

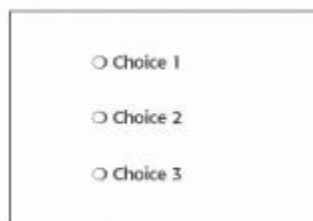
## **UNIT-5**

### **SYSTEM MENUS AND NAVIGATION SCHEMES**

#### **Structures of Menus**

##### **Single Menus**

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

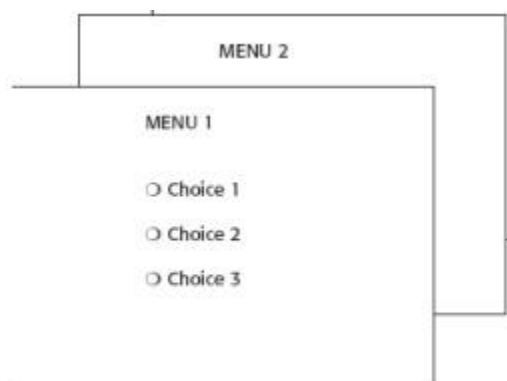


A single menu may be iterative if it requires data to be entered into it and this data input is subject to a validity check that fails. The menu will then be represented to the user with a message requesting reentry of valid data.

##### **Sequential Linear Menus**

Sequential linear menus are presented on a series of screens possessing only one path.

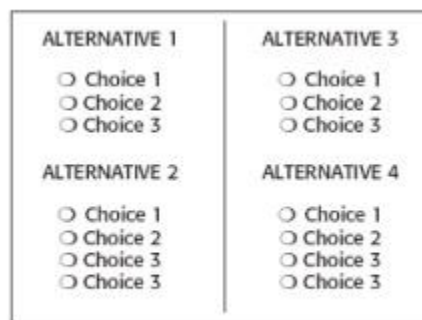
The menu screens are presented in a preset order, and, generally, their objective is for specifying parameters or for entering data.



Sequential path menus have several shortcomings. A long sequence may become tedious as menu after menu is presented.

## Simultaneous Menus

Instead of being presented on separate screens, all menu options are available simultaneously



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.

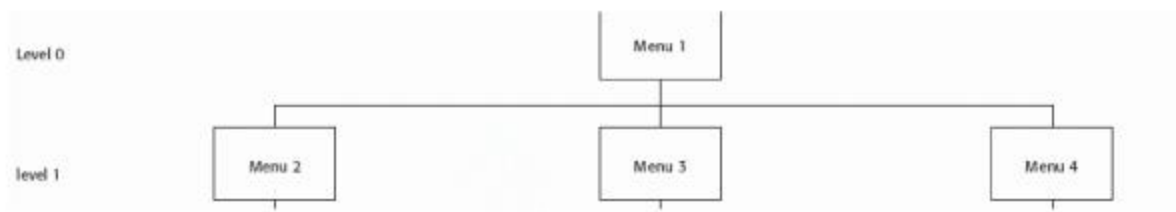
Presenting many menu dependencies and relationships on a screen, especially if poorly indicated, can also be very confusing

## Hierarchical Menus

A hierarchical structure results in an increasing refinement of choice as menus are stepped through, for example, from options, to sub options, 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 move downward through it.

Common examples of hierarchical design today are found in menu bars with their associated pull-downs



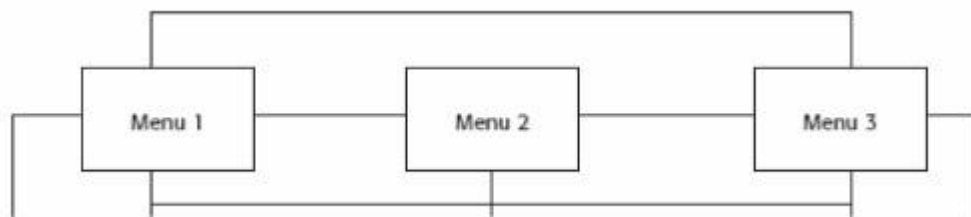
A disadvantage of a hierarchical scheme is that the defined branching order may not fit the user's conception of the task flow.

If users are not familiar with the hierarchical menu, or are unable to predict what sub options 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

## 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.

A connected menu system may be cyclical, with movement permitted in either direction between menus, or a cyclical, 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.

## **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.

They may immediately change some parameter in the current environment (bold a piece of text), they may take the user out of the current environment to perform a function without leaving the current environment (perform a spell check), or they may exit the current environment and allow the user to move to a totally new environment (Exit).

## **Functions of Menus**

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.

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 content material and the user's interests will determine the paths followed.

The user's focus is primarily on the information desired and less on the selection function. 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.

## **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.

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.

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.

## **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.

## **Completion Instructions**

Completion instructions tell users how to indicate their choices

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.

## **Formatting of Menus**

What follows is a series of guidelines for formatting menus.

### **Consistency**

Provide consistency with the user's expectations.

Provide consistency in menu:

- Formatting, including organization, presentation, and choice ordering.
- Phrasing, including titles, choice descriptions, and instructions.
- Choice selection methods.
- Navigation schemes.

### **Display**

If continual or frequent references to menu options are necessary, permanently display the menu in an area of the screen that will not obscure other screen data.

If only occasional references to menu options are necessary, the menu may be presented on demand.

- Critical options should be continuously displayed, however.

## **Presentation**

Ensure that a menu and its choices are obvious to the user by presenting them with a unique and consistent structure, location, and/or display technique.

Ensure that other system components do not possess the same visual qualities as menu choices.

## **Organization**

Provide a general or main menu.

Display:

- All relevant alternatives.
- Only relevant alternatives.
  - Delete or gray-out inactive choices.

## **Complexity**

Provide both simple and complex menus.

Simple: a minimal set of actions and menus.

Complex: a complete set of actions and menus.

## **Item Arrangement**

Align alternatives or choices into single columns whenever possible.

- Orient for top-to-bottom reading.
- Left-justify descriptions.

If a horizontal orientation of descriptions must be maintained:

- Organize for left-to-right reading.

## **Ordering**

Order lists of choices by their natural order, or

For lists associated with numbers, use numeric order.

For textual lists with a small number of options (seven or less), order by:

- Sequence of occurrence.
- Frequency of occurrence.
- Importance.
- Semantic similarity.

Use alphabetic order for:

- Long lists (eight or more options).
- Short lists with no obvious pattern or frequency.

## **Groupings**

Create groupings of items that are logical, distinctive, meaningful, and mutually exclusive.

Categorize them in such a way as to:

- Maximize the similarity of items within a category.
- Minimize the similarity of items across categories.

Present no more than six or seven groupings on a screen.

## **Line Separators**

Separate vertically arrayed groupings with subtle solid lines.

Separate vertically arrayed sub groupings with subtle dotted or dashed lines.

## **Phrasing the Menu**

A menu must communicate to the user information about:

- The nature and purpose of the menu itself.
- The nature and purpose of each presented choice.
- How the proper choice or choices may be selected.

## **Menu Titles**

Main menu:

- Create a short, simple, clear, and distinctive title, describing the purpose of the entire series of choices.



Submenus:

- Submenu titles must be worded exactly the same as the menu choice previously selected to display them.

General:

- Locate the title at the top of the listing of choices.
- Spell out the title fully using either an:
  - Uppercase font.
  - Mixed-case font in the headline style.

### **Menu Choice Descriptions**

Create meaningful choice descriptions that are familiar, fully spelled out, concise, and distinctive.

Descriptions may be single words, compound words, or multiple words or phrases.

Exception: Menu bar items should be a single word (if possible).

Place the keyword first, usually a verb.

Use the headline style, capitalizing the first letter of each significant word in the choice description.

Use task-oriented not data-oriented wording.

Use parallel construction.

### **Menu Instructions**

For novice or inexperienced users, provide menu completion instructions.

- Place the instructions in a position just preceding the part, or parts, of the menu to which they apply.

Left-justify the instruction and indent the related menu choice descriptions a minimum of three spaces to the right.

Leave a space line, if possible, between the instructions and the related menu choice descriptions.

- Present instructions in a mixed-case font in sentence style.

For expert users, make these instructions easy to ignore by:

- Presenting them in a consistent location.
- Displaying them in a unique type style and/or color.

## Intent Indicators

Cascade indicator:

- To indicate that selection of an item will lead to a submenu, place a triangle or right-pointing solid arrow following the choice.
- A cascade indicator must designate every cascaded menu.

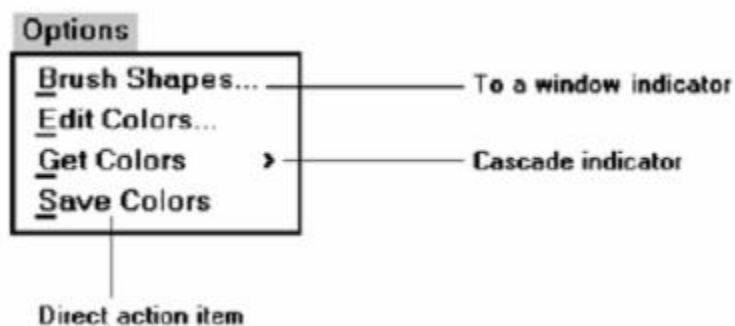
To a window indicator:

– For choices that result in displaying a window to collect more information, place an ellipsis (. . .) immediately following the choice.

- Exceptions—do not use when an action:
- Causes a warning window to be displayed.
- May or may not lead to a window.

Direct action items:

- For choices that directly perform an action, no special indicator should be placed on the menu.



## Keyboard Equivalents

To facilitate keyboard selection of a menu choice, each menu item should be assigned a keyboard equivalent mnemonic.

The mnemonic should be the first character of the menu item's description.

- If duplication exists in first characters, use another character in the duplicated item's description.
- Preferably choose the first succeeding consonant.

Designate the mnemonic character by underlining it.

Use industry-standard keyboard access equivalents when they exist.

### **Keyboard Accelerators**

For frequently used items, provide a keyboard accelerator to facilitate keyboard selection.

The accelerator may be one function key or a combination of keys.

- Function key shortcuts are easier to learn than modifier plus letter shortcuts.

Pressing no more than two keys simultaneously is preferred.

- Do not exceed three simultaneous keystrokes.

Use a plus (+) sign to indicate that two or more keys must be pressed at the same time.

Accelerators should have some associative value to the item.

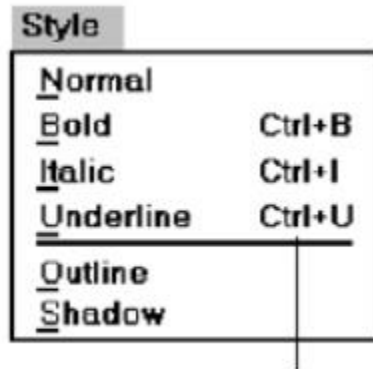
Identify the keys by their actual key top engraving.

If keyboard terminology differences exist, use:

- The most common keyboard terminology.

Terminology contained on the newest PCs.

Separate the accelerator from the item description by three spaces.



## Kinds of Graphical Menus

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.

1. Menu Bar

2. Pull-Down Menus

3. Cascading Menus

4. PopUp Menus

5. Tear Off Menus

6. Iconic Menus

7. Pie Menus

### **1. Menu Bar:**

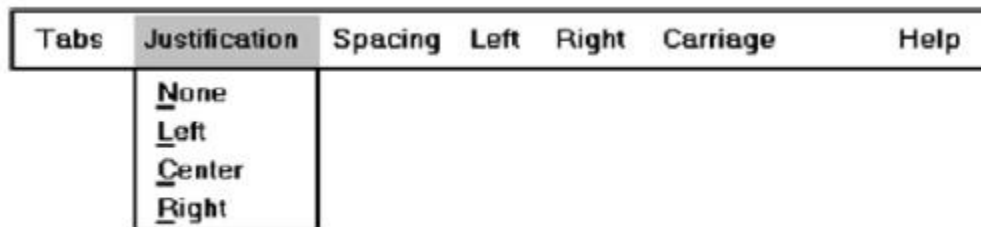
The highest level graphical system menu is commonly called the menu bar. The Menu bar consists of a series of textual words or buttons. These are used to represent the application alternatives or choices to the user. All primary window must have a menu bar. All menu bars must have an associated pull down menu which containing atleast 2 choices. It does not allows the user to turn off the display of the menu bar. The window title will be the menu bar title. It do not display choices that are never available to the user. It separate the bar from the remainder of the screen by a different background. They require the moving pointer from the main working area to select the choices. It consumes a full row of

screen space. It usually does not hide the screen working area. This is called Menu Bar.



## 2. Pull-Down Menus:

These are the first level menus which are used to provide access to the common and most frequently used application actions that take place on a wide variety of different windows. There are a smaller number of items. No window space is consumed when they are not used. These menus are easy to browse. The items are smaller than full-sized buttons. It displays all possible alternatives. Each consists of at least 2 choices. Title is not necessary for a pull-down menu. If a pull-down choice leads to another pull-down, then it provides a cascade indicator to denote it. This is called Pull-Down Menu.



## 3. Cascading Menus:

Proper usage:

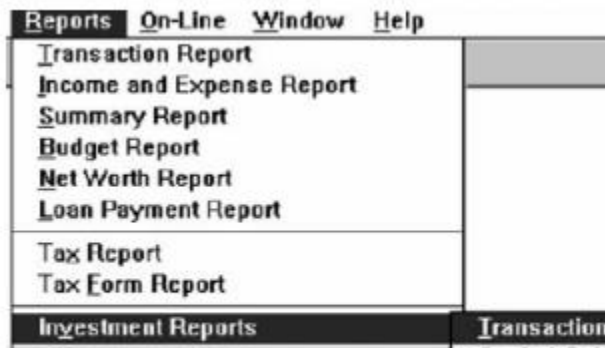
- To reduce the number of choices presented together for selection (reduce menu breadth).
- When a menu specifies many alternatives and the alternatives can be grouped in meaningful related sets on a lower-level menu.
- When a choice leads to a short, fixed list of single-choice properties.
- When there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

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.



#### 4. Popup Menus:

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

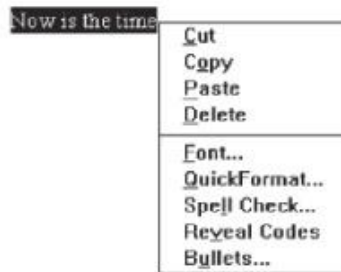
The advantages of pop-up menus are:

- 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.

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.



## 5.Tear Off Menus:

A tear off menu is a pull down menu that can be positioned any where on the screen.It possess all the characteristics of a pull down.It require extra steps to retrieve.It hides the screen working area.This is called Tear off menus.

## 6.Iconic Menus:

It is the picture of menu items or objects in a graphic form.The purpose of an iconic menu is to remind users of the functions,commands,attributes or application choices available.Icons must be meaningful and clear.This is called 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.



### **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.

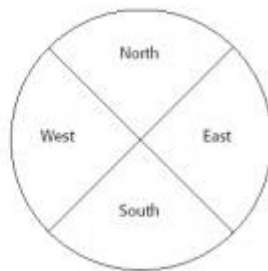
Icons must be meaningful and clear. They should help enhance recognition and hasten option selection.

### **7.Pie Menus:**

A pie menu is a circular representation of menu items. It can be used as an alternative for pull down menu or popup menus. This is called pie menus.

Consider using for:

Mouse-driven selections, with one- or two-level hierarchies, short lists, and choices conducive to the format.



## **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.

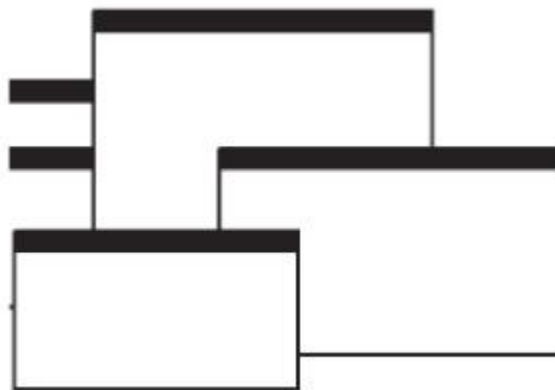
They are perceived as crowded and more visually complex because window borders are flush against one another, and they fill up the whole screen. Crowding is accentuated if borders contain scroll bars or control icons. Viewer attention may be drawn to the border, not the data.

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.

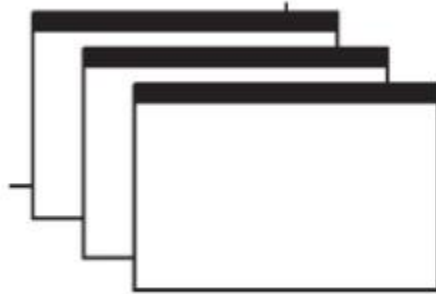
That overlapping windows represent a three-dimensional space is not always realized by the user.

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.

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.

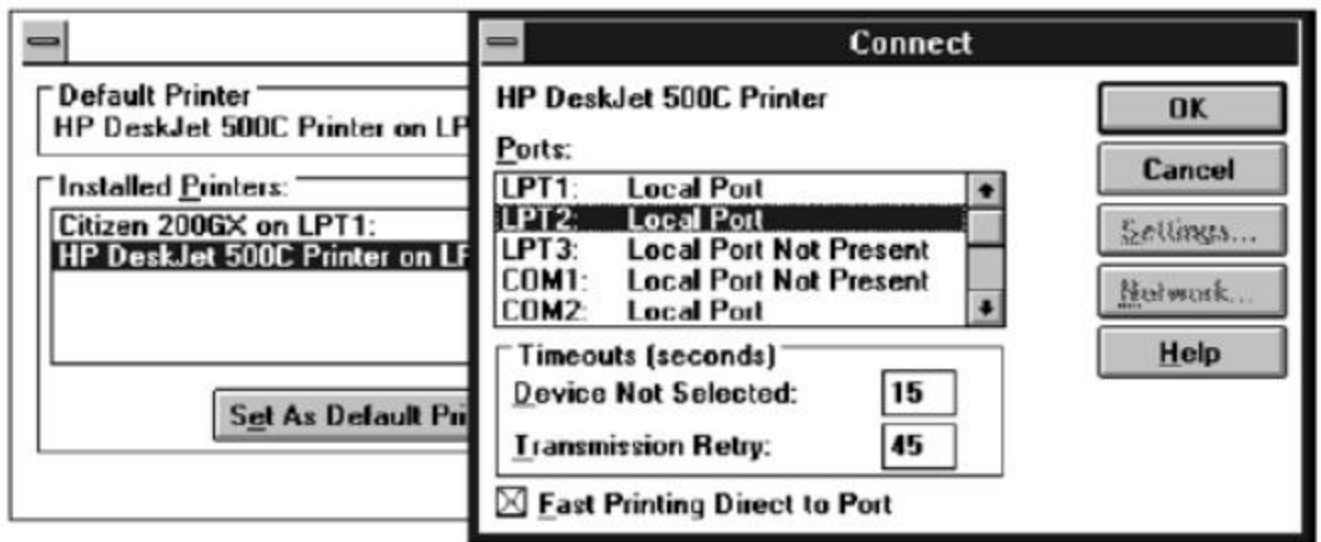
### Guidelines:

- 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.

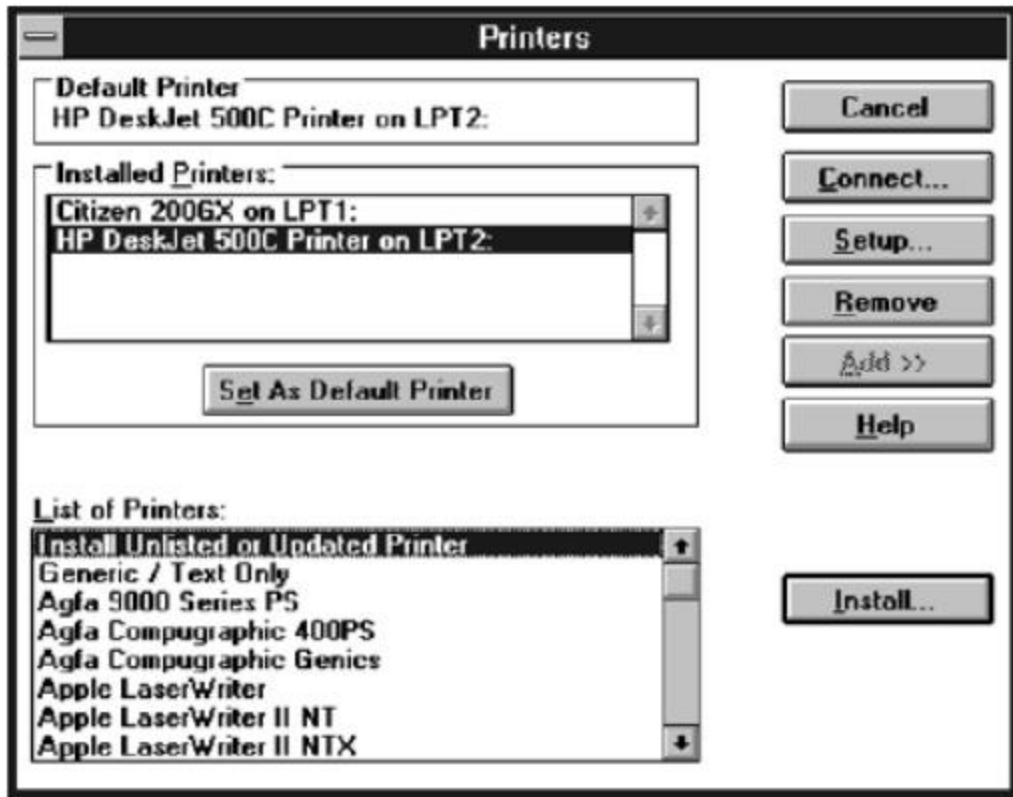
Unfolding:

- 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.



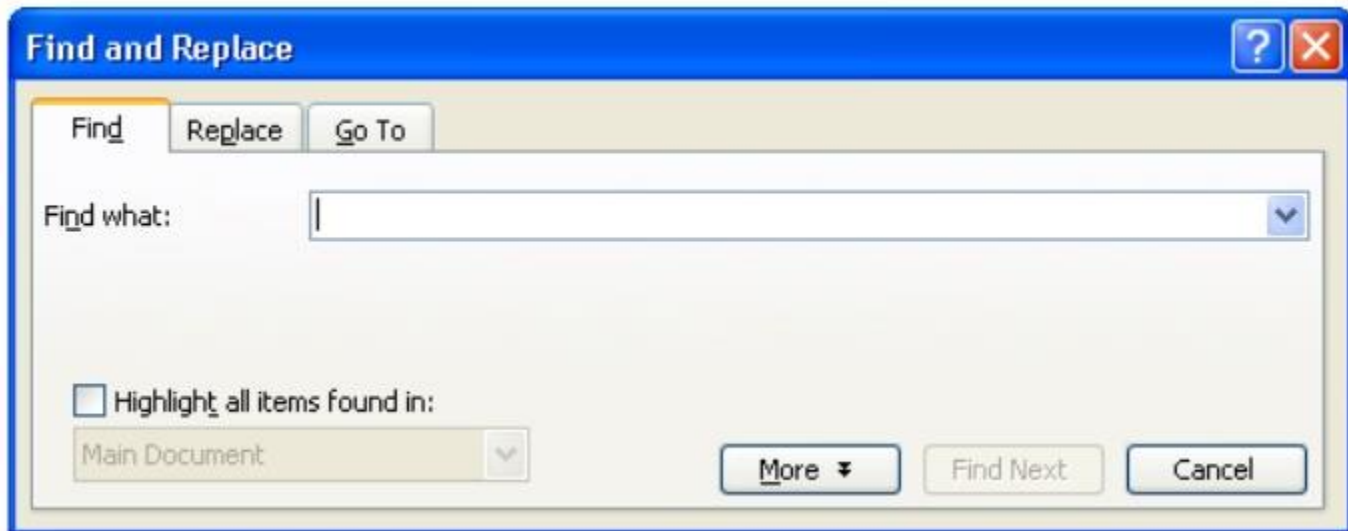
**Cascaded Window**





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:

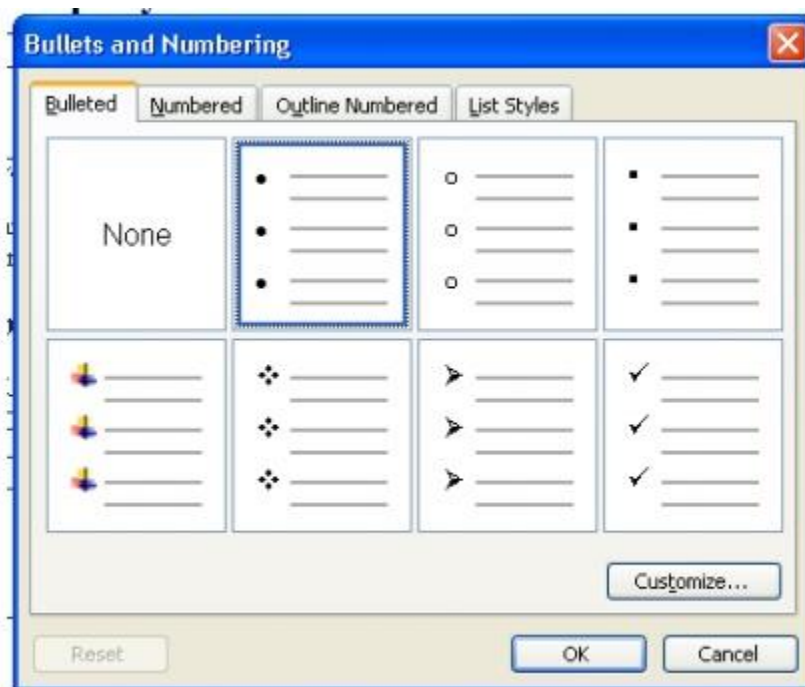
- OK.
- Cancel.
- Others as necessary.

## 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.
- 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

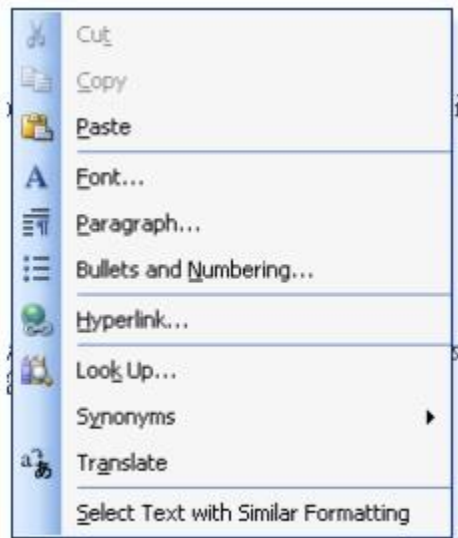


Use to present a set of controls.

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.

## 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.

## **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— 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.

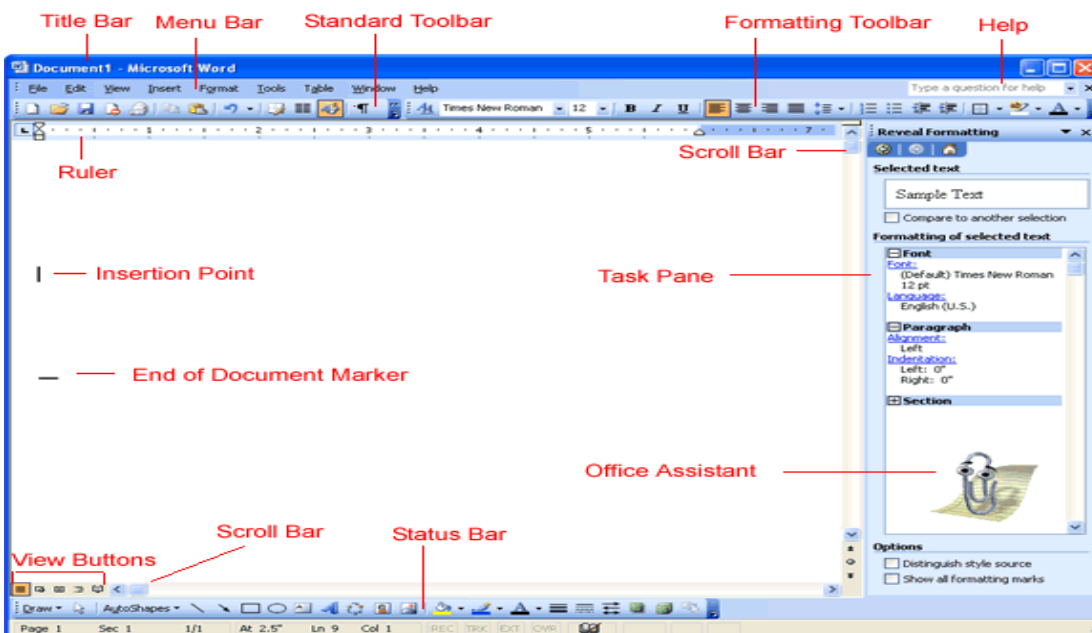
When the grip is dragged the window resizes, following the same conventions as the sizing border. Three angled parallel lines in the lower-right corner of a window designate the size grip.

## 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*.



# 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:**

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.

## **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.

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.