CBCS SCHEME

USN 1 C R 20 I S 1 0 1 — **18CS734**

Seventh Semester B.E. Degree Examination, Dec.2023/Jan.2024

User Interface Design

Time: 3 hrs. Max. Marks: 100

Note: ***Answer any FIVE full questions, choosing ONE full question from each module.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-1

1 a. Define GUI. Write the difference between GUI and webpage design. **(10 Marks)**

b. Define user interface Design with example. Explain the importance and benefits of Good user Interface Design. **(10 Marks)**

OR

2 a. Discuss the general principles of UID. **(10 Marks)**

b. Mention the advantages and disadvantages of GUI in details. **(10 Marks)**

Module-2

3 a. What is requirement analysis? What are the methods involved in it? What is the impact of it on UI design? **(10 Marks)**

b. Define obstacles and pitfalls mention the general observation of design and common pitfalls and also explain five commandments used in Designing. **(10 Marks)**

OR

4 a. Explain the importance of human consideration in UI design with suitable example. **(10 Marks)**

b. Explain briefly about human interaction speed. **(10 Marks)**

Module-3

5 a. Explain in brief the structure of Menu's. **(10 Marks)**

b. Describe the components of a web navigation system with illustration. **(10 Marks)**

OR

6 a. Write a note on Graphical menus for the following

i) Pull down menu

ii) Pop up menu **(10 Marks)**

b. Describe at least four guidelines to be followed in phasing of menu, during the development of system menus. **(10 Marks)**

Module-4

7 a. Discuss briefly about the types of windows with example. (Any five) **(10 Marks)**

b. Write a note on the following

i) Track ball

ii) Joystick **(10 Marks)**

OR

8 a. Explain briefly about window management. **(10 Marks)**

b. Write a note on components of a windows. **(10 Marks)**

Module-5

9 a. Explain briefly the following selection control

i) Radio buttons

ii) Checkboxes (10 Marks)

b. Explain the purpose of prototypes. Discuss any two kinds of prototypes with their importance to the system developers. (10 Marks)

OR

10 a. Explain the following with respect to kinds of Tests.

i) Think – Aloud Evaluation

ii) Usability Test (10 Marks)

b. Explain the types of presentation control. (10 Marks)

* * * * *

Seventh Semester B.E. Degree Examination, Dec.2023/Jan.2024

18CS734 - User Interface Design

MODULE-1

1 a. Define GUI. Write the difference between GUI and webpage design.

GUI is a collection of techniques and mechanisms to interact with something. In a *graphical interface, the primary interaction mechanism is a pointing device of some* kind.

Difference between GUI and 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. Typically organized in a meaningful fashion. A notion of private and shared data exists.	Source not always trusted. Often not placed onto the Web by users or known people and organizations. Highly variable organization. Privacy often suspect.
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, Customer service support, if provided, usually provided. Personal support desk also usually provided.	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.
Reliability	Tightly controlled in business systems,	Susceptible to disruptions caused by user, telephone proportional to degree of willingness line and cable providers, Internet service providers, to invest resources and effort. hosting servers, and remotely accessed sites.

1 b. Define user interface Design with example. Explain the importance and benefits of Good user Interface Design.

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.

Benefits of good design:

- Training costs are lowered because training time is reduced.
- Support line costs are lowered because fewer assist calls are necessary.
- Employee satisfaction is increased because aggravation and frustration are reduced.
- Customers benefit because of the improved service they receive.

Based on an actual system requiring processing of 4.8 million screens per year and illustrated in Table 1.1

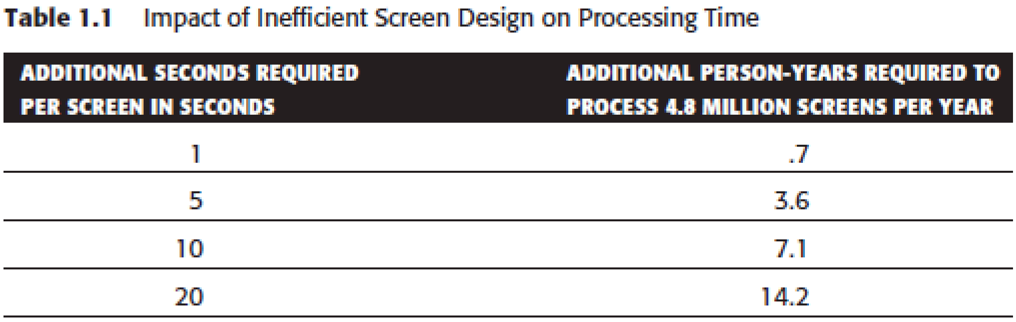


Table 1.1 Impact of Inefficient Screen Design on Processing Time

ADDITIONAL SECONDS REQUIRED PER SCREEN IN SECONDS	ADDITIONAL PERSON-YEARS REQUIRED TO PROCESS 4.8 MILLION SCREENS PER YEAR
1	.7
5	3.6
10	7.1
20	14.2

Studies have also shown that the proper formatting of information on screens does have a significant positive effect on performance. The benefits of a well-designed screen have also been under experimental scrutiny.
A general rule of thumb: every dollar invested in usability returns $10 to $100 (IBM, 2001).

2 a. Discuss the general principles of UID.

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

User compatibility. Design must be appropriate and compatible with the needs of the user or client. Effective design starts with understanding the user's needs and adopting the user's point of view. One very common error among designers is to assume that users are all alike. Task and job compatibility. The organization of a system should match the tasks a person must do to perform the job.

Product compatibility. The intended user of a new system is often the user of other systems or earlier versions of the new system. Habits, expectations, and a level of knowledge have been established and will be brought to bear when learning the new system.

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.

When an error does occur, provide constructive messages

Predictability:

o 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.

o All expectations should be fulfilled uniformly and completely

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

2 b. Mention the advantages and disadvantages of GUI in details.

Advantages

Symbols recognized faster than text : symbols can be recognized faster and more accurately than text, and that the graphical attributes of icons, such as shape and color, are very useful for quickly classifying objects, elements, or text by some common property.

Faster learning : pictorial representation aids learning, and symbols can also be easily learned.

Faster use and problem solving : Visual or spatial representation of information has been found to be easier to retain and manipulate and leads to faster and more successful problem solving. Symbols have also been found to be effective in conveying simple instructions.

Easier remembering: Because of greater simplicity, it is easier for casual users to retain operational concepts.

More natural: Graphic representations of objects are thought to be more natural and closer to innate human capabilities.

Exploits visual/spatial cues: Spatial relationships are usually found to be understood more quickly than verbal representations. Visually thinking is believed to be better than logical thinking.

Fosters more concrete thinking: Displayed objects are directly in the high-level task domain, or directly usable in their presented form. There is no need mentally to decompose tasks into multiple commands with complex syntactic form. The need for abstract thinking is therefore minimized.

Provides context: Displayed objects are visible, providing a picture of the current context.

Fewer errors: More concrete thinking affords fewer opportunities for errors. Reversibility of actions reduces error rates because it is always possible to undo the last step. Error messages are less frequently needed.

Increased feeling of control: The user initiates actions and feels in control. This increases user confidence and hastens system mastery.

Immediate feedback: The results of actions furthering user goals can be seen immediately.

Learning is quickened. If the response is not in the desired direction, the direction can be changed quickly.

Predictable system responses: Predictable system responses also speed learning.

Easily reversible actions: The user has more control. This ability to reverse unwanted actions also increases user confidence and hastens system mastery.

Less anxiety concerning use: Hesitant or new users feel less anxiety when using the system because it is so easily comprehended, is easy to control, and has predictable responses and reversible actions.

More attractive: Direct-manipulation systems are more entertaining, cleverer, and more appealing. This is especially important for the cautious or skeptical user. May consume less space: Icons may take up less space than the equivalent in words. More information can often be packed in a given area of the screen. This, however, is not always the case.

Replaces national languages: Language-based systems are seldom universally applicable.

Easily augmented with text displays: Where graphical design limitations exist, direct-manipulation systems can easily be augmented with text displays. The reverse is not true.

Low typing requirements: Pointing and selection controls, such as the mouse or trackball, eliminate the need for typing skills.

Smooth transition from command language system: Moving from a command language to a direct-manipulation system has been found to be easy. The reverse is not true.

Disadvantages

Greater design complexity : design potential may not necessarily result in better design, unless the choices are thoughtfully selected and consistently and simply applied. Proper window types must also be chosen and colors selected from a seemingly unending rainbow of alternatives. With graphics, the skill of the designer is increasingly challenged. Poor design can undermine acceptance.

Learning still necessary : The first time one encounters many graphical systems, what to do is not immediately obvious. The meanings of many words and icons may not be known. It is not often possible to guess their meanings, especially the more arbitrary symbols.

Lack of experimentally-derived design guidelines : The graphical interface is still burdened today by a lack of widely available experimentally-derived design guidelines.

Inconsistencies in technique and terminology: Many differences in technique, terminology, and look and feel exist among various graphical system providers, and even among successive versions of the same system. These inconsistencies occur because of copyright and legal implications, product differentiation considerations, and our expanding knowledge about the interface. The result is that learning, and relearning, for both designers and users is much more difficult than it should be.

Working domain is the present: While direct-manipulation systems provide context, they also require the user to work in the present.

Not always familiar : Symbolic representations may not be as familiar as words or Numbers.

Human comprehension limitations: Human limitations may also exist in terms of one’s ability to deal with the increased complexity of the graphical interface. The variety of visual displays can still challenge all but the most sophisticated users.

Window manipulation requirements: Window handling and manipulation times are still excessive and repetitive. This wastes time and interrupts the decision making needed to perform tasks and jobs.

Production limitations: The number of symbols that can be clearly produced using today's technology is still limited.

Few tested icons exist: Icons, like typefaces, must appear in different sizes, weights, and styles. As with text, an entire font of clearly recognizable symbols must be developed. Inefficient for touch typists: For an experienced touch typist, the keyboard is a very fast and powerful device. Moving a mouse or some other pointing mechanism may be slower. Inefficient for expert users. Inefficiencies develop when there are more objects and actions than can fit on the screen.

Not always the preferred style of interaction: Not all users prefer a pure iconic interface.

Not always fastest style of interaction: Another study has found that graphic instructions on an automated bank teller machine were inferior to textual instructions.

Increased chances of clutter and confusion: A graphical system does not guarantee elimination of clutter on a screen. Instead, the chance for clutter is increased, thereby increasing the possibility of confusion.

May consume more screen space: Not all applications will consume less screen space. A listing of names and telephone numbers in a textual format will be more efficient to scan than a card file.

Hardware limitations: Good design also requires hardware of adequate power, processing speed, screen resolution, and graphic capability. Insufficiencies in these areas can prevent a graphic system's full potential from being realized.

MODULE-2

3 a. What is requirement analysis? What are the methods involved in it? What is the impact of it on UI design?

A requirement is an objective that must be met. A product description is developed and refined, based on input from users or marketing.

Direct methods:

Individual Face-to-Face Interview

• A one-on-one visit with the user to obtain information. It may be structured or somewhat open-ended.

Telephone Interview or Survey

• A structured interview conducted via telephone.

Traditional Focus Group

• A small group of users and a moderator brought together to verbally discuss the requirements.

Facilitated Team Workshop

• A facilitated, structured workshop held with users to obtain requirements information. Similar to the Traditional Focus Group

Observational Field Study

• Users are observed and monitored for an extended time to learn what they do.

Requirements Prototyping

• A demo, or very early prototype, is presented to users for comments concerning functionality.

User-Interface Prototyping

• A demo, or early prototype, is presented to users to uncover user-interface issues and problems.

Usability Laboratory Testing

• Users at work are observed, evaluated, and measured in a specially constructed laboratory.

Card Sorting for Web Sites

• A technique to establish groupings of information for Web sites.

Indirect methods

MIS Intermediary

• A company representative defines the user’s goals and needs to designers and developers.

Paper Survey or Questionnaire

• A survey or questionnaire is administered to a sample of users using traditional mail methods to obtain their needs.

Electronic Survey or Questionnaire

• A survey or questionnaire is administered to a sample of users using e-mail or the Web to obtain their needs.

Electronic Focus Group

• A small group of users and a moderator discuss the requirements online using workstations.

Marketing and Sales

• Company representatives who regularly meet customers obtain suggestions or needs, current and potential.

Support Line

• Information collected by the unit that helps customers with day-to-day problems is analyzed

(Customer Support, Technical Support, Help Desk, etc.).

E-Mail or Bulletin Board

• Problems, questions, and suggestions from users posted to a bulletin board or through e-mail are analyzed.

User Group

• Improvements are suggested by customer groups who convene periodically to discuss software usage.

Competitor Analyses

• A review of competitor's products or Web sites is used to gather ideas, uncover design requirements and identify tasks.

Trade Show

• Customers at a trade show are presented a mock-up or prototype and asked for comments.

Other Media Analysis

• An analysis of how other media, print or broadcast, present the process, information, or subject matter of interest.

System Testing

• New requirements and feedback are obtained from ongoing product testing.

3 b. Define obstacles and pitfalls mention the general observation of design and common pitfalls and also explain five commandments used in Designing.

Obstacles and Pitfalls in the Development Path.
The path is littered with obstacles and traps, many of them human in nature.
Pitfalls are the common UI design problems. Gould (1988) has made these general observations about design:

- Nobody ever gets it right the first time.
- Development is chock-full of surprises.
- Good design requires living in a sea of changes.
- Making contracts to ignore change will never eliminate the need for change.
- Even if you have made the best system humanly possible, people will still make mistakes when using it.
- Designers need good tools.
- You must have behavioral design goals like performance design goals.

Common pitfalls are:

- No early analysis and understanding of the user's needs and expectations.
- A focus on using design features or components that are "neat" or "glitzy."
- Little or no creation of design element prototypes.
- No usability testing.

- No common design team vision of user interface design goals.
- Poor communication between members of the development team

Five commandments used in Designing.
Gain a complete understanding of users and their tasks: The users are the customers. people expect a level of design sophistication from all interfaces, including Web sites. The product, system or Web site must be geared to people’s needs, not those of the developers.

Solicit early and ongoing user involvement: Involving the users in design from the beginning provides a direct conduit to the knowledge they possess about jobs,tasks, and needs.
Perform rapid prototyping and testing: Prototyping and testing the product will quickly identify problems and allow you to develop solutions. Prototyping and testing must be continually performed during all stages of development to uncover all potential defects.

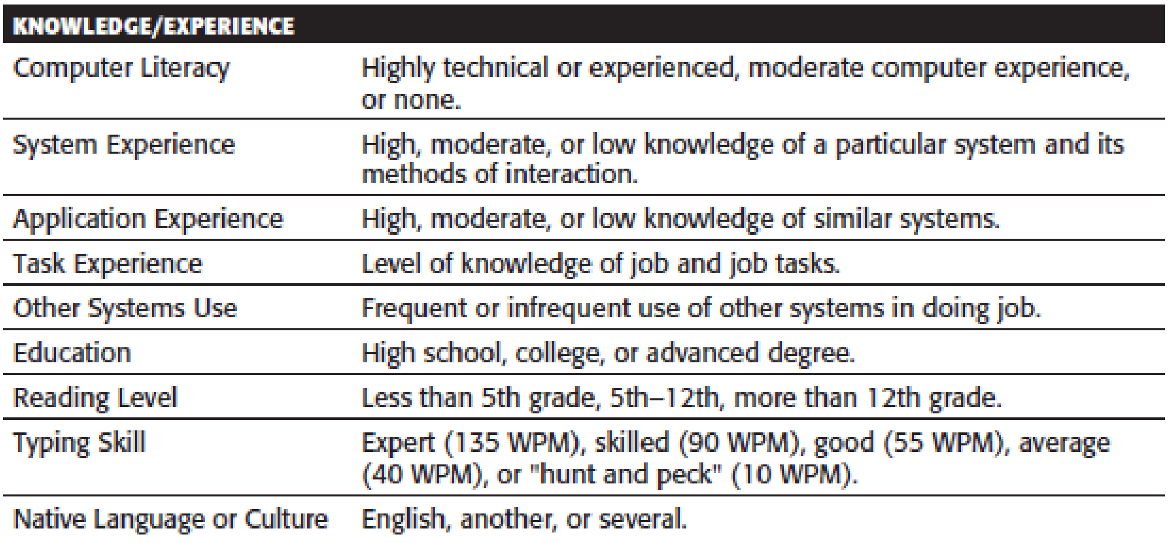
Modify and iterate the design as much as necessary: While design will proceed through a series of stages, problems detected in one stage may force the developer to revisit a previous stage. This is normal and should be expected. Establish user performance and acceptance criteria and continue testing and modifying until all design goals are met.

Integrate the design of all the system components: A system is being constructed, not simply software. Concurrent development of all pieces will point out possible problems earlier in the design process, allowing them to be more effectively addressed.

4 a. Explain the importance of human consideration in UI design with suitable example

1. The User’s Knowledge and Experience

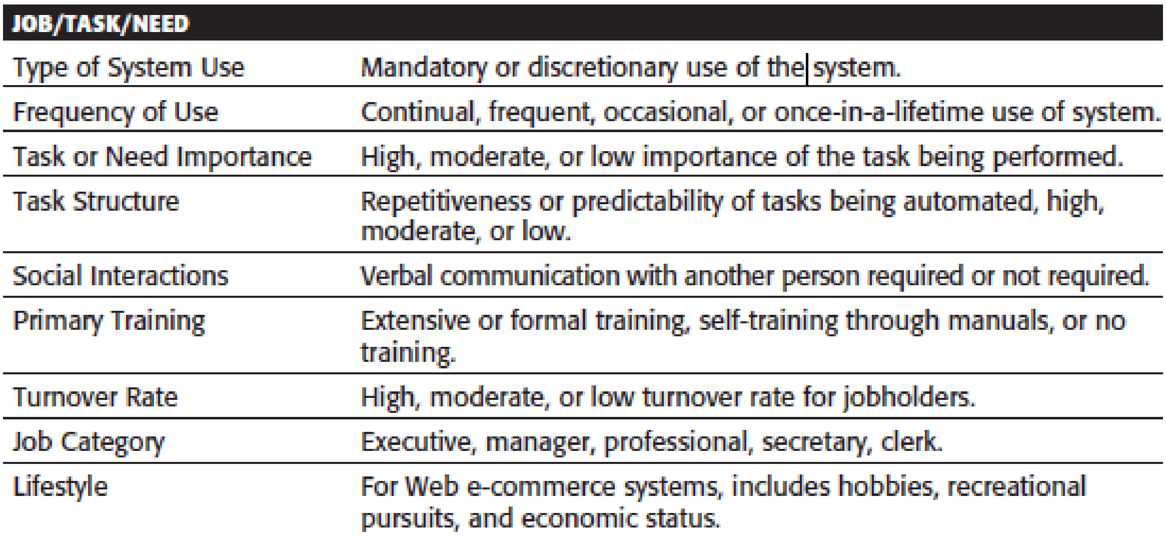
The knowledge possessed by a person, and the experiences undergone, shape the design of the interface in many ways. The following kinds of knowledge and experiences should be identified.



KNOWLEDGE/EXPERIENCE	
Computer Literacy	Highly technical or experienced, moderate computer experience, or none.
System Experience	High, moderate, or low knowledge of a particular system and its methods of interaction.
Application Experience	High, moderate, or low knowledge of similar systems.
Task Experience	Level of knowledge of job and job tasks.
Other Systems Use	Frequent or infrequent use of other systems in doing job.
Education	High school, college, or advanced degree.
Reading Level	Less than 5th grade, 5th–12th, more than 12th grade.
Typing Skill	Expert (135 WPM), skilled (90 WPM), good (55 WPM), average (40 WPM), or "hunt and peck" (10 WPM).
Native Language or Culture	English, another, or several.

2. The User’s Tasks and Needs

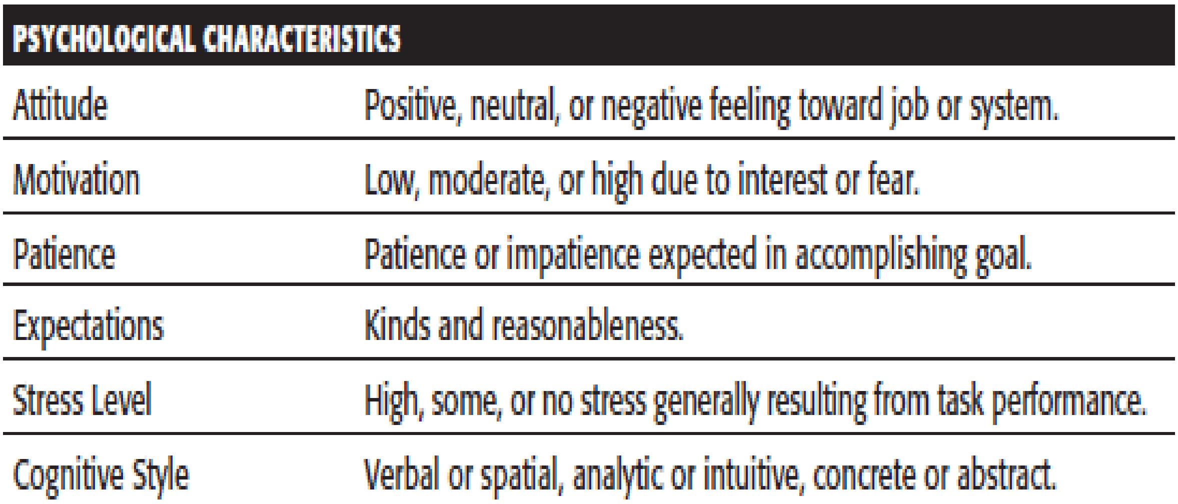
The user’s tasks and needs are also important in design. The following should be determined:



JOB/TASK/NEED	
Type of System Use	Mandatory or discretionary use of the system.
Frequency of Use	Continual, frequent, occasional, or once-in-a-lifetime use of system.
Task or Need Importance	High, moderate, or low importance of the task being performed.
Task Structure	Repetitiveness or predictability of tasks being automated, high, moderate, or low.
Social Interactions	Verbal communication with another person required or not required.
Primary Training	Extensive or formal training, self-training through manuals, or no training.
Turnover Rate	High, moderate, or low turnover rate for jobholders.
Job Category	Executive, manager, professional, secretary, clerk.
Lifestyle	For Web e-commerce systems, includes hobbies, recreational pursuits, and economic status.

3. The User’s Psychological Characteristics

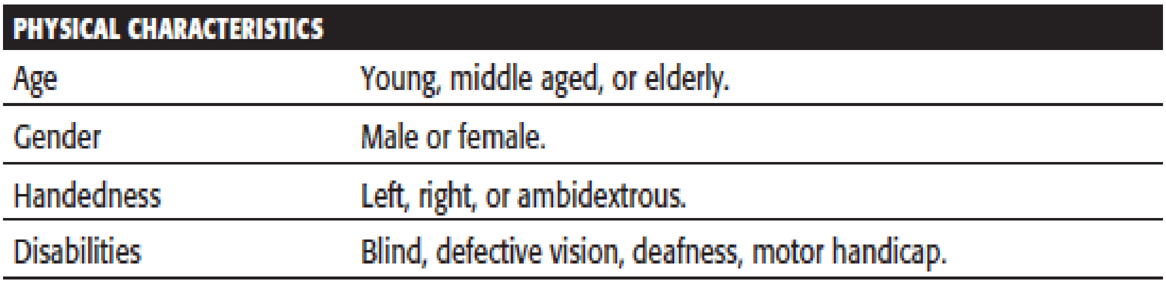
A person’s psychological characteristics also affect one’s performance of tasks requiring motor, cognitive, or perceptual skills.



PSYCHOLOGICAL CHARACTERISTICS	
Attitude	Positive, neutral, or negative feeling toward job or system.
Motivation	Low, moderate, or high due to interest or fear.
Patience	Patience or impatience expected in accomplishing goal.
Expectations	Kinds and reasonableness.
Stress Level	High, some, or no stress generally resulting from task performance.
Cognitive Style	Verbal or spatial, analytic or intuitive, concrete or abstract.

4. The User’s Physical Characteristics

The physical characteristics of people can also greatly affect their performance with a system.



PHYSICAL CHARACTERISTICS	
Age	Young, middle aged, or elderly.
Gender	Male or female.
Handedness	Left, right, or ambidextrous.
Disabilities	Blind, defective vision, deafness, motor handicap.

4 b. Explain briefly about human interaction speed.

Reading. The average adult, reading English prose in the United States, has a reading speed in the order of 250 to 300 words per minute. Proofreading text on paper has been found to occur at about 200 words per minute, on a computer monitor, about 180 words per minute (Ziefle, 1998). Nontraditional reading methods have also been explored in research laboratories. One technique that has dramatically increased reading speeds is called Rapid Serial Visual Presentation, or RSVP. In this technique single words are presented one at a time in the center of a screen. New words continually replace old words at a rate set by the reader. Bailey (1999a) tested this technique with a sample of people whose paper document reading speed was 342 words per minute (with a speed range of 143 to 540 words per minute). Single words were presented on a screen in sets at a speed sequentially varying ranging from 600 to 1,600 words per minute. After each set a comprehension test was administered.

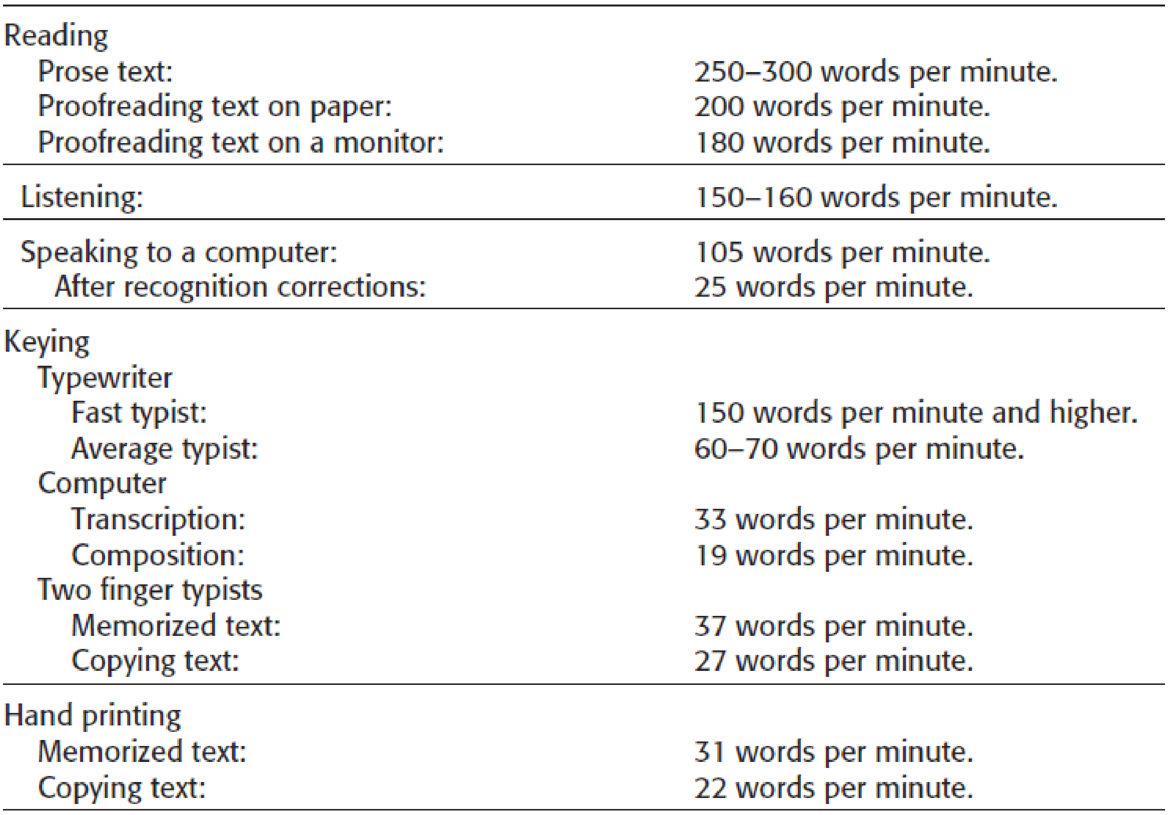
For measured comprehension scores of 75 percent or higher, the average reading speed was 1,212 words per minute. This is about 3.5 times faster than reading in the traditional way. Bailey concludes that computer technology can help improve reading speeds, but non-traditional techniques must be used. Listening. Words can be comfortably heard and understood at a rate of 150 to 160 words per minute. This is generally the recommended rate for audio books and video narration (Williams, 1998). Omoigui et al. (1999) did find, however, that when normal speech is speeded up using compression, a speed of 210 words per minute results in no loss of comprehension.

Speaking. Dictating to a computer occurs at a rate of about 105 words per minute (Karat et al., 1999; Lewis, 1999). Speech recognizer misrecognitions often occur, however, and when word correction times are factored in, the speed drops significantly to an average of 25 words per minute. Karat et al. (1999) also found that the speaking rate of new users was 14 words per minute during transcription and eight words per minute during composition.

Keying. Fast typewriter typists can key at rates of 150 words per minute and higher. Average typing speed is considered to be about 60 to 70 words per minute. Computer keying has been found to be much slower, however. Speed for simple transcription found by Karat et al. (1999) was only 33 words per minute and for composition only 19 words per minute. In this study, the fastest typists typed at only 40 words per minute, the slowest at 23 words per minute. Brown (1988) reports that two-finger typists can key memorized text at 37 words per minute and copied text at 27 words per minute. Something about the computer, its software, and the keyboard does seem to significantly degrade the keying process. (And two-finger typists are not really that bad off after all.)

Hand printing. People hand-print memorized text at about 31 words per minute. Text is copied at about 22 words per minute (Brown, 1988).

Table 1.3 Average Human Interaction Speeds



Reading	
Prose text:	250–300 words per minute.
Proofreading text on paper:	200 words per minute.
Proofreading text on a monitor:	180 words per minute.
Listening:	150–160 words per minute.
Speaking to a computer:	105 words per minute.
After recognition corrections:	25 words per minute.
Keying	
Typewriter	
Fast typist:	150 words per minute and higher.
Average typist:	60–70 words per minute.
Computer	
Transcription:	33 words per minute.
Composition:	19 words per minute.
Two finger typists	
Memorized text:	37 words per minute.
Copying text:	27 words per minute.
Hand printing	
Memorized text:	31 words per minute.
Copying text:	22 words per minute.

MODULE-3

5 a. Explain in brief the structure 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 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, as illustrated in Figure 4.1

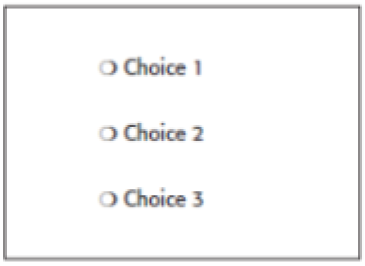


Figure 4.1

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 rentry 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. All the menus are important to the process at hand and must be answered in some manner by the user. A sequential linear menu is illustrated in Figure 4.2.

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.

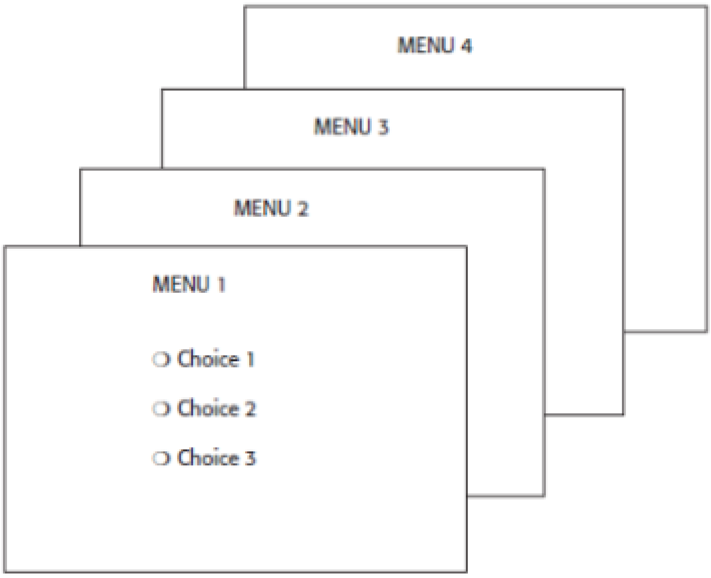


Figure 4.2 Sequential linear menus.

Simultaneous Menus

Instead of being presented on separate screens, all menu options are available simultaneously, as illustrated in Figure 4.3. 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.

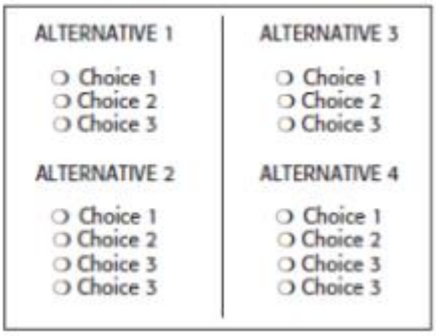


Figure 4.3 Simultaneous menus.

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 moves downward through it. Hierarchical structures are characterized by depth and breadth, depth being the number of choice levels one must traverse tor each the destination, breadth being the number of alternatives found at each level.

A hierarchical menu is illustrated in Figure 4.4. Note that the top level of the tree is considered level 0 with subsequent levels numbered sequentially beginning with number1. Starting at the top, level 0, two selections, or mouse clicks, are required to reach level 2.

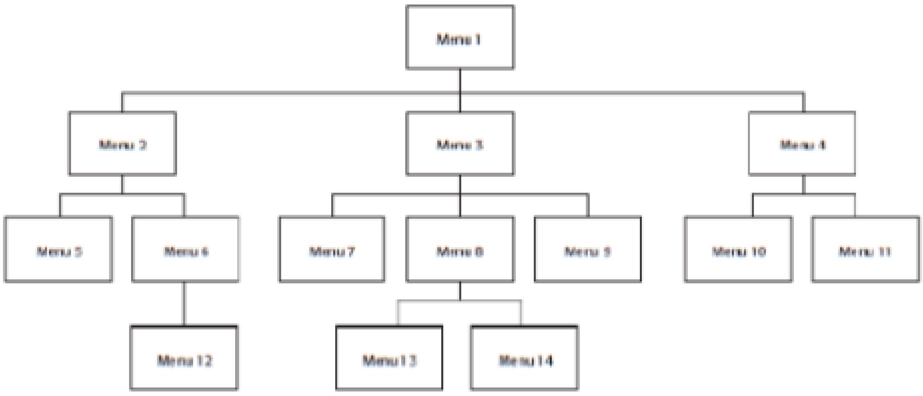


Figure 4.4 : A hierarchical menu

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. 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. (In a hierarchical menu system, the ability to go back to a previous menu or to return to the top-level menu are also examples, although restricted, of connected menus.)

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. An example connected menu structure is represented in Figure 4.5.

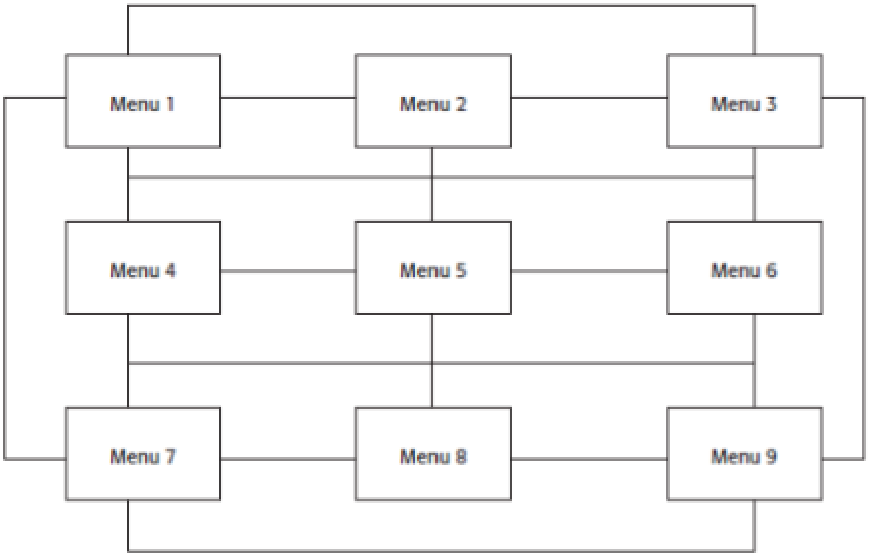


Figure 4.5 Connected menus.

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. They are, in essence, a set of simultaneous menus imposed on hierarchical menus. In a graphical system, for example, existing together are a simultaneous menu, the menu bar, and a hierarchy—the menu bar and its pull-downs. 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).

5 b. Describe the components of a web navigation system with illustration.

(a). *Browser Command Buttons:* Rather then relying on the browser's buttons, provide navigation controls within the application for movement within the application. They can take the form of links or command buttons such as *Next* and *Previous*.

(b). *Web Site Navigation Bars:* A Web site navigation bar is a menu, an array of textual phrases, graphical images or icons, or command buttons

- Provide a global navigation bar at the top of each page.

■ Provide a local category or topical links navigation bar on the left side of a page.

■ Place minor illustrative, parenthetical, or footnote links at the end of the page.

■ For long pages, provide a navigation bar repeating important global or local links at the page bottom.

- Figures 1, 2 and 3 provides a Web site contains at least three levels of navigation links, *global,* or site-wide, links, indicating the site's total scope or categories of available information.

- *Local* specific navigation links within the category or topical area being displayed and *minor* illustrative, parenthetical, or footnote links.

- Figure 4. shows an evolving standard in design is to locate the global links at the page top, the categorical links in a columnar array down the pages left side, and the minor links at the page bottom, making it easy for users to find each.

a. Textual Phrases: Textual phrases are words, or short pieces of highlighted text, serving as links. Textual phrase links possess two distinct structures.

(1) Explicit: An *explicit* menu is a listing of textual phrase links set apart from the main page content, often in a toolbar. A typical explicit menu is shown in Fig 1.

(2) Embedded: An *embedded* menu is a link contained within the textual content of a page. Certain words or phrases are designated as links, highlighted, and when selected display the linked component for the user.

b. Graphical Images or Icons: Graphical images or icons may appear in an array in the form of a navigation bar, or be individually located at relevant points in a page.

c. Command Buttons: Command buttons may appear in an array in the form of a navigation bar, or be individually located at relevant points in a page.

(c). *Other Web Site Navigation Elements:* In addition to Navigation bars, a number of other Web site elements are also important components of the Web navigation system.

- Among these are overviews, including executive summaries, site maps, indexes, and tables of contents.
- Other elements are historical rails and search engines.

<u>Overviews</u>

■ Provide:

— An executive summary that provides a preview of the site and contains links to all major concepts.

— A site map illustrating the site's hierarchical structure and th relationships of components.

— Both global and local maps.

— An alphabetized site index.

— A table of contents.

■ Allow accessibility from any point in the Web site.

Historical Trails:

■ Provide:

— Breadcrumb Trails.

- Locate at the top of the page below the navigation links.

— History Lists.

— History Trees.

— Footprints.

— Bookmarks.

Search Facility:

■ Provide a search facility: Another form of navigation support is provided by a site search facility. Provide such a facility within larger sites.

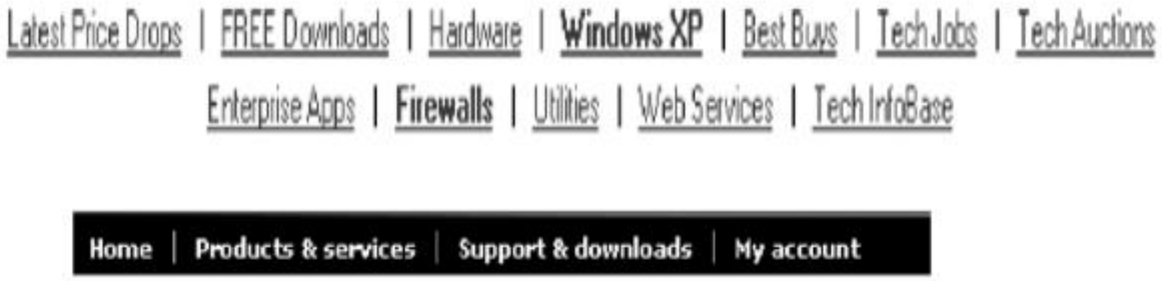
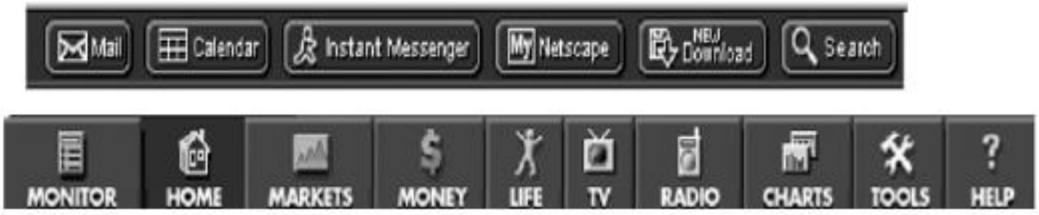

Latest Price Drops | FREE Downloads | Hardware | Windows XP | Best Buys | Tech Jobs | Tech Auctions
Enterprise Apps | Firewalls | Utilities | Web Services | Tech InfoBase
Home | Products & services | Support & downloads | My account

Figure 1. Textual explicit listing navigation bars.



Graphical or iconic navigation bars.

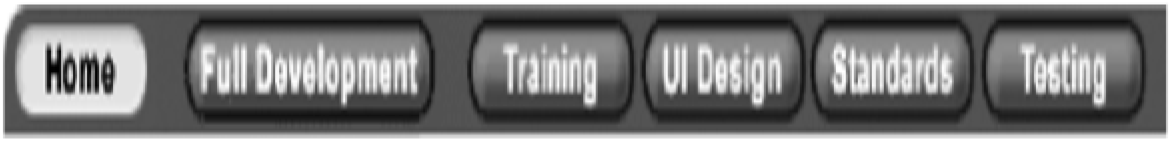


Figure 3. Command button navigation bar.

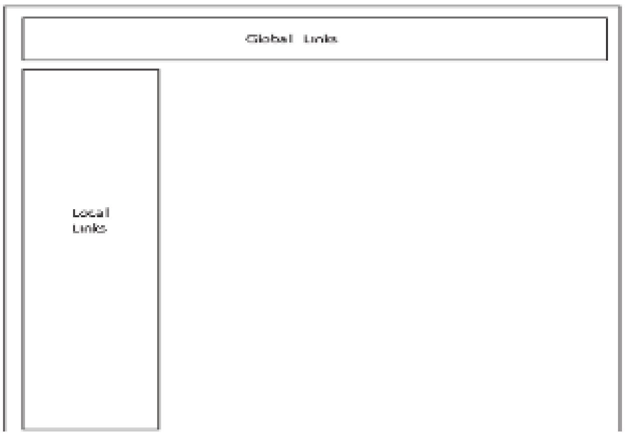


Figure 4. Web navigation component locations.

6 a. Write a note on Graphical menus for the following

i) 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.

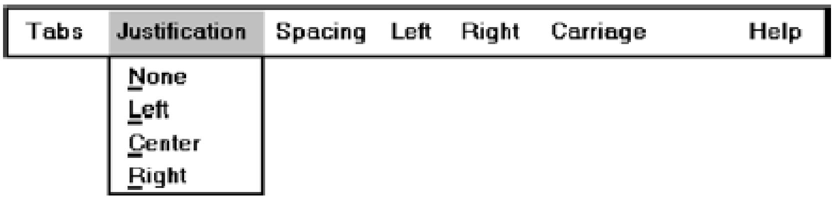


Figure 4.24 Menu bar pull-down.

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.

ii) Pop up menu

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

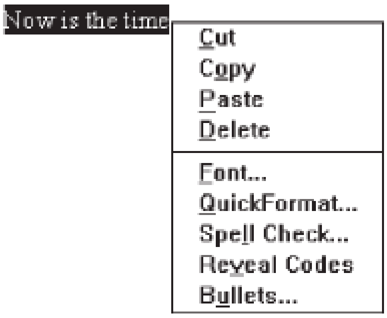


Figure 4.35 Pop-up menu.

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 offscreen 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.

6 b. Describe at least four guidelines to be followed in phasing of menu during the development of system menus.

Menu

• A menu must communicate to the user information about:

o The nature and purpose of the menu

itself.

o The nature and purpose of each presented choice.

o 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. Superfluous titles may be omitted.

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.

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.

MODULE-4

7 a. Discuss briefly about the types of windows with example. (Any five)

1. Primary windows

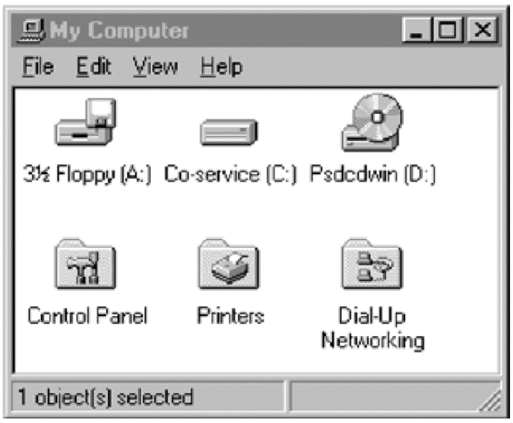


Figure 5.7 Microsoft Windows primary window.

■ 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.

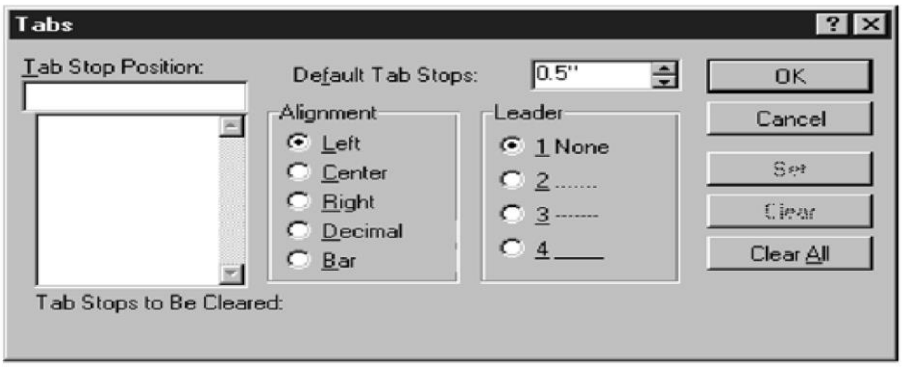
— Use for providing context for dependent windows to be created.

— Do not:

• Divide an independent function into two or more primary windows.

• Present unrelated functions in one primary window.

2. Secondary window


Default Tab Stops: 0.5"

Alignment: ● Left ○ Center ○ Right ○ Decimal ○ Bar

Leader: ● 1 None ○ 2 .... ○ 3 ------- ○ 4 ____

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).

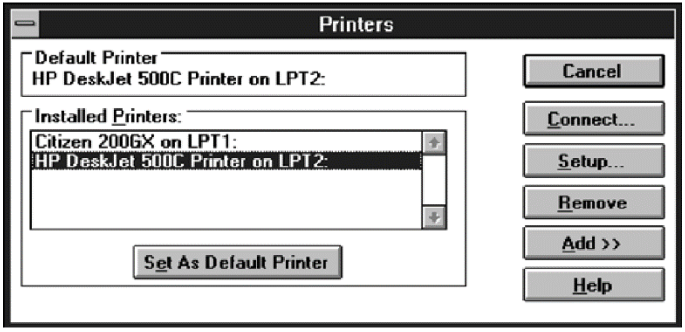
Use carefully because it constrains what the user can do.

■ 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*

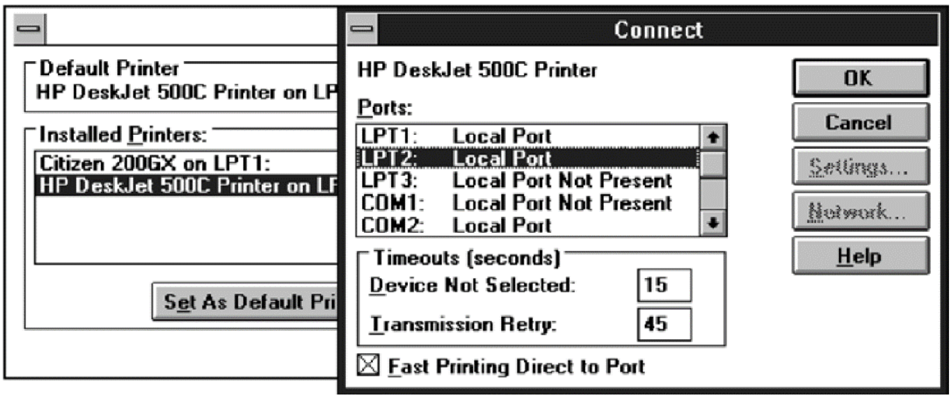


Printers

Default Printer
HP DeskJet 500C Printer on LPT2:

Installed Printers:
Citizen 200GX on LPT1:
HP DeskJet 500C Printer on LPT2:

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



Default Printer
HP DeskJet 500C Printer on LP

Installed Printers:
Citizen 200GX on LPT1:
HP DeskJet 500C Printer on LP

Connect

HP DeskJet 500C Printer

Ports:

LPT1: Local Port
LPT2: Local Port
LPT3: Local Port Not Present
COM1: Local Port Not Present
COM2: Local Port

Timeouts (seconds)
Device Not Selected: 15
Transmission Retry: 45

☒ Fast Printing Direct to Port

Figure 5.10 Cascading Connect secondary window.

■ Cascading:

Purpose:

To provide advanced options at a lower level in a complex dialog.

Guidelines:

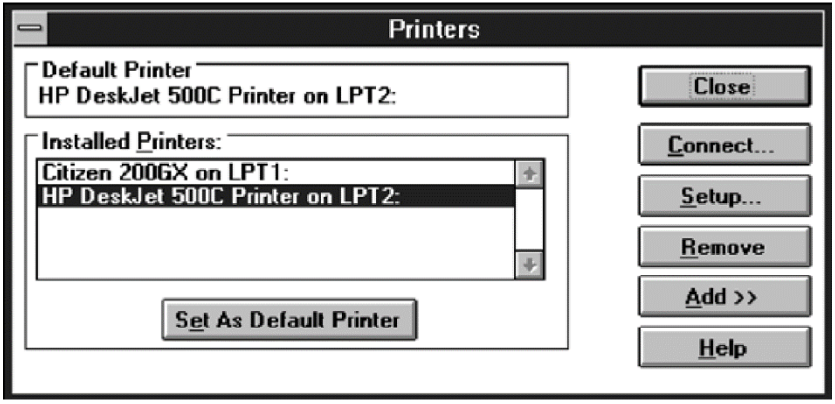
Provide a command button leading to the next dialog box with a "To a Window" indicator, an ellipsis (. . .).

- 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.

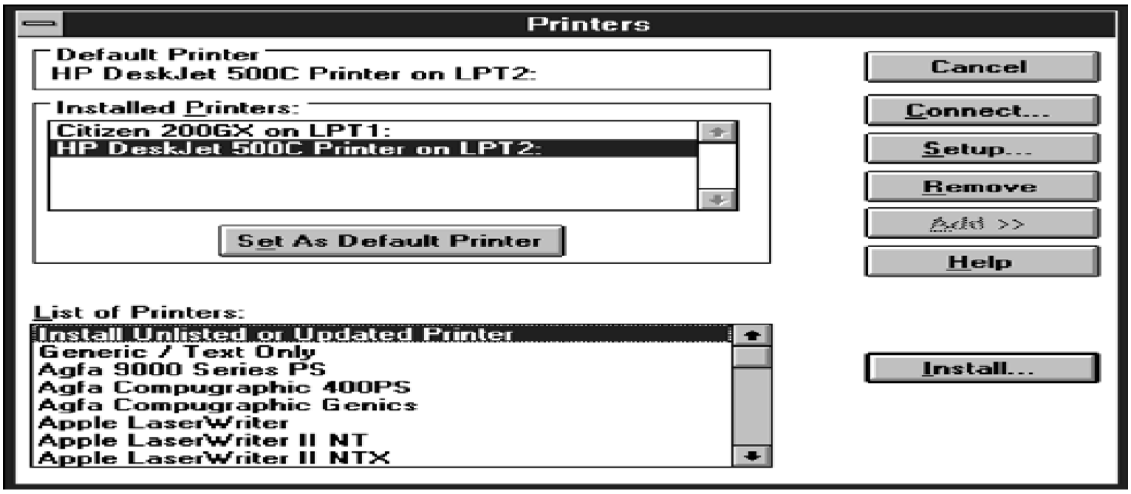


Printers

Default Printer
HP DeskJet 500C Printer on LPT2:

Installed Printers:
Citizen 200GX on LPT1:
HP DeskJet 500C Printer on LPT2:

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



Default Printer
HP DeskJet 500C Printer on LPT2:

Installed Printers:
Citizen 200GX on LPT1:
HP DeskJet 500C Printer on LPT2:

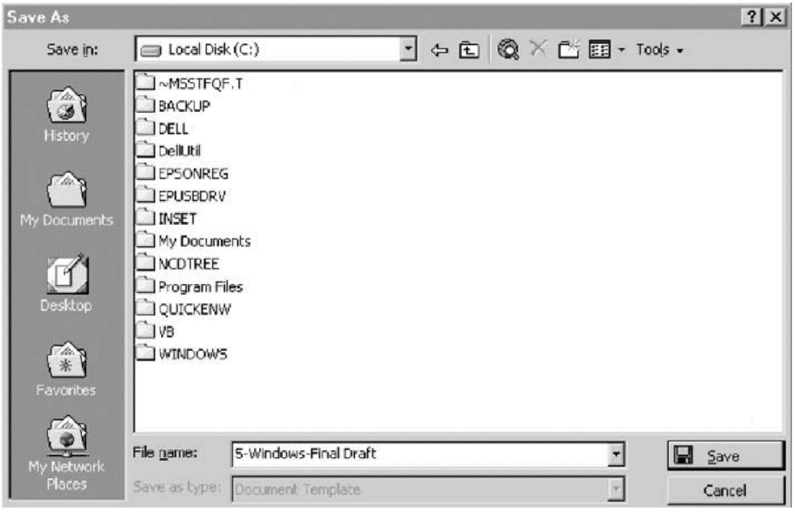
List of Printers:
Install Unlisted or Updated Printer
Generic / Text Only
Agfa 9000 Series PS
Agfa Compugraphic 400PS
Agfa Compugraphic Genics
Apple LaserWriter
Apple LaserWriter II NT
Apple LaserWriter II NTX

Figure 5.12 Unfolded Printers secondary window.

3. 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.



Save in: Local Disk (C:)

~MSSTFQF.T
BACKUP
DELL
DellUtil
EPSONREG
EPUSBDRV
INSET
My Documents
NCDTREE
Program Files
QUICKENW
VB
WINDOWS

File name: 5-Windows-Final Draft
Save as type: Document Template

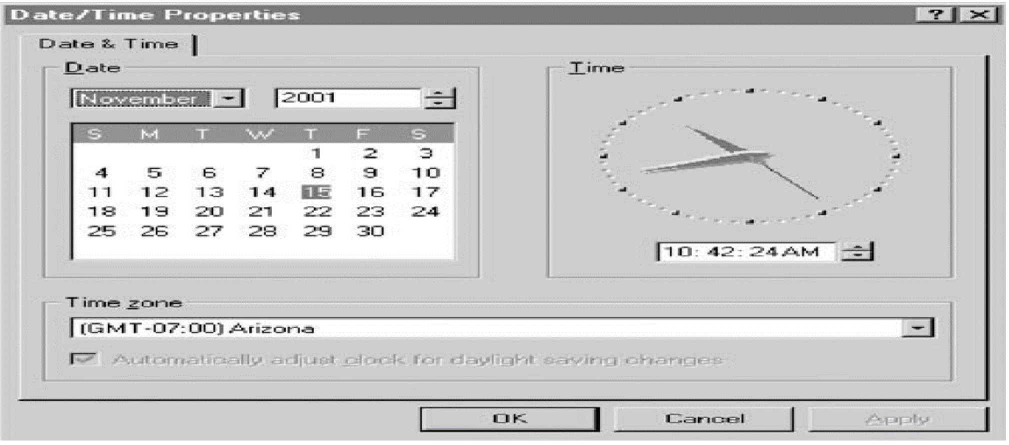
4. Property sheet

- ■ 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.



Date/Time Properties

Date & Time

Date: November 2001

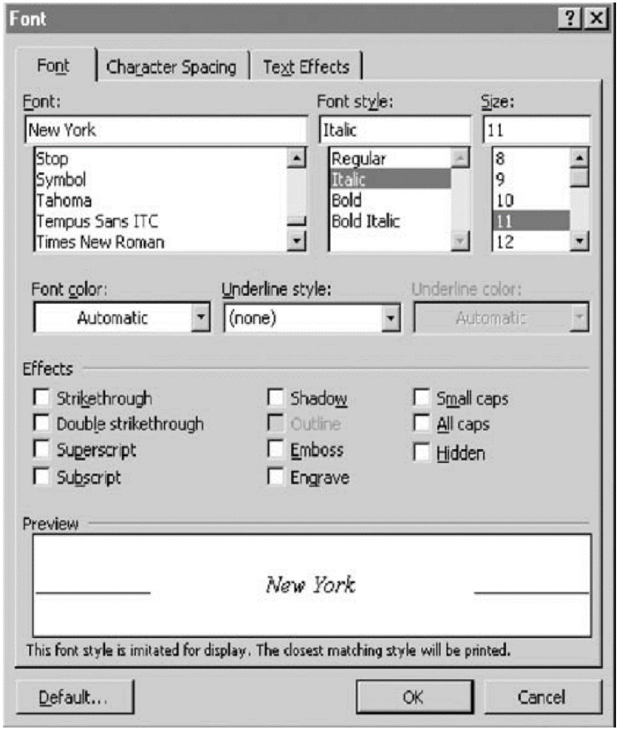
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Time: 10:42:24AM

Time zone: (GMT-07:00) Arizona

☑ Automatically adjust clock for daylight saving changes

Figure 5.14 Microsoft Windows property sheet.



Font

Font | Character Spacing | Text Effects

Font: New York

Stop
Symbol
Tahoma
Tempus Sans ITC
Times New Roman

Font style: Italic

Regular
Italic
Bold
Bold Italic

Size: 11

8
9
10
11
12

Font color: Automatic

Underline style: (none)

Underline color: Automatic

Effects

□ Strikethrough □ Shadow □ Small caps
□ Double strikethrough □ Outline □ All caps
□ Superscript □ Emboss □ Hidden
□ Subscript □ Engrave

Preview

New York

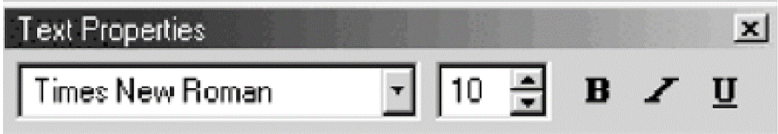
This font style is imitated for display. The closest matching style will be printed.

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.



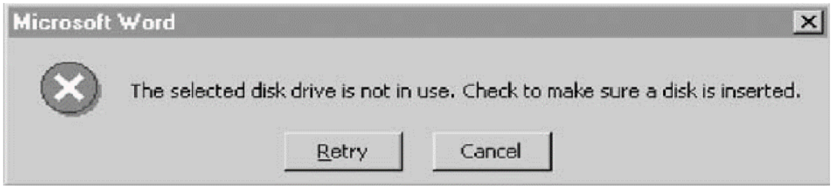
Text Properties

Times New Roman 10

Figure 5.16 Microsoft Windows property inspector.

5. 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.



Microsoft Word

The selected disk drive is not in use. Check to make sure a disk is inserted.

6. Palette Windows

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

Alternately, design them as fixed in size.

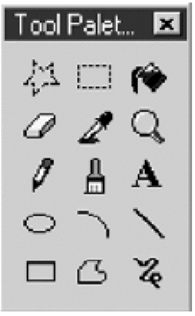


Figure 5.18 Microsoft Windows palette window.

7. 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.

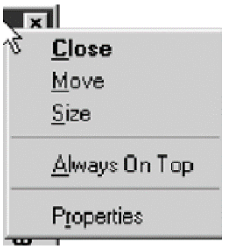


Figure 5.19 Microsoft Windows pop-up window.

7 b. Write a note on the following

1)**Track ball**

— A spherical object (ball) that rotates freely in all directions in its socket.

— Direction and speed is tracked and translated into cursor movement.

■ Advantages:

— Direct relationship between hand and pointer movement in terms of direction and speed.

— Does not obscure vision of screen.

— Does not require additional desk space (if mounted on keyboard).

■ Disadvantages:

— Movement is indirect, in a plane different from the screen.

— No direct relationship exists between hand and pointer movement in terms of distance.

— Requires a degree of eye-hand coordination.

— Requires hand to be removed from keyboard keys.

— Requires different hand movements.

— Requires hand to be removed from keyboard (if not mounted on keyboard).

— Requires additional desk space (if not mounted on keyboard).

— May be difficult to control.

— May be fatiguing to use over extended time.

2)Joystick

— A stick or bat-shaped device anchored at the bottom.

— Variable in size, smaller ones being operated by fingers, larger ones requiring the whole hand.

— Variable in cursor direction movement method, force joysticks respond to pressure, movable ones respond to movement.

— Variable in degree of movement allowed, from horizontal-vertical only to continuous.

■ Advantages:

— Direct relationship between hand and pointer movement in terms of direction.

— Does not obscure vision of screen.

— Does not require additional desk space (if mounted on keyboard).

■ Disadvantages:

— Movement indirect, in plane different from screen.

— Indirect relationship between hand and pointer in terms of speed and distance.

— Requires a degree of eye-hand coordination.

— Requires hand to be removed from keyboard keys.

---Requires different hand movements to use.

— Requires hand to be removed from keyboard (if not mounted on keyboard).

— Requires additional desk space (if not mounted on keyboard).

— May be fatiguing to use over extended time.

— May be slow and inaccurate.

8 a. Explain briefly about window management.

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

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-centred 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:

A workbook is a scheme for managing a set of views that uses the metaphor of a book or notebook. Within the workbook, views of objects, in the form of sections, are presented within the workbook’s primary window, rather than in individual child windows. Tabs are used as a navigational interface to move between different sections.

Description:

— A window or task management technique that consists of a set of views organized like a tabbed notebook.

— It is based upon the metaphor of a book or notebook.

— Views of objects are presented as sections within the workbook’s primary windows; child windows do not exist.

— Each section represents a view of data.

— Tabs can be included and used to navigate between sections.

— Otherwise, its characteristics and behavior are similar to those of the multiple document interface with all child windows maximized.

■ Proper usage:

— To manage a set of views of an object.

— To optimize quick navigation of multiple views.

— For content where the order of the sections is significant.

■Advantages:

— Provides a grouping and focus for a set of activities within the larger environment of the desktop.

— Conserves screen real estate.

— Provides the greater simplicity of the single-document window interface.

— Provides greater simplicity by eliminating child window management.

— Preserves some management capabilities of the multiple-document interface.

■ Disadvantage:

— Cannot present simultaneous views.

Projects:

A project is similar to a multiple-document interface (MDI), but does not visually contain the child windows. Objects represented by icons contained within it can be opened into primary windows that are peers with the parent window. Opened peer windows in the project do not share the menu bar or other areas contained with the parent window.

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.

8 b. Write a note on components of a windows.

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.
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.

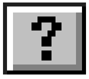


Figure 5.2 What's This? button.

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.

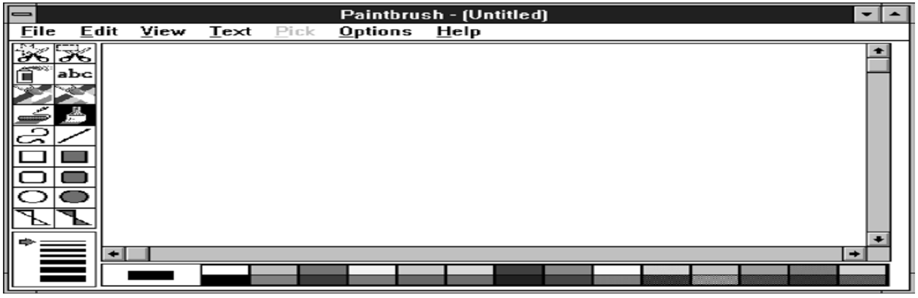
9. Split Box

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.

MODULE-5

9 a. Explain briefly the following selection control

i) Radio buttons

■ Description:

— A two-part control consisting of the following:

• Small circles, diamonds, or rectangles.

• Choice descriptions.

— When a choice is selected:

• The option is highlighted.

• Any existing choice is automatically unhighlighted and deselected.

■ Purpose:

— To set one item from a small set of mutually exclusive options (2 to 8).

■ Advantages:

— Easy-to-access choices.

— Easy-to-compare choices.

— Preferred by users.

■ Disadvantages:

— Consume screen space.

— Limited number of choices.

■ Proper usage:

— For setting attributes, properties, or values.

— For mutually exclusive choices (that is, only one can be selected).

— Where adequate screen space is available.

— Most useful for data and choices that are:

• Discrete.

• Small and fixed in number.

• Not easily remembered.

• In need of a textual description to meaningfully describe the alternatives.

• Most easily understood when the alternatives can be seen together and compared to one another.

• Never changed in content.

— Do not use:

• For commands.

• Singly to indicate the presence or absence of a state.

■ Choice Descriptions

Provide meaningful, fully spelled-out choice descriptions clearly describing the values or effects set by the radio buttons.

- Display in a single line of text.
- Display using mixed-case letters, using the sentence style.
- Position descriptions to the right of the button. Separate them by at least one space from the button.

■ When a choice is conditionally unavailable for selection, display the choice description grayed out or dimmed.

■ Include a None choice if it adds clarity.

ii) Checkboxes

■ Description:

— A two-part control consisting of a square box and choice description.

— Each option acts as a switch and can be either “on” or “off.”

• When an option is selected (on), a mark such as an “X” or “check” appears within the square box, or the box is highlighted in some other manner.

• Otherwise the square box is unselected or empty (off).

— Each box can be:

• Switched on or off independently.

• Used alone or grouped in sets.

■ Purpose:

— To set one or more options as either on or off.

■ Advantages

— Easy-to-access choices.

— Easy-to-compare choices.

— Preferred by users.

■ Disadvantages:

— Consume screen space.

— Limited number of choices.

— Single check boxes difficult to align with other screen controls.

■ Proper usage:

— For setting attributes, properties, or values.

— For nonexclusive choices (that is, more than one can be selected).

— Where adequate screen space is available.

— Most useful for data and choices that are:

• Discrete.

• Small and fixed in number.

• Not easily remembered.

• In need of a textual description to describe meaningfully.

• Most easily understood when the alternatives can be seen together and compared to one another.

Never changed in content.

— Can be used to affect other controls.

— Use only when both states of a choice are clearly opposite and unambiguous.

9 b. Explain the purpose of prototypes. Discuss any two kinds of prototypes with their importance to the system developers.

A prototype is primarily a vehicle for exploration, communication, and evaluation. Its purpose

is to obtain user input in design, and to provide feedback to designers. Its major function is the communicative role it plays, not accuracy or thoroughness. A prototype enables a design to be better visualized and provides insights into how the software will look and work. It also aids in defining tasks, their flow, the interface itself, and its screens.

Programmed Facades

■ Description:

— Examples of finished dialogs and screens for some important aspects of the system.

— Created by prototyping tools.

— Medium-fidelity to high-fidelity prototypes.

■ Advantages:

— Provide a good detailed specification for writing code.

— A vehicle for data collection.

■ Disadvantages:

— May solidify the design too soon.

— May create the false expectation that the “real thing” is only a short time away.

— More expensive to develop.

— More time-consuming to create.

— Not effective for requirements gathering.

— Not all of the functions demonstrated may be used because of cost, schedule limitations, or lack of user interest.

— Not practical for investigating more than two or three approaches.

Prototype-Oriented Languages

■ Description:

— An example of finished dialogs and screens for some important aspects of the system.

— Created through programming languages that support the actual programming process.

— A high-fidelity prototype.

■ Advantages:

— May include the final code.

— Otherwise, generally the same as those of programmed facades.

■ Disadvantages:

— Generally the same as for programmed facades

10 a. Explain the following with respect to kinds of Tests.

i) Think - Aloud Evaluation

■ Description:

— Users perform specific tasks while thinking out load.

— Comments are recorded and analyzed.

■ Advantages:

— Utilizes actual representative tasks.

— Provides insights into the user’s reasoning.

■ Disadvantages:

— May be difficult to get users to think out loud.

■ Guidelines:

— Develop:

• Several core or representative tasks.

• Tasks of particular concern.

— Limit session to 60 to 90 minutes.

ii) Usability Test

■ Description:

— An interface evaluation under real-world or controlled conditions.

— Measures of performance are derived for specific tasks.

— Problems are identified.

■ Advantages:

— Utilizes an actual work environment.

— Identifies serious or recurring problems.

■ Disadvantages:

— High cost for establishing facility.

— Requires a test conductor with user interface expertise.

— Emphasizes first-time system usage.

— Poorly suited for detecting inconsistency problems.

10 b. Explain the types of presentation control.

Presentation controls are purely informational. They provide details about other screen elements or controls, or assist in giving the screen structure. Common presentation controls are

1. Static Text Fields

■ Description:

— Read-only textual information.

■ Purpose:

— To identify a control by displaying a control caption.

— To clarify a screen by providing instructional or prompting information.

— To present descriptive information.

■ Proper usage:

— To display a control caption.

— To display instructional or prompting information.

— To display descriptive information.

Caption:
HEADING
This message is very important!

Figure 7.84 Static text field.

Static Text Field Guidelines
■ Captions:
— Include a colon (:) as part of the caption.
— Include a mnemonic for keyboard access.
— When the associated control is disabled, display it dimmed.
— Follow all other presented guidelines for caption presentation and layout.
■ Instructional or prompting information:
— Display it in a unique and consistent font style for easy recognition and differentiation.
— Follow all other presented guidelines for prompting and instructional information.
■ Descriptive information:
— Follow all other guidelines for required screen or control descriptive information.

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

3. Column Headings
■ Description:
— Read-only textual information that serves as a heading above columns of text or numbers.
— Can be divided into two or more parts.
■ Purpose:
— To identify a column of information contained in a table.
■ Proper usage:

— To display a heading above a column of information contained in a table.

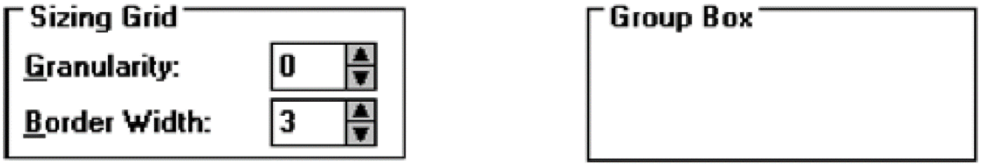
■ Guidelines:

— Heading:

- Provide a brief heading.
- Can include text and a graphic image.
- Capitalize the first letter of each significant word.
- Do not include an ending colon (:).

— The width of the column should fit the average size of the column entries.

— Does not support keyboard access.



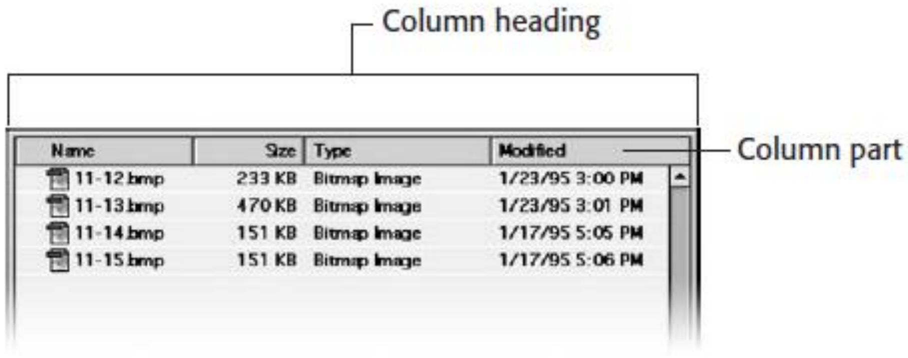
Sizing Grid

Granularity: 0

Border Width: 3

Group Box

Fig : Group Boxes



Name	Size	Type	Modified
11-12.bmp	233 KB	Bitmap Image	1/23/95 3:00 PM
11-13.bmp	470 KB	Bitmap Image	1/23/95 3:01 PM
11-14.bmp	151 KB	Bitmap Image	1/17/95 5:05 PM
11-15.bmp	151 KB	Bitmap Image	1/17/95 5:06 PM

Figure 7.86 Column heading control.

4. ToolTips

■ Description:

— A small pop-up window containing descriptive text that appears when a pointer is moved over a control or element either:

- Not possessing a label.
- In need of additional descriptive or status information.

■ Purpose:

— To provide descriptive information about a control or screen element.

■ Advantages:

— Identifies an otherwise unidentified control.

— Reduces possible screen clutter caused by control captions and descriptive information.

— Enables control size to be reduced.

■ Disadvantages:

— Not obvious, must be discovered.

— Inadvertent appearance can be distracting.

■ Proper usage:

— To identify a control that has no caption.

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

ToolTip Guidelines

■ Display after a short time-out.

■ For toolbars, provide a brief word as a label.

— Use mixed case in the headline style of presentation with no ending punctuation.

■ For other elements, provide a brief phrase presenting descriptive or status information.

— Use mixed case in the sentence style of presentation.

■ Present ToolTips at the lower-right edge of the pointer.

— Display them fully on the screen.

— For text boxes, display ToolTips centered under the control.

■ Display them in the standard system ToolTip colors.

■ Remove the ToolTip when the control is activated or the pointer is moved away.

■ Don’t substitute ToolTips for good design.

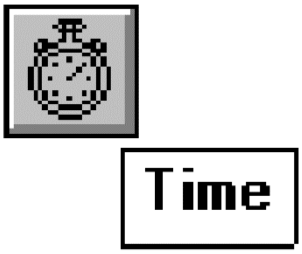


Fig : Tool Tip

5. Balloon Tips

■ Description:

— A small pop-up window that contains information in a word balloon.

— Components can include:

• Title.

• Body text.

• Message Icons.

— Appear adjacent to the item to which they apply, generally above or to left.

— Only one tip, the last posted, is visible at any time.

— Tips are removed after a specified time period.

■ Purpose:

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

■ Advantages:

— Provides useful reminder and status information.

■ Disadvantages:

— If overused they lose their attention-getting value.

— If overused in situations the user considers not very important, their continual appearance can be aggravating.

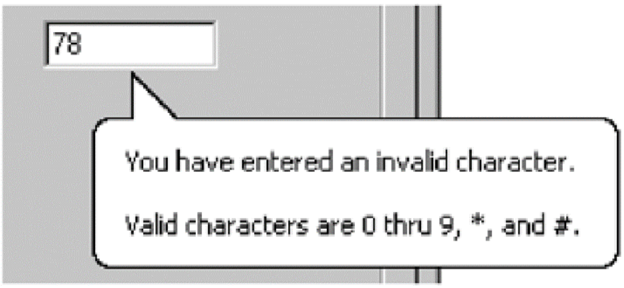
■ Proper usage:

— To display noncritical:

• Reminder information.

• Notification information.

— Do not use tips to display critical information.



78

You have entered an invalid character.

Valid characters are 0 thru 9, *, and #.

Fig:Balloon Tip

Balloon Tip Guidelines

■ General:

— Use a notification tip to inform the user about state changes.

— Use a reminder tip for state changes that the user might not usually notice.

— Point the tip of the balloon to the item it references.

— Do not use them to replace ToolTips.

— Do not overuse balloon tips.

■ Content:

— Restrict them to a length of 100 characters, including title and body text.

— Title text should:

• If the tip refers to an icon or other image representing a specific object, include:

— The object's name, using its normal capitalization.

— The object's status, using sentence-style presentation without ending punctuation.

• Be presented in bold.

— Body text should:

• Include a description of the situation in one or two brief sentences.

• Include a brief suggestion for correcting the situation.

• Be presented using mixed-case in the sentence style.

6. Progress Indicators

■ Description:

— A rectangular bar that fills as a process is being performed, indicating the percentage of the process that has been completed.

■ Purpose:

— To provide feedback concerning the completion of a lengthy operation.

■ Proper usage:

— To provide an indication of the proportion of a process completed.

Progress Indicator Guidelines

■ When filling the indicator:

— If horizontally arrayed, fill it from left to right.

— If vertically arrayed, fill it from bottom to top.

■ Fill it with a color or a shade of gray.

■ Include descriptive text for the process, as necessary.

■ Place text outside of the control.

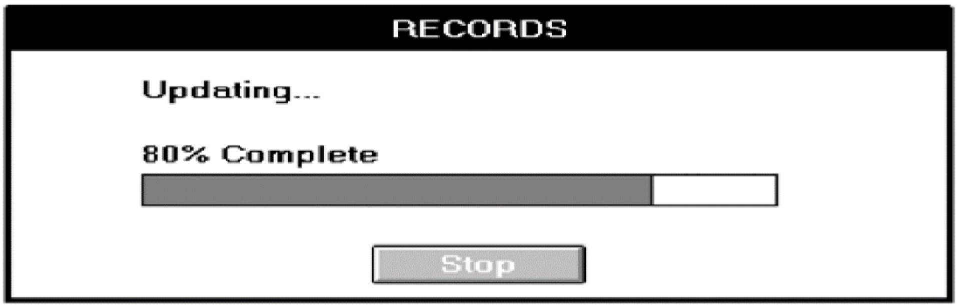


Fig : Progress Bar

Sample Box

■ Description:

— A box illustrating what will show up on the screen based upon the parameter or parameters selected.

— May include text, graphics, or both.

■ Purpose:

— To provide a representation of actual screen content based upon the parameter or parameters selected.

■ Guidelines:

— Include a brief label.

— Use mixed case in the headline style.

— Locate it adjacent to the controls upon which it is dependent.

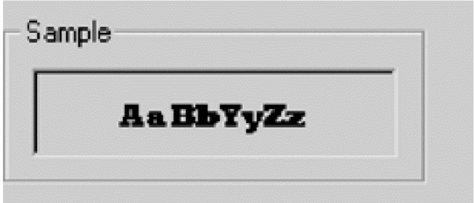


Fig : Sample Box

Scrolling Tickers

■ Description:

— Text that scrolls horizontally through a container window.

■ Advantages:

— Consume less screen space than full text.

■ Disadvantages:

— Hard to read.

— Time-consuming to interpret.

— Distracting.

■ Guideline:

— Do not use.