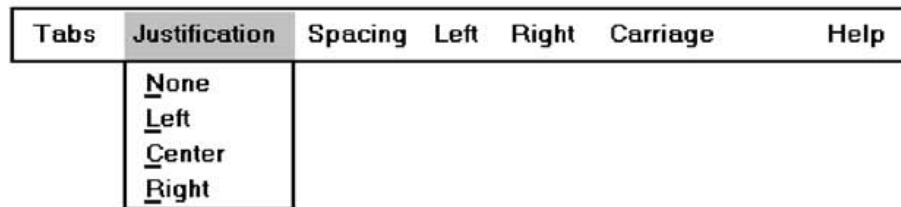
- Proper usage:
  - To initiate frequently used application actions that take place on a wide variety of different windows.
  - A small number of items.
  - Items best represented textually.
  - Items whose content rarely changes.

The *advantages* of pull-down menus are:

- ✓ The menu bar cues a reminder of their existence.
- ✓ They may be located relatively consistently on the screen.
- ✓ No window space is consumed when they are not used.
- ✓ They are easy to browse through.



- ✓ Their vertical orientation is most efficient for scanning.
- ✓ Their vertical orientation is most efficient for grouping.
- ✓ Their vertical orientation permits more choices to be displayed.
- ✓ They allow for display of both keyboard equivalents and accelerators.
- The *disadvantages* of pull-down menus are:
  - ✓ They require searching and selecting from another menu before seeing options.
  - ✓ They require looking away from main working area to read.
  - ✓ They require moving the pointer out of working area to select (unless using keyboard equivalents).
  - ✓ The items are smaller than full-size buttons, slowing selection time.
  - ✓ They may obscure the screen working area.



Menu bar pull-down.

### ***Display***

- Display all possible alternatives.
- Gray-out or dim items that cannot be chosen due to the current state of an application.

### ***Location***

- Position the pull-down directly below the selected menu bar choice.

### ***Size***

- Must contain a minimum of two choices.
- Restrict to no more than 5 to 10 choices, preferably 8 or less.

### ***Title***

- Not necessary on a pull-down menu. The title will be the name of the menu bar item chosen.

### ***Item Descriptions***

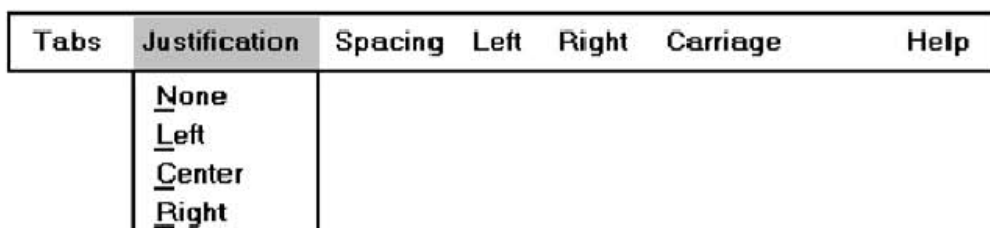
- Use mixed-case, headline-style words to describe choices.
  - If the choices can be displayed graphically, for example, as fill-in patterns, shades, or colors, textual descriptions are not necessary.
- Do not:
  - Identify a menu item by the same wording as its menu title.
  - Change the meaning of menu items through use of the Shift key.
  - Use scrolling in pull-downs.
  - Place instructions in pull-downs.

### Organization

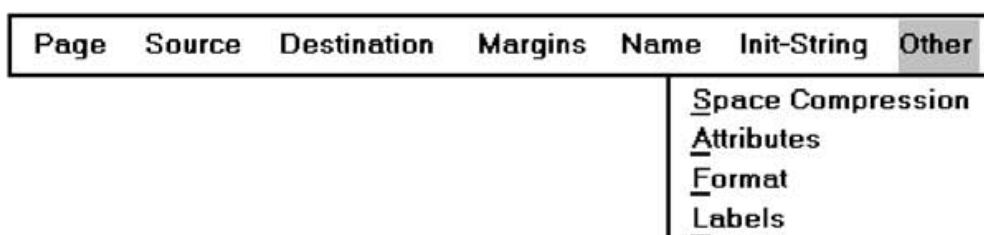
- Follow standard platform ordering schemes when they exist.
  - Place application-specific choices where they fit best.
- Place frequent or critical items at the top.
- Separate destructive choices from other choices.
- Align choices into columns, with:
  - Most frequent choices toward the top.
  - Related choices grouped together.
  - Choices found on more than one pull-down consistently positioned.
- Left-align choice descriptions.
- Multicolumn menus are not desirable. If necessary, organize top-to-bottom, then left-to-right.

### Layout

- Leave the menu bar choice leading to the pull-down highlighted in the selected manner (reverse video or reverse color).
- Physically, the pull-down menu must be wide enough to accommodate the longest menu item description and its cascade or accelerator indicator.
- Align the first character of the pull-down descriptions under the second character of the applicable menu bar choice.
- Horizontally, separate the pull-down choice descriptions from the pull-down borders by two spaces on the left side and at least two spaces on the right side.
  - The left-side border will align with the left side of the highlighted menu bar choice.
  - The right-side border should extend, at minimum, to the right side of its highlighted menu bar choice.



— Pull-downs for choices on the far right side of the menu bar, or long pull-down descriptions, may require alignment to the left of their menu bar choice to maintain visibility and clarity.



### **Groupings**

- Provide groupings of related pull-down choices:
  - Incorporate a solid line between major groupings.
  - Incorporate a dotted or dashed line between subgroups.
  - Left-justify the lines under the first letter of the columnized choice descriptions.
  - Right-justify the lines under the last character of the longest choice description.
  - Display the solid line in the same color as the choice descriptions.



### **Mark Toggles or Settings**

- If a menu item establishes or changes the attributes of data or properties of the interface, mark the pull-down choice or choices whose state is current or active “on.”
  - For nonexclusive items, display a check mark to the left of the item description.
    - If the two states of a setting are not obvious opposites, a pair of alternating menu item descriptions should be used to indicate the two states.
  - For exclusive choices, precede the choice with a contrasting symbol such as a diamond or circle.

### **Pull-Downs Leading to Another Pull-Down**

- If a pull-down choice leads to another pull-down, provide a cascade indicator as follows:
  - Place an arrow or right-pointing triangle after the choice description.
  - Align the triangles to the right side of the pull-down.
  - Display the triangle in the same color as the choice descriptions.



### **Pull-Downs Leading to a Window**

- For pull-down choices leading to a window:
  - Place an ellipsis (three dots) after the choice description.
  - Do not separate the dots from the description by a space.
  - Display the ellipsis in the same color as the choice descriptions.



### **Keyboard Equivalents and Accelerators**

- Provide unique mnemonic codes by which choices may be selected through the typewriter keyboard.
  - Indicate the mnemonic code by underlining the proper character.
- Provide key accelerators for choice selection.
  - Identify the keys by their actual key-top engravings.
  - Use a plus (+) sign to indicate that two or more keys must be pressed at the same time.
  - Enclose the key names within parentheses ( ).

— Right-align the key names, beginning at least three spaces to the right of the longest choice description.

— Display the key alternatives in the same color as the choice descriptions.

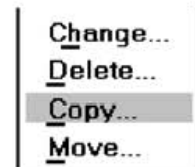
<u>F</u> ind...	(Ctrl+F)
F <u>i</u> nd <u>N</u> ext	(F3)
F <u>i</u> nd <u>P</u> revious	(Shift+F3)
<u>R</u> eplace...	(Ctrl+R)

### Separation

- Separate the pull-down from the remainder of the screen, but visually relate it to the menu bar:
  - Using a background color the same as the menu bar.
  - Displaying choice descriptions in the same color as the menu bar.
  - Incorporating a solid-line border completely around the pull-down in the same color as the choice descriptions.
- A drop shadow (a heavier shaded line along two borders that meet) may also be included.

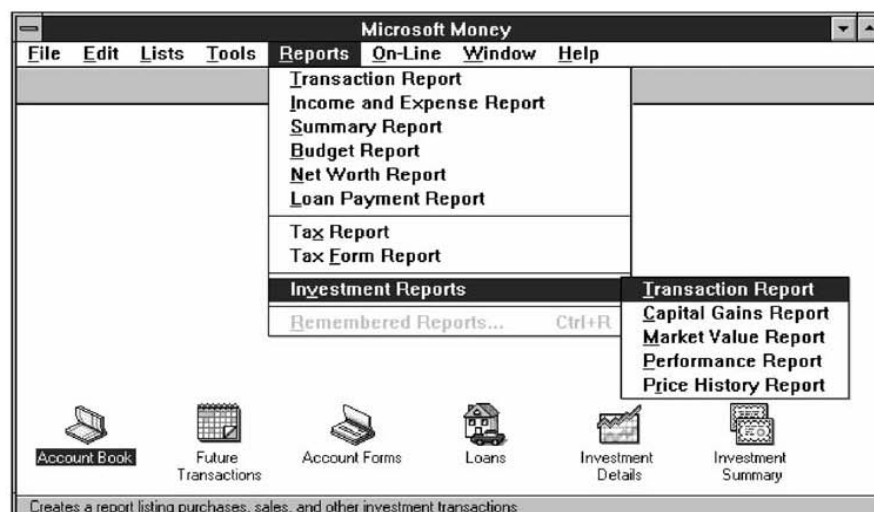
### Selection Cursor

- Use a reverse video, or reverse color, selection cursor the same color as the menu bar to surround the choice.
- Create a consistently sized cursor as wide as the pull-down menu.



## Cascading Menus

- Proper usage:
  - To reduce the number of choices presented together for selection (reduce menu breadth).
  - When a menu specifies many alternatives and the alternatives can be grouped in meaningful related sets on a lower-level menu.
  - When a choice leads to a short, fixed list of single-choice properties.
  - When there are several fixed sets of related options.
  - To simplify a menu.
  - Avoid using for frequent, repetitive commands.



Cascading menu.

- The *advantages* of cascading menus are that:
  - ✓ The top-level menus are simplified because some choices are hidden.
  - ✓ More first-letter mnemonics are available because menus possess fewer alternatives.
  - ✓ High-level command browsing is easier because subtopics are hidden.
- The *disadvantages* of cascading menus are:
  - ✓ Access to submenu items requires more steps.
  - ✓ Access to submenu items requires a change in pointer movement direction.
  - ✓ Exhaustive browsing is more difficult; some alternatives remain hidden as pull downs become visible.

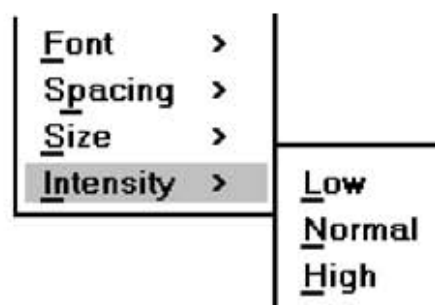
### ***Cascade Indicator***

- Place an arrow or right-pointing triangle to the right of each menu choice description leading to a cascade menu.
- Separate the indicator from the choice description by one space.
- Display the indicator in the same color as the choice descriptions.



### ***Location***

- Position the first choice in the cascading menu immediately to the right of the selected choice.
- Leave the choice leading to the cascading menu highlighted.



### ***Levels***

- Do not exceed three menu levels (two cascades).
  - Only one cascading menu is preferred.

### ***Title***

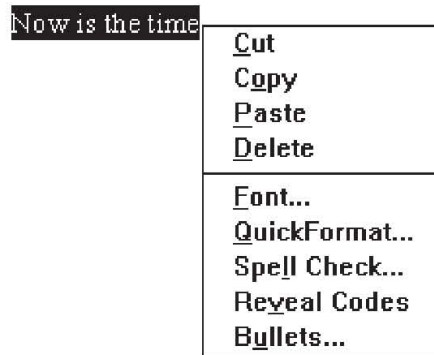
- Not necessary on the cascading menu.
  - The title will be the name of the higher-level menu item chosen.

### ***Other Guidelines***

- Follow the organization, content, layout, separation, and selection cursor guidelines for the kind of menu from which the menu cascades.

## Pop-up Menus

- Use to present alternatives or choices within the context of the task.



Pop-up menu.

- The *advantages* of pop-up menus are:
  - ✓ They appear in the working area.
  - ✓ They do not use window space when not displayed.
  - ✓ No pointer movement is needed if selected by button.
  - ✓ Their vertical orientation is most efficient scanning.
  - ✓ Their vertical orientation most efficient for grouping.
  - ✓ Their vertical orientation allows more choices to be displayed.
  - ✓ They may be able to remain showing (“pinned”) when used frequently.
  - ✓ They allow for display of both keyboard equivalents and accelerators.
- The *disadvantages* of pop-up menus are:
  - ✓ Their existence must be learned and remembered.
  - ✓ Means for selecting them must be learned and remembered.
  - ✓ They require a special action to see the menu (mouse click).
  - ✓ Items are smaller than full-size buttons, slowing selection time.
  - ✓ They may obscure the screen working area.
  - ✓ Their display locations may not be consistent.

### **Display**

- Provide a pop-up menu for common, frequent, contextual actions.
  - If the pointer is positioned over an object possessing more than one quality (for example, both text and graphics), at minimum present actions common to all object qualities.
- Items that cannot be chosen due to the current state of an application should not be displayed.
- Continue to display a pop-up until:
  - A choice is selected.
  - An action outside the pop-up is initiated.
  - The user removes the pop-up.

***Location***

- Position the pop-up:
  - Centered and to the right of the object from which it was requested.
  - Close enough to the pointer so that the pointer can be easily moved onto the menu.
  - But not so close that the pointer is positioned on an item, possibly leading to accidental selection.
- If the pointer is positioned in such a manner that the pop-up would appear off screen or clipped, position the menu:
  - As close as possible to the object, but not covering the object.
  - So that it appears fully on the screen.

***Size***

- Restrict the pop-up to no more than 5 to 10 choices, preferably 8 or less.

***Title***

- Not necessary on a pop-up menu.
- If included, clearly describe the menu's purpose.
- Locate in a centered position at the top.
- Display in uppercase or mixed-case letters.
- Separate it from the menu items by a line extending from the left menu border to the right border.

***Other Guidelines***

- Arrange logically organized and grouped choices into columns.
- If items are also contained in pull-down menus, organize pop-up menus in the same manner.
- Left-align choice descriptions.
- Use mixed-case headline-style words to describe choices.
- Separate groups with a solid line the length of the longest choice description.
- If the choice leads to a pop-up window, place an ellipsis after the choice description.
- To separate the pop-up from the screen background:
  - Use a contrasting, but complementary background.
  - Incorporate a solid line border around the pull-down.

## Tear-off Menus

- A tear-off menu is a pull-down menu that can be positioned anywhere on the screen for constant referral. As such, it possesses all the characteristics of a pull-down. It may also be called a *pushpin*, *detachable*, or *roll-up* menu.
- Its purpose is to present alternatives or choices to the screen user that are needed infrequently at some times and heavily at other times.
- **Advantages/disadvantages.** No space is consumed on the screen when the menu is not needed. When needed, it can remain continuously displayed. It does require extra steps to retrieve, and it may obscure the screen working area.
- Tear-off menus are most useful for expert users. Use these menus in situations where the items are sometimes frequently selected and other times infrequently selected. Items should be small in number and rarely change in content.
- A typical use would be to detach and permanently leave displayed a pull-down menu when it must be frequently used.
- Since a tear-off menu is a pull-down style, all pull-down guidelines should be followed.

## Iconic Menus

- Use to remind users of the functions, commands, attributes, or application choices available.
- Create icons that:
  - Help enhance recognition and hasten option selection.
  - Are concrete and meaningful.
  - Clearly represent choices.



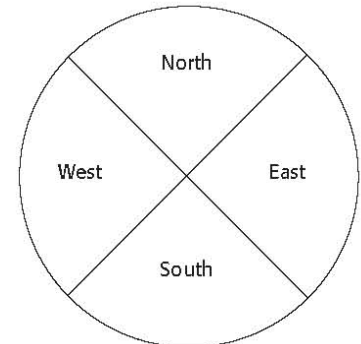
Iconic menu (from Microsoft Windows).

- **Advantages/disadvantages.** Pictures help facilitate memory of applications, and their larger size increases speed of selection. Pictures do, however, consume considerably more screen space than text, and they are difficult to organize for scanning efficiency. To create meaningful icons requires special skills and an extended amount of time. Iconic menus should be used to designate applications or special functions within an application. Icons must be meaningful and clear. They should help enhance recognition and hasten option selection.



## Pie Menu

- Consider using for:
  - Mouse-driven selections, with one- or two-level hierarchies, short lists, and choices conducive to the format.
- A pie menu is a circular representation of menu items, as illustrated in Figure.
- Research has found that this style of menu yields higher performance than the typical vertical array, especially when the menu tasks are unrelated.
- Their basic advantage is that, when presented with the mouse pointer positioned in the pie's center, average movement to any pie wedge is shorter.



Pie menu.

## Default Menu Items

- Every system will provide a set of standard menu items. Using the default items will reduce design time and encourage interface consistency. System learning time will also be reduced. Microsoft Windows, for example, provides the following standard and optional menu bar items and pull-down actions.

### *File*

- A standard element, the File menu provides all the commands needed to open, create, and save files. Some standard File functions are:
  - ✓ New
  - ✓ Open
  - ✓ Close
  - ✓ Save
  - ✓ Save As
  - ✓ Print Preview
  - ✓ Print
  - ✓ Exit

**Edit**

- A standard element, the Edit menu provides commands that affect the state of selected objects.

Some standard Edit functions are:

- ✓ Undo
- ✓ Cut
- ✓ Copy
- ✓ Paste
- ✓ Select All
- ✓ Find
- ✓ Replace

**View**

- An optional element, the View menu provides commands that affect the perspective, details, and appearance of the application. They affect the view, not the data itself. The view functions are application-specific and include the following:

- ✓ Toolbars
- ✓ Status Bar
- ✓ Magnify
- ✓ Zoom In
- ✓ Zoom Out
- ✓ Grid Points

**Window**

- The Window menu, an optional element, provides commands to manipulate entire windows. Included are items such as:

- ✓ New Window
- ✓ Arrange All
- ✓ Hide
- ✓ Show

**Help**

- The Help menu, a standard element, provides Help commands, including:

- ✓ Contents
- ✓ Search for Help On
- ✓ How to Use Help
- ✓ About (Application)

### Menu Proper Usage Summary

Menu Bar	To identify and provide access to: <ul style="list-style-type: none"> <li>• Common and frequently used application actions.</li> <li>• Actions that take place in a wide variety of different windows.</li> </ul>
Pull-Down Menu	For frequently used application actions that take place in a wide variety of different windows: <ul style="list-style-type: none"> <li>• A small number of items (5–10).</li> <li>• Items rarely changing in content.</li> </ul>
Cascading Menu	To simplify a higher-level menu. To provide easier browsing of a higher-level menu. For mutually exclusive choices. Restrict to 1–2 cascades.
Pop-Up Menu	For: <ul style="list-style-type: none"> <li>• Frequent users.</li> <li>• Frequently used contextual commands.</li> <li>• A small number of items (5–10).</li> <li>• Items rarely changing in content.</li> <li>• Items that require a small amount of screen space.</li> </ul>
Tear-Off Menu	For items: <ul style="list-style-type: none"> <li>• Sometimes frequently selected.</li> <li>• Sometimes infrequently selected.</li> <li>• Small in number (5–10).</li> <li>• Rarely changing in content.</li> </ul>
Iconic Menu	To designate applications available. To designate special functions within an application.

### Functions Not Represented by Default Items

- Having established the usability of the standard menu functions, additional system functions must be identified. Commands to accomplish these functions must be created and added to the pertinent menus. Command design guidelines include the following.

#### *Labels*

- General:
  - Provide a label for each command.
  - Use labels that indicate:
    - The purpose of the command, or
    - The result of what happens when the command is selected.
  - Use familiar, short, clear, concise words.
  - Use distinctive wording.
  - Use mixed case, with the first letter capitalized.
  - Begin commands with verbs or adjectives, not nouns.
  - Preferably, use only one word.
    - If multiple words are required for clarity, capitalize the first letter of each significant word.
    - Do not use sentences as labels.

— Provide an ellipsis (. . .) to indicate that another window will result from selection of a command.

- Do not use the ellipsis when the following window is a confirmation or warning.

➤ Dynamic labels:

— As contexts change, dynamically change the label wording to make its meaning clearer in the new context.

- For example, after a cut operation, Undo may be changed to Undo Cut.

***Disabled Commands***

➤ When a command is not available, indicate its disabled status by displaying it grayed out or subdued.

➤ If selection of a disabled command is attempted, provide a message in the information area that the “Help” function will explain why it is disabled.

***Navigation and Selection***

➤ General:

— Permit multiple methods for selecting commands.

➤ Keyboard equivalents:

— Assign a mnemonic for each command.

— A mnemonic should be as meaningful as possible. Use:

- The first letter of the command, or if duplications exist,
- The first letter of another word in the command, or
- Another significant consonant in the command.

— For standard commands, use mnemonics provided by the tool set.

➤ Keyboard accelerators:

— Assign keyboard accelerators for frequently used commands.

— For standard commands, use keyboard accelerators provided by the tool set.

## 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.

**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.
- Visibility—the portion that can be seen.
- 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.
- 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.

#### ***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.

### ***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. 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.

### ***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.

### ***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.

### ***Constraints in Window System Design***

- Windowing systems, in spite of their appeal and obvious benefits, have failed to completely live up to their expectations.
- The problems with windowing systems can be attributed to three factors: historical considerations, hardware limitations, and human limitations.

### ***Historical Considerations***

- ✓ There are few concrete window design guidelines to aid designers.
- ✓ This lack of guidelines makes it difficult to develop acceptable and agreeable window standards.
- ✓ Standardization of the interface is also inhibited by other factors.
- ✓ 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.
- ✓ 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.
- ✓ 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.

### **Components of a Window**

- A typical window may be composed of so many elements. Some appear on all windows; others only on certain kinds of windows, or under certain conditions.



Microsoft Windows primary 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. Also, textual materials, which are usually read from left to right, fit most efficiently within this structure.

**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.
- 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.

**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, since the window can no longer be increased in size. The restore button—typically inscribed with a pair of 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
- Sizing buttons are included on primary windows only.



- 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.
- On a primary window this command is accessed from the Help drop-down menu.



What's This? button.

### **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 area, extending its entire length.
- Horizontal scrolling is accomplished through a scroll bar located at the bottom of the work area.

**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 allows the user to:
  - ✓ Examine two parts of a document at the same time.
  - ✓ Display different, yet simultaneous, views of the same information.

**Toolbar**

- 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.
- If a horizontal scroll bar is included in the window, position the command area just below 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.

**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*.

## Microsoft Windows Components

COMPONENT	WINDOWS CONTAINING COMPONENT		
	PRIMARY	SECONDARY	DIALOG BOX
<p><i>Frame or Border</i></p> <ul style="list-style-type: none"> <li>• Boundary to define shape.</li> <li>• If sizable, contains control points for resizing.</li> </ul>	X	X	X
<p><i>Title Bar Text</i></p> <ul style="list-style-type: none"> <li>• Name of object being viewed in window.</li> <li>• Control point for moving window.</li> </ul>	X	X	X
<p><i>Title Bar Icon</i></p> <ul style="list-style-type: none"> <li>• Small version of icon for object being viewed.</li> <li>• Access point for commands that apply to the object.</li> </ul>	X		
<p><i>Title Bar Buttons</i></p> <ul style="list-style-type: none"> <li>• Shortcuts to specific commands.</li> </ul>	X	X	X
<p><i>Close</i></p>	X	X	X
<p><i>Minimize/Maximize/Restore</i></p>	X		
<p><i>What's This?</i></p> <ul style="list-style-type: none"> <li>– Displays context-sensitive Help about any object displayed on window.</li> </ul>		X	X
<p><i>Menu Bar</i></p> <ul style="list-style-type: none"> <li>• Provides basic and common application commands.</li> </ul>	X		
<p><i>Status Bar</i></p> <ul style="list-style-type: none"> <li>• An area used to display status information about what is displayed in window.</li> </ul>	X		
<p><i>Scroll Bar</i></p> <ul style="list-style-type: none"> <li>• Standard control to support scrolling.</li> </ul>	X		
<p><i>Size Grip</i></p> <ul style="list-style-type: none"> <li>• Control to resize window, located at right side of status bar.</li> </ul>	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 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.
- Tiled windows, the first and oldest kind of window, are felt to have these advantages:

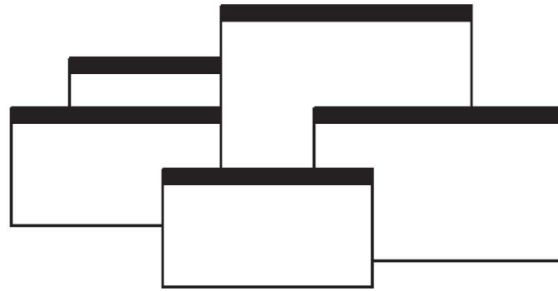


Tiled windows.

- ✓ 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.

### 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.



Overlapping windows.

- 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.



Cascading windows.

- 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.

### **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

- The Microsoft Windows windowing scheme will have the following 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.



Microsoft Windows primary window.

### Secondary Windows

- Secondary windows are supplemental windows. Secondary windows may be dependent upon a primary window or displayed independently of the primary window.
- 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
- 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.



Microsoft Windows secondary window.

- 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 secondary window can be modal or 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.
- 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:
  - 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.



### 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

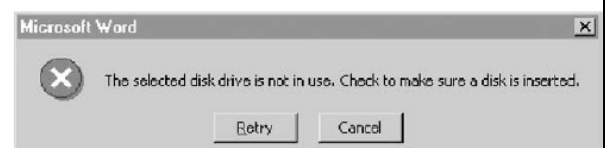
- Use for presenting the complete set of properties for an object.
- Categorize and group within property pages, as necessary.
  - Use tabbed property pages for grouping peer-related property sets.
  - The recommended sizes for property sheets are:
    - 252 DLUs wide x 218 DLUs high
    - 227 DLUs wide x 215 DLUs high
    - 212 DLUs wide x 188 DLUs high
  - Command buttons to include:
    - OK.      • Cancel.      • Apply.      • Reset.
    - Others as necessary.
  - For single property sheets, place the commands on the sheet.
  - For tabbed property pages, place the commands outside the tabbed pages.

### Property Inspectors

- Use for displaying only the most common or frequently accessed object 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.



Microsoft Windows message box.

- 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.

### Palette Windows

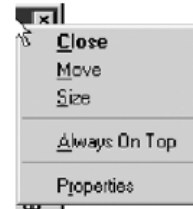
- Use to present a set of controls.
- Design as resizable.
- Alternately, design them as fixed in size.



Microsoft Windows palette window.

### 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.



Microsoft Windows pop-up window.

### Window Management

- Microsoft Windows also provides several window management schemes, a *singledocument interface*, a *multiple-document interface*, *workbooks*, and *projects*.
- To choose the right scheme to present an application's collection of related tasks or processes, consider a number of design factors, including: the intended users and their skill level, the application and its objects or tasks, and the most effective use of display space

### 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 interfaces 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.