CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC
No.132, AECS Layout
I.T. Park Road, Bangalore 560 037
T: +91 80 28524466/77, F : +91 80 28524630
E: info@cmr.ac.in I www.cmr.ac.in

IAT – 1 SOLUTION
USER INTERFACE DESIGN

1. a) Define user interface design and its importance.[05M]

Definition of UID

User interface design is a subset of a field of study called *human-computer interaction* (HCI). Human-computer interaction is the study, planning, and design of how people and computers work together so that a person's needs are satisfied in the most effective way.

Importance of good design:

With today's technology and tools, and our motivation to create really effective and usable interfaces and screens, why do we continue to produce systems that are inefficient and confusing or, at worst, just plain unusable? Is it because:

1. We don't care?
2. We don't possess common sense?
3. We don't have the time?
4. We still don't know what really makes good design?

A well-designed interface and screen is terribly important to our users. It is their window to view the capabilities of the system. To many, *it is* the system, being one of the few visible components of the product we developers create. It is also the vehicle through which many critical tasks are presented. These tasks often have a direct impact on an organization's relations with its customers, and its profitability.

1. b) Explain the concept of direct manipulation.[05M]

Direct Manipulation possess the following characteristics:

1. System is portrayed as extension of real world:

The system simply replicates actions and objects and portrays them on a different medium, the screen.

2. Continuous visibility of objects and actions:

Objects are continuously visible.Cursor action and motion occurs in physically obvious and intuitively natural ways.Nelson (1980) described this concept as "virtual reality," a representation of reality that can be manipulated. Hatfield (1981) is credited with calling it "WYSIWYG" (what you see is what you get). Rutkowski (1982) described it as "transparency," where one's intellect is applied to the task, not the tool.

3. Actions are rapid and incremental with visible display of results:

The results of actions are immediately displayed visually on the screen in their new and current form.

4. Incremental actions are easily reversible:

Finally, actions, if discovered to be incorrect or not desired, can be easily undone.

Earlier Direct Manipulation Systems

The concept of direct manipulation actually preceded the first graphical system. The earliest full-screen text editors possessed similar characteristics. Screens of **text resembling a piece of paper** on one's desk could be created (extension of real world) and then reviewed in their entirety (continuous visibility). Editing or restructuring could be easily accomplished (through rapid incremental actions) and the results immediately seen. Actions could be reversed when necessary.

Indirect Manipulation

Direct manipulation of *all* screen objects and actions may not be feasible because of the following:

 The operation may be difficult to conceptualize in the graphical system.
 The graphics capability of the system may be limited.
 The amount of space available for placing manipulation controls in the window border may be limited.
 It may be difficult for people to learn and remember all the necessary operations and actions.

Indirect manipulation substitutes words and text, such as pull-down or pop-up menus, for symbols, and substitutes typing for pointing. Most window systems are a combination of both direct and indirect manipulation.

2. a) Explain the characteristics of GUI. [10M]

1. Sophisticated Visual Presentation
2. Pick-and-Click Interaction
3. Restricted Set of Interface Options
4. Visualization
5. Object Orientation
6. Use of Recognition Memory
7. Concurrent Performance of Functions

1. Sophisticated visual presentation:

Visual presentation is the visual aspect of the interface. It is what people see on the screen.

The sophistication of a graphical system permits displaying lines, including drawings and icons. It also permits the displaying of a variety of character fonts, including different sizes and styles. The display of 16 million or more colors is possible on some screens. Graphics also permit animation and the presentation of photographs and motion video. The meaningful interface elements visually presented to the user in a graphical system include:

 windows - primary, secondary, or dialog boxes
 menus - menu bar, pulldown, pop-up, cascading
 icons - represent objects such as programs or files
 Assorted screen-based controls - text boxes, list boxes, combination boxes, settings, scroll bars, and buttons
 mouse pointer and cursor.

The objective is to reflect visually on the screen the real world of the user as realistically, meaningfully, simply, and clearly as possible.

2. Pick-and-click interaction:

To identify the element for a proposed action is commonly referred to as *pick*, the signal to perform an action as *click*.

 The primary mechanism for performing this pick-and-click is most often the mouse and its buttons. The user moves the mouse pointer to the relevant element (pick) and the action is signaled (click).
 The secondary mechanism for performing these selection actions is the keyboard. Most systems permit pick-and-click to be performed using the keyboard as well.

3. **A restricted set of interface options:**

The array of alternatives available to the user is what is presented on the screen or what may be retrieved through what is presented on the screen, nothing less, nothing more. This concept fostered the acronym WYSIWYG.

4. visualization:

Visualization is a cognitive process that allows people to understand information that is difficult to perceive, because it is **either too voluminous or too abstract**. It involves changing an entity's representation to reveal gradually the structure and/or function of the underlying system or process. Presenting specialized graphic portrayals facilitates visualization.

The best visualization method for an activity depends on what people are trying to learn from the data. The goal is not necessarily to reproduce a realistic graphical image, but to produce one that conveys the most relevant information. **Effective visualizations can facilitate mental insights, increase productivity and more accurate use of data.**

5. Object orientation:

A graphical system consists of **objects and actions**. *Objects* are what people see on the screen.Objects can be composed of *subobjects.*

A) IBM's Common User Access application breaks objects into three meaningful classes:

1.Data

2.Container

3.device.

Data objects present information. This information, either text or graphics, normally appears in the body of the screen. It is, essentially, the screen-based controls for information collection or presentation organized on the screen.

Container objects are objects to hold other objects. They are used to group two or more related objects for easy access and retrieval. There are three kinds of container objects:

- Workplace
- Folders
- Workareas.

The *workplace* is the desktop, the storage area for all objects. *Folders* are general-purpose containers for long-term storage of objects. *Workareas* are temporary storage folders used for storing multiple objects currently being worked on.

Device objects represent physical objects in the real world, such as printers or trash baskets. These objects may contain others for acting upon. A file, for example, may be placed in a printer for printing of its contents.

B) Microsoft Windows specifies the characteristics of objects depending upon the relationships that exist between them.These relationships are called collections, constraints, composites, and containers.

A ***collection*** is the simplest relationship—the objects sharing a common aspect. A collection might be the result of a query or a multiple selection of objects. Operations can be applied to a collection of objects.

A ***constraint*** is a stronger object relationship. Changing an object in a set affects some other object in the set. A document being organized into pages is an example of a constraint.

A ***composite*** exists when the relationship between objects becomes so significant that the aggregation itself can be identified as an object. Examples include a range of cells organized into a spreadsheet, or a collection of words organized into a paragraph.

A ***container*** is an object in which other objects exist. Examples include text in a document or documents in a folder. A container often influences the behavior of its content. It may add or suppress certain properties or operations of objects placed within it, control access to its content, or control access to kinds of objects it will accept.

Another important object characteristic is ***persistence***. Persistence is the maintenance of a state once it is established. An object's state (for example, window size, cursor location, scroll position, and so on) should always be automatically preserved when the user changes it.

- **Properties or Attributes of Objects**

Properties help to describe an object and can be changed by users. Examples of properties are

text styles (such as normal or italics), font sizes (such as 10 or 12 points), or window background colors (such as black or blue).

 Actions

Commands are actions that manipulate objects. They may be performed in a variety of ways, including by direct manipulation or through a command button. They are executed immediately when selected. Once executed, they cease to be relevant. Examples of commands are opening a document, printing a document, closing a window, and quitting an application.

Property/attribute specification actions establish or modify the attributes or properties of objects. When selected, they remain in effect until deselected. Examples include selecting cascaded windows to be displayed, a particular font style, or a particular color.

The following is a typical *property/attribute specification sequence:*

1. The user selects an object—for example, several words of text.
2. The user then selects an action to apply to that object, such as the action BOLD.
3. The selected words are made bold and will remain bold until selected and changed again.

A series of actions may be performed on a selected object. Performing a series of actions on an object also permits and encourages system learning through exploration.

 Application versus Object or Data Orientation

An application-oriented approach takes an action:object approach, like this:

Action> 1. An application is opened (for example, word processing).

Object> 2. A file or other object selected (for example, a memo).

An object-oriented object:action approach does this:

Object> 1. An object is chosen (a memo).

Action> 2. An application is selected (word processing).

 Views

Views are ways of looking at an object's information. IBM's SAA CUA describes four kinds of views: composed, contents, settings, and help.

Composed views present information and the objects contained within an object. They are typically associated with data objects and are specific to tasks and products being worked with. ***Contents*** views list the components of objects. ***Settings*** views permit seeing and changing object properties. ***Help*** views provide all the help functions.

6. Use of recognition memory:

Continuous visibility of objects and actions encourages use of a person's more powerful recognition memory. The "out of sight, out of mind" problem is eliminated.

7. Concurrent performance of functions:

Graphic systems may do two or more things at one time. Multiple programs may run simultaneously. When a system is not busy on a primary task, it may process background tasks (cooperative multitasking). When applications are running as truly separate tasks, the system may divide the processing power into time slices and allocate portions to each application (preemptive multitasking). Data may also be transferred between programs. It may be temporarily stored on a "clipboard" for later transfer or be automatically swapped between programs.

3. a) Compare and contrast GUI and web interface design. [05M]

GUI versus Web Design

Characteristics	GUI	WEB
Devices	User hardware variations limited. User hardware characteristics well defined Screens appear exactly as specified.	User hardware variations enormous. Screen appearance influenced by hardware being used.
User Focus	Data and applications.	Information and navigation.
Data	Typically created and used by known and trusted	Full of unknown content.
Information	Sources are trusted. Properties generally known. Typically placed into system by users or known people and organizations.	Source not always trusted. Often not placed onto the Web by users or known people and organizations.
User Tasks	Install, configure, personalize, start, use, and Open, use, and close data files. Familiarity with applications often achieved.	Link to a site, browse or read pages, fill out forms, upgrade programs. register for services, participate in transactions, download and save things.
Presentation	Windows, menus, controls, data, toolbars Presented as specified by designer. Generally standardized by toolkits and style specifications. guides.	Two components, browser and page. Within page, any combination of text, images, audio, video, and animation. May not be presented as specified by the designer—dependent on browser, monitor, and user Little standardization.

Navigation	Through menus, lists, trees, dialogs, and wizards.	Through links, bookmarks, and typed URLs.
Interaction	Interactions such as clicking menu choices, pressing buttons, selecting list choices, and cut/copy/paste occur within context of active program.	Basic interaction is a single click. This can cause extreme changes in context, which may not be noticed.
Response Time	Nearly instantaneous	Quite variable, depending on transmission speeds, page content, and so on. Long times can upset the user.
System Capability	Unlimited capability proportional to sophistication of hardware and software.	Limited by constraints imposed by the hardware, browser, software, client support, and user willingness to allow features because of response time, security, and privacy concerns.
Task Efficiency	Targeted to a specific audience with specific tasks. Only limited by the amount of programming undertaken to support it.	Limited by browser and network capabilities. Actual user audience usually not well understood. Often intended for anyone and everyone.
Consistency	Major objective exists within and across applications.Aided by platform toolkit and design guidelines. Universal consistency in GUI products generally	Sites tend to establish their own identity.Frequently standards set within a site. Frequent ignoring of GUI guidelines for identical created through toolkits and design guidelines. components, especially controls.
User Assistance	Integral part of most systems and applications. Documentation, both online and offline,	No similar help systems. Accessed through standard mechanisms. The little available help is built into the page oriented to product or service offered.
Integration	Seamless integration of all applications into the platform environment is a major objective.	Apparent for some basic functions within most Web sites (navigation, printing, and so on.) in accomplishing this objective. Sites tend to achieve individual distinction rather than integration.
Security	Tightly controlled, proportional to degree of willingness to invest resources and effort. Not an issue for most home PC users.	Renowned for security exposures. Browser-provided security options typically understood by average users. When employed, may have function-limiting side effects.

3. b) Give short notes on windows presentation styles.[05M]

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.

I) Tiled Windows

Figure 5.4 Tiled windows.

Tiled windows, the first and oldest kind of window, are felt to have these **advantages**:

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

Perceived **disadvantages** include the following:

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

II) Overlapping Windows

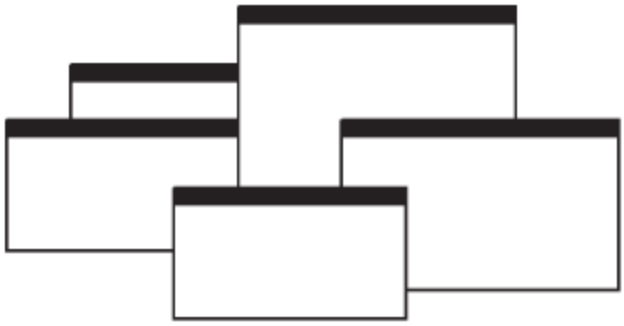


Figure 5.5 Overlapping windows.

Overlapping windows, illustrated in Figure 5.5, may be placed on top of one another like papers on a desk. They possess a three-dimensional quality, appearing to lie on different planes.
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.

III) Cascading Windows

Figure 5.6 Cascading windows.

A special type of overlapping window has the windows automatically arranged in a regular progression.

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.

IV) Picking a Presentation Style

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

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

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

4. a) Explain the general principles of User Interface Design. [10M]

General Principles

The design goals in creating a user interface are described below. They are fundamental to the design and implementation of all effective interfaces, including GUI and Web ones. These principles are general characteristics of the interface, and they apply to all aspects.

 Aesthetically Pleasing
— Provide visual appeal by following these presentation and graphic design principles:
 Provide meaningful contrast between screen elements.
 Create groupings.
 Align screen elements and groups.
 Provide three-dimensional representation.
 Use color and graphics effectively and simply.
 Clarity
— The interface should be visually, conceptually, and linguistically clear, including:
 Visual elements
 Functions
 Metaphors
 Words and text
 Compatibility
— Provide compatibility with the following:
 The user
 The task and job
 The product
— Adopt the user's perspective.
 Comprehensibility
— A system should be easily learned and understood. A user should know the following:
 What to look at
 What to do
 When to do it
 Where to do it
 Why to do it
 How to do it
— The flow of actions, responses, visual presentations, and information should be in a sensible order that is easy to recollect and place in context.
 Configurability
— Permit easy personalization, configuration, and reconfiguration of settings.
 Enhances a sense of control.
 Encourages an active role in understanding.
 Consistency
— A system should look, act, and operate the same throughout.
Similar components should:
 Have a similar look.
 Have similar uses.
 Operate similarly.
— The same action should always yield the same result.
— The function of elements should not change.
— The position of standard elements should not change.
— In addition to increased learning requirements, inconsistency in design has a number of other prerequisites and by-products, including:
— More specialization by system users.
— Greater demand for higher skills.
— More preparation time and less production time.
— More frequent changes in procedures.
— More error-tolerant systems (because errors are more likely).
— More kinds of documentation.
— More time to find information in documents.
— More unlearning and learning when systems are changed.
— More demands on supervisors and managers.

— More things to do wrong.
 Control
--The user must control the interaction.
 Actions should result from explicit user requests.
 Actions should be performed quickly.
 Actions should be capable of interruption or termination.
 The user should never be interrupted for errors.
— The context maintained must be from the perspective of the user.
— The means to achieve goals should be flexible and compatible with the user's skills, experiences, habits, and preferences.
— Avoid modes since they constrain the actions available to the user.
— Permit the user to customize aspects of the interface, while always providing a proper set of defaults.
 Directness
— Provide direct ways to accomplish tasks.
 Available alternatives should be visible.
 The effect of actions on objects should be visible.
 Efficiency
— Minimize eye and hand movements, and other control actions.
 Transitions between various system controls should flow easily and freely.
 Navigation paths should be as short as possible.
 Eye movement through a screen should be obvious and sequential.
— Anticipate the user's wants and needs whenever possible.
 Familiarity
— Employ familiar concepts and use a language that is familiar to the user.
— Keep the interface natural, mimicking the user's behavior patterns.
— Use real-world metaphors.
 Flexibility
— A system must be sensitive to the differing needs of its users, enabling a level and type of performance based upon:
 Each user's knowledge and skills.
 Each user's experience.
 Each user's personal preference.
 Each user's habits.
 The conditions at that moment.
 Forgiveness
— Tolerate and forgive common and unavoidable human errors.
— Prevent errors from occurring whenever possible.
— Protect against possible catastrophic errors.
 Predictability
— The user should be able to anticipate the natural progression of each task.
 Provide distinct and recognizable screen elements.
 Provide cues to the result of an action to be performed.
— All expectations should be fulfilled uniformly and completely.
— When an error does occur, provide constructive messages.
 Recovery
— A system should permit:
 Commands or actions to be abolished or reversed.
 Immediate return to a certain point if difficulties arise.
— Ensure that users never lose their work as a result of:
 An error on their part.
 Hardware, software, or communication problems.
 Responsiveness

— The system must rapidly respond to the user's requests.
— Provide immediate acknowledgment for all user actions:
 Visual.
 Textual.
 Auditory.
 Simplicity
— Provide as simple an interface as possible.
— Five ways to provide simplicity:
 Use progressive disclosure, hiding things until they are needed.
— Present common and necessary functions first.
— Prominently feature important functions.
— Hide more sophisticated and less frequently used functions.
 Provide defaults.
 Minimize screen alignment points.
 Make common actions simple at the expense of uncommon actions being made harder.
 Provide uniformity and consistency.
 Transparency
— Permit the user to focus on the task or job, without concern for the mechanics of the interface.
 Workings and reminders of workings inside the computer should be invisible to the user.
 Trade-Offs
— Final design will be based on a series of trade-offs balancing often- conflicting design principles.
— People's requirements always take precedence over technical requirements.

5. Mention the advantages & disadvantages of GUI in detail. [10M]

Graphical Systems

Advantages

1. Symbols recognized faster than text
2. Faster learning
3. Faster use and problem solving
4. Easier remembering
5. More natural
6. Exploits visual/spatial cues
7. Fosters more concrete thinking
8. Provides context
9. Fewer errors
10. Increased feeling of control
11. Immediate feedback
12. Predictable system responses
13. Easily reversible actions
14. Less anxiety concerning use
15. More attractive
16. May consume less space
17. Replaces national
18. Easily augmented with text
19. Low typing requirements
20. Smooth transition from command language system

Disadvantages

1. Greater design complexity
2. Learning still necessary
3. Lack of experimentally-derived design principles

4. Inconsistencies in technique and terminology
5. Working domain is the present
6. Not always familiar
7. Human comprehension limitations
8. Window manipulation requirements
9. Production limitations
10. Few tested icons exist
11. Inefficient for touch typists
12. Inefficient for expert users
13. Not always the preferred style of interaction
14. Not always the fastest style of interaction
15. Increased chances of clutter and confusion
16. The futz and fiddle
17. May consume more screen space
18. Hardware limitations

6.a) Explain windows types in detail.

Types of windows

1. Primary windows

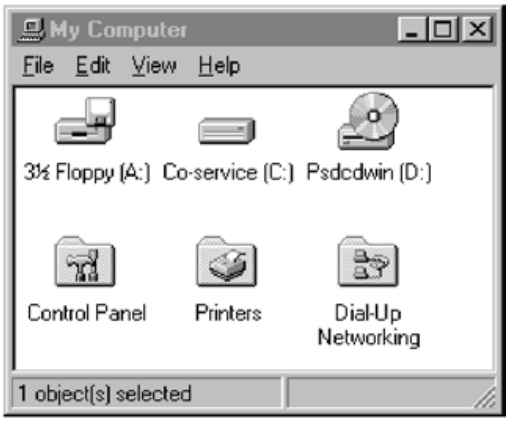


Figure 5.7 Microsoft Windows primary window.

■ 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.

2. Secondary window

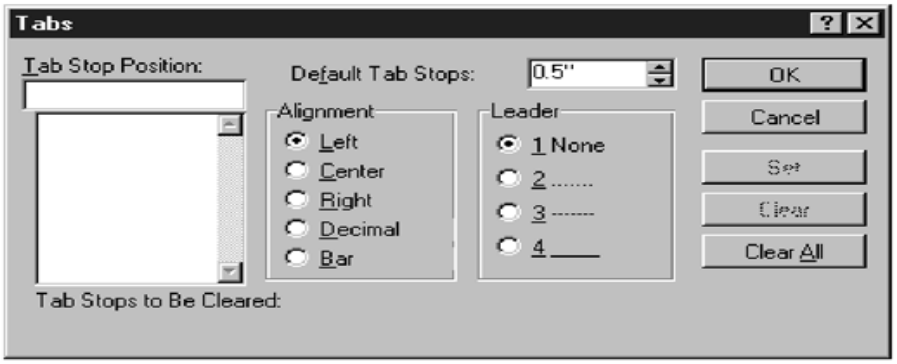


Figure 5.8 Microsoft Windows secondary window.

■ Proper usage:

— For performing subordinate, supplemental, or ancillary actions that are:

• Extended or more complex in nature.

• Related to objects in the primary window.

— For presenting frequently or occasionally used window components.

■ Important guidelines:

— Should typically not appear as an entry on the taskbar.

— A secondary window should not be larger than 263 dialog units x 263 dialog units.

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

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

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

Model-prompt and modeless-word search

■ 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 and Unfolding*

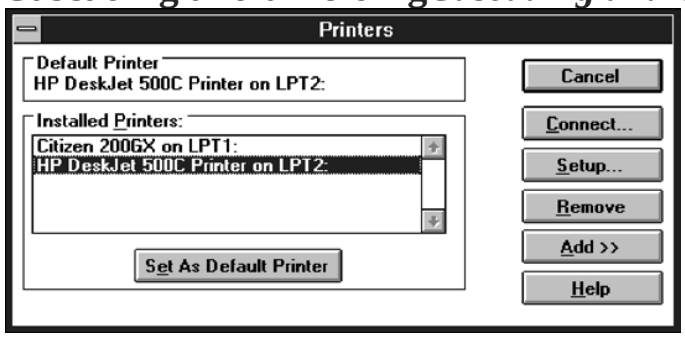


Figure 5.9 Printers secondary window with Connect... cascade button.

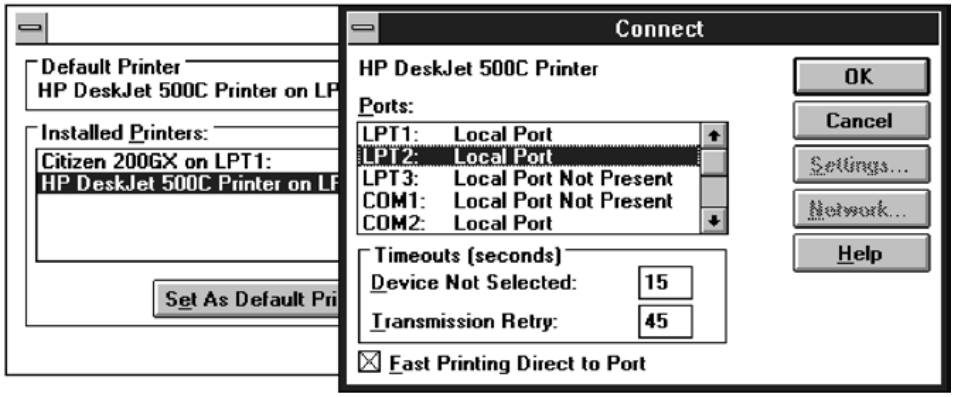


Figure 5.10 Cascading Connect secondary window.

■ 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.

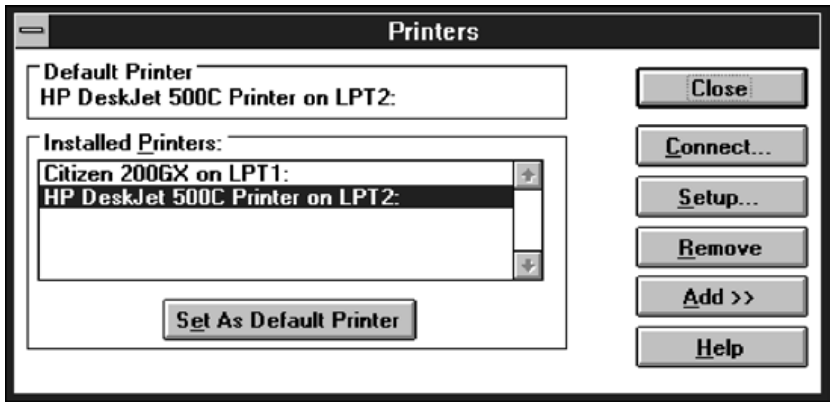


Figure 5.11 Printers secondary window with *Add* >> unfolding button.

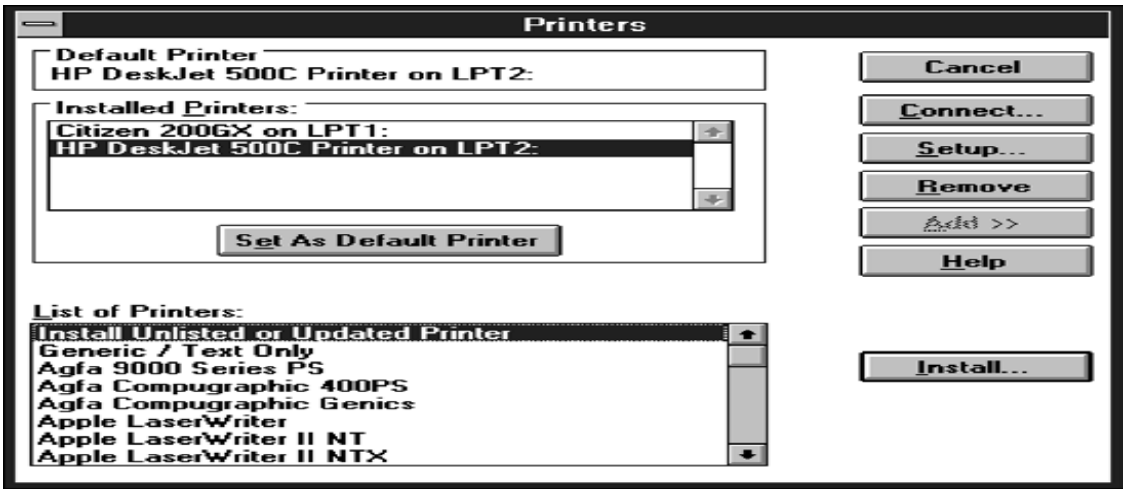
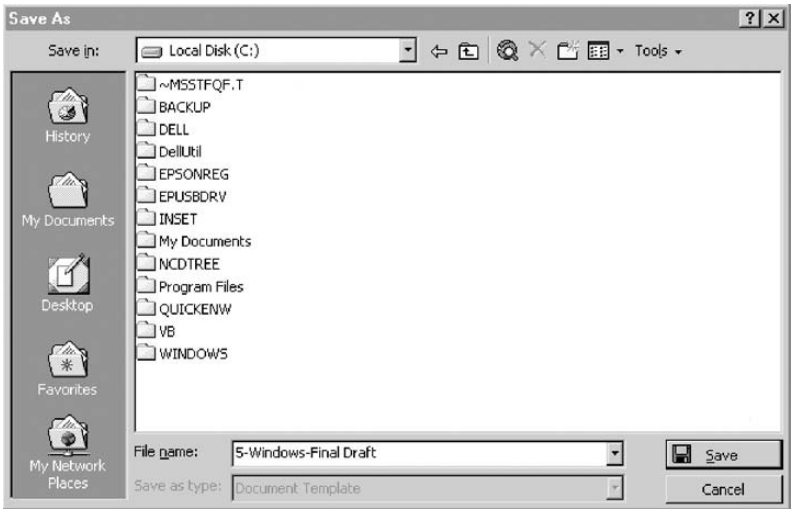


Figure 5.12 Unfolded Printers secondary window.

Dialog box

- 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 sheet. 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.

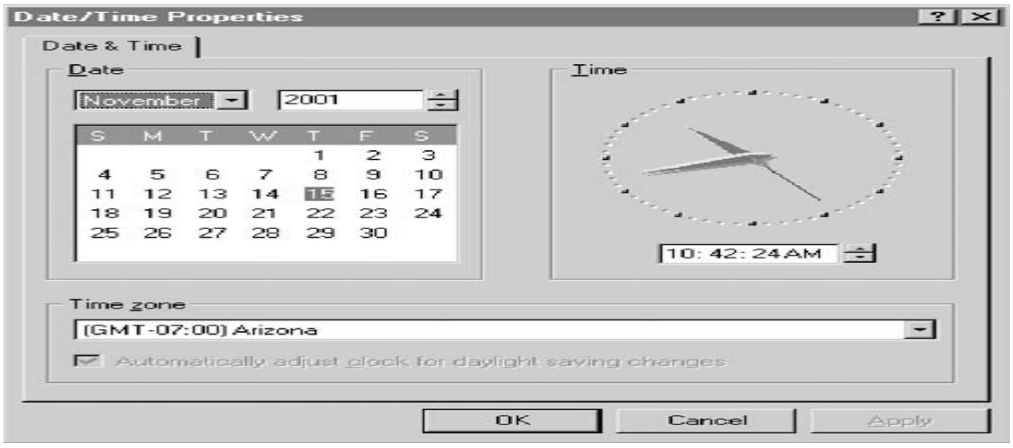


Figure 5.14 Microsoft Windows property sheet.

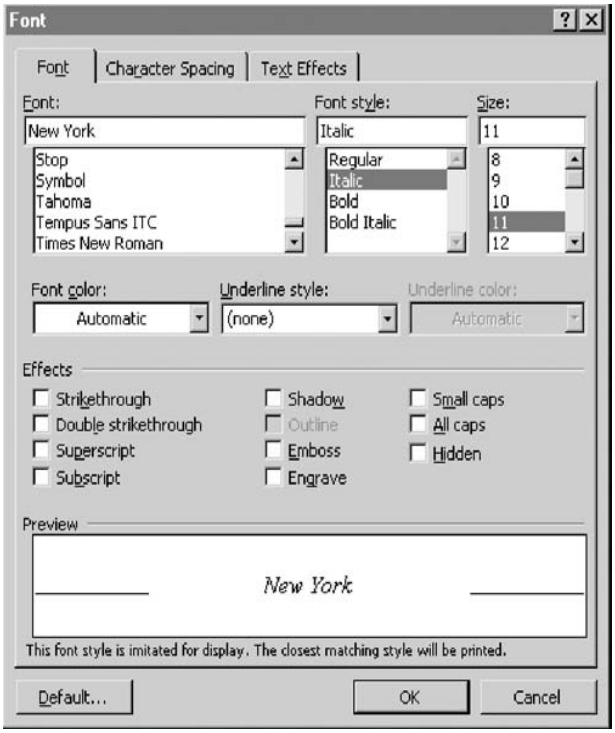


Figure 5.15 Microsoft Windows property sheet tabbed pages.

Property Inspectors

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

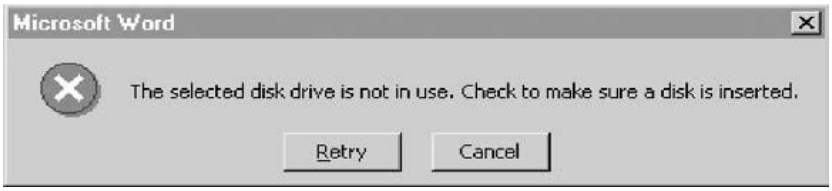
Figure 5.16 Microsoft Windows property inspector.

Message Boxes

- Use for displaying a message about a particular situation or condition.
- Command buttons to include:

OK.
Cancel.
Help.
Yes and No.
Stop.
Buttons to correct the action that caused the message box to be displayed.

- Enable the title bar close box only if the message includes a cancel button.
- Designate the most frequent or least destructive option as the default command button.



Palette Windows

- Use to present a set of controls.
- Design as resizable.

Alternately, design them as fixed in size.

Figure 5.18 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.

Figure 5.19 Microsoft Windows pop-up window.

7.a) Briefly explain components of windows with diagrams.[10M]

Components of a Window

A typical window may be composed of up to a dozen or so elements. Some appear on all windows; others only on certain kinds of windows, or under certain conditions. For consistency purposes, these elements should always be located in the same position within a window.

Figure 5.1 Microsoft Windows primary window.

Window components are

1. Frame

A window will have a frame or border, usually rectangular in shape, to define its boundaries
and distinguish it from other windows. While a border need not be rectangular, this shape is a preferred shape for most people. Also, textual materials, which are usually read from left to right, fit most efficiently within this structure. The border comprises a line of variable thickness and color.

2. 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.

3. 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 × 16 version of the icon of the object being viewed. When clicked with the secondary mouse button, the commands applying to the object are presented. Microsoft suggests that:

- If the window contains a tool or utility (that is, an application that does not create,

load, and save its own data files), a small version of the application's icon should be placed there instead.

- If the application creates, loads, and saves documents or data files and the window
  represents the view of one of its files, a small version of the icon that represents its document or data file type should be placed there.
- Even if the user has not yet saved the file, display the data file icon rather than the
  application icon, and again display the data file icon after the user saves the file.

4. Window Sizing Buttons

window's title bar must have equivalent commands on the pop-up or shortcut menu for that window. When these buttons are displayed, use the following guidelines:
When a window does not support a command, do not display its command button.
The *Close* button always appears as the rightmost button. Leave a gap between it and any other buttons. The *Minimize* button always precedes the *Maximize* button. The *Restore* button always replaces the *Maximize* button or the *Minimize* button when that command is carried out.

5. 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.

Figure 5.2 What's This? button.

6. Menu Bar

A menu bar is used to organize and provide access to actions. It is located horizontally at the top of the window, just below the title bar. A menu bar contains a list of topics or
items that, when selected, are displayed on a pull-down menu beneath the choice. A system will typically provide a default set of menu actions that can be augmented by an application. In the past, some platforms have called the menu bar an *action bar*.

7. 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*.

8. 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.

9. 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.

10. 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*. Each toolbar band includes a single-grip handle to enable the user to resize or rearrange the toolbars. When the user moves the pointer over the grip, it changes to a two-headed arrow. When the user drags the grip, the pointer changes to a split move pointer.

11. Command Area

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

12. 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.

13. 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.

8. a) Explain window characteristics in detail.[06 M]

- 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

Windows do seem to be useful in the following ways.

- ***Presentation of Different Levels of Information***
- ***Presentation of Multiple Kinds of Information***
- ***Sequential Presentation of Levels or Kinds of Information***
- ***Access to Different Sources of Information***
- ***Combining Multiple Sources of Information***
- ***Performing More Than One Task***
- ***Reminding***
- ***Monitoring***
- ***Multiple Representations of the Same Task***
- **Constraints in Window System Design**
- ***Historical Considerations*****-**lack of guidelines,Standardization
- ***Hardware Limitations*****-**screens are not large enough, the slower processing speeds and smaller memory sizes
- ***Human Limitations*****-**full screens with screens containing overlapping windows
- ***Other Limitations*****-**include the necessity for window borders

8.b) Explain button and its types.[04M]

Buttons

- Description:

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

A button comes in three styles.
1. The first resembles the control commonly found on electrical or mechanical devices and is sometimes called a pushbutton.

Fig - 1 -Command buttons

2. The second style is square or rectangular in shape with an icon or graphic inside. It may have an associated label.

Fig - 2 Toolbar buttons with controls

3.The third style is square or rectangular in shape with a symbol inscribed inside.

Fig- 3 A symbol button