PC MAINTENANCE, TROUBLESHOOTING & REPAIRS

(AN ENGINEER'S PERSPECTIVE)

By

NWANI SUNDAY
PGD/MCS/2000/2001/1002

A project submitted to the Department of Mathematics/Computer Science, Federal University of Technology, Minna Niger State In partial fulfillment of the requirement for the award of a Post-Graduate Diploma in Computer Science

APRIL, 2002.

CERTIFICATION

This project work has been examined and found acceptable in partial fulfillment of the requirement for the Post Graduate Diploma in Computer Science of the Department of Mathematics/Computer Science of the Federal University of Technology, Minna.

....

Mr. L. N. Ezeako **Date**
(Project Supervisor)

....

Mr. L. N. Ezeako **Date**
(Head of Department)

....

External Examiner **Date**

DEDICATION

This project work is dedicated to God Almighty for His grace and mercies throughout the period of this programme.

ACKNOWLEDGEMENT

My heart felt gratitude goes to the Lord Almighty for His Love and care especially in this project.

My profound appreciation to my project supervisor, Mr. L. N. Ezeako for his unequalled support and fatherly care in making sure that this project work becomes a reality.

I am deeply grateful to all the lecturers, the entire staff and fellow students of Mathematics/ Computer Science Department of the Federal University of Technology, Minna.

Also, use this medium to express my sincere gratitude to my former boss in person of Engr. Peter Adaji, Director AMI-TECH Computers Ltd, Minna for his patience, candid advice and support throughout the duration of this programme.

Special thanks goes to Mr. Hakimi Danladi lecturer in the Department of Maths/Computer. God used him to open my eyes to start this postgraduate diploma in computer science. God will surely reward you.

My thanks is also extended to Miss Matemi Yemisi Faith and the entire staff of God's Time Computers for your un-quantifiable assistance in this project work. I called you when it mattered most and you did not fail me. Remain blessed.

And finally, my thanks go to anyone who had assisted me in one way or the other in this project work. God will surely reward you all. (Amen).

TABLE OF CONTENTS

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

ABSTRACT

Generally speaking, electronic equipment is accompanied by service manuals provided by the manufacturer. These manuals are expected to provide solutions to many problems that may arise in the cause of using the equipment. Personal computers are one of the latest developments in the world of electronics. But more often than not, these manuals provide little or no information on what to do when a fault occurs in the personal computer (PC). The consequences are better imagined than observed.

This project work is aimed at understanding the faults in PC, provision of an unambiguous step-by-step troubleshooting and how to carry out an effective repair/maintenance with minimum delay.

CHAPTER ONE

1.0 PRELIMINARIES

1.1 INTRODUCTION

Of all discoveries and inventions of the 21st century, computers standout in terms of sophistication, speed, quality of work done, efficiency and accuracy. Suffice to say that computers (Personal Computers) have been found useful and indispensable in various fields of human endeavour such as medical diagnosis, scientific research, military operations, banking institutions, educational systems etc. This write-up is therefore intended to discuss and analyse personal computers so as to understand how to maintain them, troubleshoot (find the cause of a fault) and carry-out an effective repair. At this point it should be noted that personal computers repair goes beyond swapping of boards (modular repairs). A proper understanding includes a sound knowledge of the hardware, software and a set of diagnostic tools. It is also an open secret that an effective maintenance, troubleshooting and repair culture will keep personal computers in a proper working condition.

It is believed that computer users, system engineers and other professionals will find this work useful.

1.2 DEFINITION OF BASIC TERMS USED

1. BACKUP: A copy of system file or personal file created, which can be used in case of damage to the original copy.
2. CMOS RAM: Complementary metal oxide semiconductor. It is used to save the configuration of selling each time the computer starts up.
3. BOOTING: The process of starting a PC.

4. VDU (Visual Display Unit): This is another name for the monitor. It is an output device that displays the result of every operation in the computer.
5. AC: - Alternating current i.e. current flow from a public power supply such as NEPA.
6. D.C: Direct Current i.e. current from some other source e.g. a battery.
7. CD-ROM: - Compact disk read only memory used for playing music or installation of programs in a CD.
8. DVD : Digital versatile disk. It is used to play music plus video.
9. Modem: Modulator-de modulator. It converts digital signals into analogue signals and vice versa. It is a major hardware used in INTERNET technology.
10. BIOS: - Basic input /output system. It forms a powerful link between the microprocessor and other component on the motherboard.
11. PCI : Peripheral component interconnect. This is a bus used for digital modules.
12. ISA (Industry standard architecture): A bus used for analogue modules.
13. SIMM: Single in the memory modules.
14. DIMM: Dual- in-line memory modules.
15. POST: Power On Self Test, A start up text carried out by the PC to check to check the presence or absence of some devices in a PC during booting process. Some beep codes could be heard if a devices is absent or not connected e.g. RAM or monitor.
16. Pixels: - Picture elements. A kind of representation of the trio dots- one for each of the primary colours of red, blue and green in monitors.

17. Hz (hertz): Number of cycles per second.

18. MHz (mega Hertz): A million cycles per second.

19. GHz (Giga Hertz): 1000 million cycles per second

1.3 AN OVERVIEW OF COMPUTER HARDWARE

The computer hardware comprises of pieces of components/devices that forms the computer system. These components/devices work with software (computer program) so as to give a meaningful and useful result to the user.

The major units or aspects of the hardware include:

(a) The monitor or VDU (Visual Display Unit)

(b) System Unit, which also contains the following.

(i) Motherboard

(ii) Hard Disk Drive (HDD)

(iii) Floppy Disk Drive (FDD)

(iv) Power Supply Unit (PSU)

(c) Keyboard

(d) Mouse and other necessary peripherals.

Detailed understanding and analysis of each of these individual components/parts will be handled in the next chapter.

Also, the above computer hardware and other peripherals could be stand-alone PC or can be connected to work with other computers (in what is popularly called a network).

We will concentrate on the standalone PC 's for this project work.

CHAPTER TWO

2.0 ISSUES OF MAINTENANCE

2.1 THEORY AND ANALYSIS OF SYSTEM MAINTENANCE

There is no magic prescription for fixing broken computers, though just plodding, methodical analysis to eliminate what is working right until what's left is the problem. Some involve stripping all the data and programs off the disk and starting over from scratch: others involve substituting components. There's always need for backups plus the master copies of the software to be kept handy.

Repair isn't a very pleasant subject either. Unless the problem is as cleaning dust out of the machine or resetting cards and cables, there's certainly going to be swapping of components. Most problems in running computers are due to software problems.

With the exception of blowing dust out of Floppy disk drives and tape drives, hardware work involves taking the case of the computer apart.

These involve using some tools and this take us to the next point.

2.2 TYPES OF MAINTENANCE

Basically, there are two types of maintenance viz: Corrective Maintenance (CM), and Preventive Maintenance (PM).

Corrective Maintenance: This is a type of maintenance carried out to solve a particular problem once and for all. For example, a hard disk drive that has crashed and has been replaced with a new one is an example of corrective maintenance.

Preventive Maintenance: This is a type of maintenance that is carried out so as to avoid a known fault from occurring. For example, re-formatting the hard disk drive twice a year is a preventive maintenance.

Preventive Maintenance: This is a type of maintenance carried out so as to avoid a known fault from occurring.

For example, re-formatting the hard –disk drive twice a year is a form of preventive. In this work, both type of maintenance are being considered.

2.3 TOOLS AND TEST EQUIPMENT

A selection of basic tools and test equipment is needed before trying to tackle system repair or maintenance. Test equipment allows measurement of important circuit. Parameters such as voltage, current, resistance, capacitance and semiconductors junction conditions.

Certain tools and test equipment are necessary during maintenance, troubleshooting and repair of personal computer. Some of them include:

(a) Electronic Digital Meter: This is handy test equipment that is required during maintenance, troubleshooting and repairing of PCs. For example, voltage level can be ascertained in PC power supply unit using the metre.

(b) Flat screwdriver: It is a tool used in loosening flat-headed screws in PCs.

(c) Star screwdriver: This a tool used in loosening/tightening star-headed screws in a PC.

(d) Allen keys: These are hexagonal-shaped keys that are used in tightening or loosening screws of the same type. Some of these Allen screw are found in some monitors and printers.

(e) Needle nose pliers: They are used in tightening or loosening screws in deep holes or hiding corners in a PC.

(f) Cathode Ray Oscilloscope: It is a test equipment that is used in the following ways:

(i) It shows Voltage versus time on a graphical display.

(ii) One can observe AC and DC signals watch any unusual signals occur in real time (raster and analog video signals).

(iii) When used correctly an Oscilloscope allows one to witness signals and event, which occur in microsecond. The scope as it is sometimes called is used in checking ac, dc, and presence/absence of voltages in both low and high voltage circuits in a monitor and other electronic equipment.

(g) Hand blower: This is used in blowing out dust from the PC. It is a known fact that accumulation of dust in a PC hinders its performance.

(h) Hand lens: This is used to view damages done to components like cracks, burns, or loose contacts.

Fortunately, most maintenance-related issues are easy to spot and resolve. Below are the basic steps to take while performing preventive maintenance.

STEP 1: FILE BACKUPS

By creating a "copy" of the system files (or even just a part of them) the copy can be restored and work continues in the event of a disaster. Some backup software's are needed to format the media and handle the backup and restore operation with windows 95/98, the native backup pallet is usually used (click on start, program, accessories, system tools and backup) the backup software should be compatible with one another

STEP 2: CMOS BACKUPS

All PCs use a sophisticated set of configuration settings (everything from "Date" and "Time" to "memory Hole") which define how the system should be operated. These settings are stored in a small amount of very low-power memory, called CMOS RAM. Each time the PC starts, motherboard BIOS reads the CMOS RAM and copies the contents into low system memory (the BIOS Data Area, BDA), while the system power is off, CMOS contents can be lost. In most cases, this will prevent the system from even starting until you reconfigure the CMOS set up from scratch. By making a backup of the CMOS set up, the cost setting can be restored in a matter of minutes. CMOS backup are simply screens of the CMOS set up page.

What is needed to perform a CMOS back up is a printer. The printer should be attached to the PCs parallel port.

After starting the CMOS setup routine, visit each page of the set up and use the <print screen> key to "capture" each page to the printer.

Because every BIOS is written differently, be sure to check for sub-menus that might be buried under each menu option.

TIPS:

CMOS backup are quick and simple but by following the point below, the most benefits can be obtained:

- Print out every CMOS set up page
- Keep the printer pages taped to the PC 's housing or with the system's original documentation.
- The CMOS set up should be backup wherever a change is made to the system configuration.

CLEANING

The first set of procedures involves exterior cleaning. Items needed for cleaning include:

- Mild Ammonia -based cleaner
- A supply of paper towels or clean lint-free cloths
- A canister of electronics - grade compressed air
- A small static - safe vacuum cleaner

STEP 3: CLEAN THE CASE

The case should be cleaned with cloth lightly dampened with ammonia cleaner to remove dust, dirt or stains from the exterior of the PC.

Care should be taken not to accidentally alter the CD-ROM volume or sound master controls.

STEP 4: CLEAN THE AIR INTAKE

The air intakes of the casing should be carefully attended to, accumulation of dust or debris around the intake should be cleaned off. If the filter is washable, it might be rinsed in simple soap solution and water for best cleaning. If there's no intake filter, simply clean round the intake area.

STEP 5: CLEAN THE SPEAKERS

Multi Media speaker offer a countless number of ridges and openings there are just perfect for accumulating dust and debris.

Compressed air can be used to gently dust out speaker's opening. The long-thin air nozzle should not be inserted into speaker because it can easily puncture the speaker core and ruin it but the nozzle should be removed and air sprayed directly from the can. A clean cloth lightly dampened with ammonia solution should be used to removed dirt and stains from the speaker housing.

STEP 6: CLEAN THE KEYBOARD

Accumulation can jam keys or cause repeated keystrokes. Attach long thin nozzle to the can of compressed air and use the air to blow through the horizontal gap between key rows. The keyboard should be kept away from the face as this will kick up a lot of dust. Ammonia dampened cloth should be used to remove dirt and stain from the key and keyboard housing.

If any key seems unresponsive or "sticky" the corresponding key can be removed and a bit of good quality electronic contact cleaner sprayed into the key cap.

The <ENTER> key or > should not be removed as they are held in place by metal brackets that re extremely difficult to reattach once the key is removed.

STEP 7: CLEAN THE MONITOR

Several issues are important when cleaning a monitor, ventilation, case and CRT. Monitor rely no vent openings for proper cooling. The vacuum cleaner should be used to carefully remove any accumulations of dust and debris from the vent underneath the case.

The vent openings should not block by any of this.

NEXT is to use clean cloth, lightly the monitor's plastic case. Under no circumstance should cleaner be sprayed directly, on the monitor housings as active circuitry is directly under the top vent.

Ammonia should not be used to clean the CRT face. The CRT is often tinted with antiglare and other coating and even mild chemicals can react with some coatings. Clean tap water only should be used to clean CRT face and should be dried completely.

STEP 8: CLEAN THE MOUSE

Mouse is partially susceptible to dust and debris which are carried from the mouse pad up into the mouse ball and rollers. When enough foreign matter has accumulated the mouse cursor hesitates or refuses to move completely.

Loosen the retaining ring and remove the mouse ball. Using a clean cloth and an ammonia solution. The mouse ball should be dried thoroughly and set aside with the retaining ring. Clean cloth should be used to clean the rollers completely compressed air can should be used to blow out any remaining dust or debris that might still be inside the mouse. Finally, the mouse ball should be replaced and secured into place with its retaining ring.

EXTERNAL CHECK

The PC needs to be power up for this check. Three checks should be performed every four (4) months (three times per year).

STEP 9: CHECK THE ETERNAL CABLES

Myriad of external cables interconnect the computers to its peripheral device each cable should be examined and verify that it is securely connected. The following cable should be checked:

- AC power cable for the PC
- AC power cable for the monitor
- AC power cable for the printer
- AC/DC power pack for the an external modem (if used)
- Keyboard cable
- Mouse cable
- Joystick cable (if used)
- Video cable form the sound board
- Serial port cable to external modem
- Parallel port cable to printer

STEP 10: CLEAN THE FLOPPY DISK DRIVE

Floppy disk are a “contact” media - the read/write head of the floppy drive actually contact the floppy disk. This contact transfers some of the magnetic oxide from the floppy disk to the drive's read/write heads.

Eventually, enough oxides can accumulate on the read/write head to cause reading or writing problems with the floppy drive. The floppy drive should be periodically cleaned to remove excess oxides.

2.4 : COMPONENTS OF A PERSONAL COMPUTER

A modern PC is both simple and complicated. It is simple in the sense that over the years many of the components used to construct a system have become integrated with other components in less actual part. It is complicated in the sense that each part in a modern system performs many more functions than did the same types in folder systems: the PC is a modular device and this modularity makes problem determination and repairs possible.

Many repairs just require finding and replacing the faulty component so the first step is to identify what's in the "box". The components needed to assemble a basic modern PC system are listed below:

- Motherboard
- Processor
- Memory (RAM)
- Case (Chassis)
- Power supply
- Hard disk
- CD-ROM, CD-R OR DVD-ROM drive
- Keyboard
- Mouse
- Video Card
- Monitor (display)

* Sound card

MOTHERBOARD

The motherboard is the core of the system it is really the PC - everything else is connected to it and it controls in the system. Motherboards are available in several different shapes or form factors. It usually contains the following individual components:-

* Processor socket (or slot)

* Processor voltage regulators

* Chipset

* Level 2 cache (normally found in the CPU today)

* Memory (SIMM or DIMM) sockets

* Bus slots

* ROM BIOS

* Clock (CMOS battery)

* Super I/O chip

The chipset contains all the primary circuitry that makes up the motherboard; in essence, the chipset is the motherboard. The chipset controls the CPU or processor bus the L2 cache and main memory the PC I (Peripheral component interconnect) bus, the ISA (Industry Standard Architecture) bus, system resources and more.

If the processor represents the engine of the system the, chipset represents the chassis in which the engine is installed.

The chipset plays a big role in determining what sort of features a system can support. e.g. which processor to be used, which type and how much memory can be installed, what speed the machine can be run, and what types of system buses the system can support are all tied in to the motherboard chipset.

The ROM BIOS contains the initial POST (Power-on self Test) Program, bootstrap loader (which loads the operating system) drives for items that are built into the board (the actual BIOS code), and usually system set up program (often called CMOS set up) for configuring the system.

PROCESSOR

The processor is often thought of as the engine of the computer. It is also called the CPU (central Processing Unit), it is the single most important chip in the system because it is the primary circuit that carries out the program instructions of whatever software is being run.

Modern processors contain literally millions of transistors, etched onto a tiny square of silicon called a die, which is about the size of a thumbnail. The processor has the distinction of being one of the most expensive parts of most computers, even though it is also one of the smallest parts.

Intel is generally credited with creating the first microprocessor in 1971 with the introduction of a chip called the 4004. Today, Intel have almost total control over the processor market, at least for PC systems, this means tat all PC-Compatible systems use either Intel Processor or intel compatible processors from a handful of competitors (such as AMD or Cyrix).

In some ways the success of the PC, and the Intel architecture it contains has limited the growth of the PC. In other ways, however, its success has caused a huge number of programs, peripherals, and accessories to be developed, and the PC to become a de facto standard in the industry. The original 8088 processor used in the first PC contained closed to 30,000 transistors and at less than 5mhz.

The most recent processors from Intel contain close to 30 million transistors and run at over 500mhz. Intel has already demonstrated processors running at 1Ghz.

Processors can be identified by two main parameters: how wide they are and how fast they are. The speed of a processor is a fairly simple concept. Speed is counted in megahertz (MHz), which means millions of cycles per second and faster is better. The width of a processor is a little more complicated to discuss because there are 3 main specifications in a processor that are expressed in width. They are:

* Data input and output bus

* Internal registers

* Memory address bus

MEMORY (RAM)

The system memory is often called RAM (Random Access Memory). This is the primary memory, which holds all the programs and data the processor is using at a given time. RAM requires power to maintain storage, so when the computer turn off everything in the RAM is clear; when it is turn back on, the memory must be reloaded with programs for the processor to run. The initial program for the processor come from a special type of memory called ROM (Read Only Memory), which is not erased when the power to the system is turned off. The ROM contains instructions to get the system to load or boot an operating system and other programs one of the disk drives into the main RAM memory so that the systems can run normally and perform useful work.

Newer operating systems allow several programs to run at one time, with each program or data file that is loaded using some of the main memory.

Generally, the more the system has, the more programs you can run simultaneously. Memory is the workspaces for the computer processor. It is a temporary storage area where the programs and data being operated on by the processor must reside. Memory storage is considered temporary because the data and programs remain there only as long as the computer has electrical power or is not reset. Before being shut down or reset, any data that has been changed should be saved to a more permanent storage devices of some type (usually a hard disk) so it can be reloaded into memory again in the future.

Main memory is called RAM because it can randomly (and quickly) access any location in memory. ROM is also randomly accessible, yet is normally differentiated from the system RAM because it cannot normally be written to. RAM can refer to both the physical chips that make up the memory in the system and the logical napping and layout of that memory. Logical napping and layout refer to how the memory addresses are mapped to actual chips and what address locations contain which types of system information. Memory temporarily stores programs when they are running, along with the data being used by those programs. RAM chips are sometimes termed volatile storage because when the computer is turn off or an electrical outage occurs, whatever is stored in RAM is lost unless it is saved to the hard drive. Physically, the main memory is a collection of chips or modules containing chips that are normally plugged into the motherboard. These chips or modules vary in their electrical and physical design and must be compatible with the system into which they are being installed to function properly.

CASE (CHASSIS)

The case is the frame or chassis that houses the motherboard, power supply, disk drives, adapter cards, and any other physical components in the system. There are several different styles of cases available from small or slim version that sits horizontally on a desktop to huge tower types that stand vertically on the floor. In addition to the physical styles, different cases are designed to accept different form factor motherboards and power supplies.

Some cases have factor make installing or removing components easy, such as a screw less design that require no tools to disassemble, side open panels or trays that allow easy motherboard access, removable cages or brackets that give access to disk drive, etc some cases include additional cooling fans for heavy duty systems, and some are even available with air filters that ensure that the interior will remain clean and dust free. Most cases include power supply but power supplies and cases can also purchase separately.

FLOPPY DISK DRIVE

The floppy drive is a simple, inexpensive, low capacity removable media magnetic storage device. FLOPPY disks were the primary medium software distribution and system backup for years with the advent of CD, ROM and DVD-ROM discs as the primary method of installing or loading new software in a system, and with inexpensive high capacity tape drives for back up the floppy drive is not used often in most modern systems. Floppy drives is the first device from which a PC attempts to boot, and it is still the primary method that is used for loading initial operating system start up software and core hardware diagnostics.

Recent advancement in technology have created new types of floppy drives with more than 120MB storage, making the drive much more usable for temporary backups or for moving files from system to system.

HARD DISK DRIVE

A hard disk drive is a sealed unit that a PC uses for nonvolatile data storage. Nonvolatile or permanent storage device retains the data even when there is no power supplied to the computer.

Because the hard drive is expected to retain its data until a user deliberately erase it, the PC uses it to store its most crucial programming and data. As a result, when the hard disk fails, the consequences are usually very serious.

An understanding of the hard disk function is needed to maintain service and expand a PC system properly.

A hard disk drive contains rigid, disk-shaped platters usually constructed of aluminium or glass. Unlike floppy disks, the platters cannot be removed; the reason why they are sometimes called fixed disk drives. Removable hard disk drives are also available. The basic physical construction of a hard disk drive consist of spinning disks with heads that move over the disks and store data in tracks and sectors. The head read and write data in concentric rings called tracks, which are divided up into segments called sectors, which normally store 512 bytes each. Hard disks drives usually have multiple disks called platters, that are stacked on top of each other and spin in union, each with two sides on which the drive stores data.

CD-ROM DRIVE

The CD-ROM is a read-only optical storage medium capable of holding up to 682MB of data (approximately 333,000 pages of text), 74 minutes of high-fidelity audio or some combination of the two on a single side of a file-inch disk. The CD-ROM is similar to the familiar audio compact disc and can, in fact. Play in a normal audio player. Accessing data from a CD-ROM is quite a bit faster than from a floppy disk but considerably slower than a modern hard drive.

The term CD-ROM refers to both the discs themselves and the drive that reads them. Although identical in appearance to audio CDs, CD-ROMs store data instead of (or in addition to) audio. The CD-ROM drives in PC s that read the data audio disc bear a strong resemblance to an audio CD-player. This water base is coated with a metallic film, usually an aluminium alloy. The aluminium film is the portion of the disk that the CD-ROM drive reads for information. The aluminum film or strait is then covered by a plastic polycarbonate coating that protects the underlying data. A label is usually placed on the top of the disc, and all reading occurs from the bottom. CD-ROMs are single-sided.

KEYBOARD

The keyboard is one of the basic components of the computer system. It is the primary input device and is used for entering data and commands into the system, in the years, since the introduction of the original IBM PC. IBM has created three different keyboard designs for PC systems; Microsoft has augmented.

These designs have become de facto standards in the industry and are shared by virtually all PC manufacturers. With the introduction of Window 95, a modified version of the standard 101 key design (created by Microsoft) appeared, called the 104 Key Windows Keyboard.

The primary keyboard types are as follows:

- 101 key enhanced keyboard
- 104 key window keyboard
- 83 key PC windows keyboard (obsolete)
- 84 key AT keyboard (obsolete)

The enhanced keyboard is available in several different variations, but all basically the same electrically and all can be interchanged.

The 101-keyboard layout can be divided into the following sections:

- Typing Area
- Numeric Keypad
- Cursor and Screen Control
- Function Keys

MOUSE

With the advent of computer operating system that used a graphical user interface (GUI), it became necessary to have a device that enabled a user to point at or select items that were shown on the screen. Although there are many different types of pointing devices in the market today, the first and most popular device for this purpose is the mouse.

By moving the mouse across a desk or table top, a corresponding pointer can be moved across the computer screen, allowing items to be more easily selected or manipulated than they can with a keyboard alone. Stand mice, as used on PCs have two buttons, one for selecting menus, mice are also available with a third button, a wheel, or a stick, which can be used to scroll the display or for other special functions.

VIDEO CARD

The video card controls the information that is seen on the monitor. All video cards have four basic parts: a video chip or chip set RAM, a DAC (Digital to Analog Converter) and a BIOS. The video chip actually controls the information on the screen by writing data to the video RAM. The DAC reads the video RAM and converts the digital data there into analog signals to drive the monitor. The BIOS holds the primary video driver that allows the display to function during boot time and at a DOS prompt in basic text mode.

More enhanced drivers are then usually loaded from disk to enable advanced video modes for windows or applications software.

MONITOR (DISPLAY)

In most systems, the monitor is housed in its own protective case, separate from the system case and chassis. In portable systems and some low-cost PCs, however, the monitor is built into the system case.

Monitors are generally classified by 3 major criteria: diagonal size in inches, resolution in pixels and fresh rate in hertz (Hz). Desktop monitors usually range from 14 to 21 diagonal measure (although the actual viewable are is smaller than the advertised measure).

LCD monitors in portable systems range from 11 to 14. Resolution ranges from 640 x 480 pixels (horizontal measurement first and then vertical) to 1600 x 1200 pixels. Each pixel in the monitor is made of a trio dots, one each for the colours red, blue and green.

An average monitor is capable of refreshing 60 times per second (60Hz), whereas higher quality monitors might refresh at 100Hz. the refresh rate measures how often the display of the screen is redrawn from the contents of the video adapter memory. Both the resolution and refresh rate of the monitor are tied into the capability of the system video adapter. Most monitors are capable of supporting several different resolutions and refresh rates (with the common exception of LCD screens in portables).

CHAPTER THREE

3.0 TROUBLESHOOTING ANALYSIS

3.1 BASIC TROUBLESHOOTING GUIDELINES

Before starting any system troubleshooting, there are a few basic steps that should be performed to ensure consistent starting point and to allow isolating the failed component.

1. Turn off the system and any peripherals from devices. Disconnect all external peripherals from the system except for the keyboard and the video display.

2. Make sure the system is plugged into a properly grounded power outlet.

3. Make sure the keyboard and the video displays are connected to the system. Turn on the video display, and turn up the brightness and contrast controls to at least two thirds of the maximum. Some displays have on screen controls that may not be intuitive. Consult the display documentation for more information on how to adjust these settings.

4. To allow the system to boot from a hard disk, make sure there is no floppy disk in the floppy with DOS or diagnostics on it in the floppy drive for testing.

5. Turn on the system. Observe the power supply and chassis fan (if any), and the lights on either the system front panel or power supply. If the fans don't spin and the lights on either the system front panel or power supply. Or the motherboard might be defective.

6. Observe the power-on-self test (POST). If no errors are detected, the system beeps once and boot up. Errors that display on the screen (non-fatal

errors) and which do not lock up the system display a text message that varies according to BIOS type and version. Record any error that occur and refer to the codes for information. Errors that lock up the system (fatal error) are indicated by a series of audible beeps.

7. Confirm that the operating system loads successfully.

PROBLEMS DURING THE POST

Problems that occur during the POST are usually caused by incorrect hardware configuration or installation. Actual hardware failure is a far less frequent cause.

If you have a POST error, check the following:

1. Are all cables correctly connected and secured?
2. Are the configuration setting correct in set up?
3. Are all drivers properly installed?
4. Are switches and jumpers on the baseboard correct, if changed from the default setting?

3.2 THE TROUBLESHOOTING PROCESS

Regardless of how complex a particular computer or peripheral device may be, a dependable troubleshooting procedure can be broken down into 4 basic steps:

(1) Define the symptoms

(2) Identify and isolate the potential source (or location) of the problem

(3) Repair or replace the suspected component or assembly

(4) Retest the unit thoroughly to be sure that the problem has been solved. If the problem is not solved start again from the computer troubleshooting, not just for PC memory systems.

DEFINE THE SYMPTOM:

A firm understanding of all symptoms is needed before opening up the system. The following should be considered:

Is the disk or tape inserted properly? Is the power or access LED fit? Does the problem occur only when the computer is tapped or moved? By recognizing and understanding the symptoms, it can be much easier to trace a problem to the appropriate assembly or component.

The symptom should be written down. The note taking may seem tedious, but written record of symptom and circumstances will help keep fit on the task hand.

IDENTIFY AND ISOLATE:.

Before trying to isolate a problem within a piece of computer hardware, it's good to be first sure that it is the test equipment itself that is causing the problem. It should be remembered that storage device work because of an intimate mingling of hardware and software. A faulty or improperly configured piece of software can cause configuring system errors.

After configuring that the failure lies in the system's hardware, the possible problem can be identified. The actual repair process can only begin after the potential problem area is identified and it possibly tracking of the fault to components level.

REPAIR OR REPLACE

After understanding what is wrong and where to look, the actual repair process can now begin which will correct the symptoms. Most storage devices are a mix of both electronics circuit and electromechanical devices, so most procedures will require the exchange of electronic or electromechanical parts. As a general rule, all procedures should be considered important and should be careful.

Parts are usually classified as components subassemblies. Components can serve many different purposes in a computer Resistors, capacitors, transformers, monitor, and ICs are just a few type of component part within themselves a detective component must be replaced. A subassembly can be composed of many individual components. Unlike components sub assembling serve a single specific purpose in a storage device (i.e in read/write head or amplifier PC board), but it can usually be repaired by locating and replacing a faulty keyboard then repair it, also monitor are always being advised to replace than repair.

RETEST

When a repair is finally complete, the system must be reassembled carefully before testing it. Guards, housings, cables and shields must be restored before final testing. If the symptoms persist, the symptoms need to be re-evaluated and narrow the problem to another part of the equipment. If normal operation is restored (or greatly improved the computer's items functions should be tested.

Storage devices are just a collection of assemblies and each assembly is a collection of part. Normally, everything works together, but when one part fails it may cause one or more interconnected part to fail as well.

Several repair attempts should be made while repairing a computer, be prepared to make several attempts before the computer is repaired completely.

POST REPAIR TESTING

One should be familiar with the idea of "burn-in" testing once the repair is complete. In modern electronics, the general rule of reliability suggests that a device will fail very quickly or will last a long time. With this rule in mind, it is usually desirable to run the PC for at least 24 hours after the repair is done. This will "stress" the new part or drive, and (hopefully) any marginal part will fail right there while the system is still on the workbench. This kind of testing improves the reliability of the repair.

3.3 DIAGNOSTIC SOFTWARE

Typical software tool include the commercial standard like DEFRAG, SCANDISK and CHKDSK.

- DATA REG. ZIP

A hard disk crash is frustrating and frightening. Even with the fully backed up the time and expense of troubleshooting and disk replacement can be an expense proposition. Recovering data from a crashed hard disk is more of an art than a science, and there are few tools that are actually up to the task, but Tiramisu is one of the only hardware tools available that can reconstruct data from failed drives. Tiramisu is designed to help devices that have been hit by from failed drives. Tiramisu is designed to help devices that have been hit by a virus, scratched by a head crash, accidentally formatted or partitioned, corrupted by a power failure, or damaged by buggy applications. Tiramisu scans the disk even when there are physical damages - the found data are analyzed and reconstructed. The program works on drives without a readable boot sector, readable FAT or readable directory entries.

- **DUGIDE ZIP**

Every hard drive ever manufactured has its own "geometry" - the number of cylinders, heads, sectors and so on. Which defines the way a drive retain data. When a new drive is installed, or old drive is replaced, or the contents of CMOS are lost, and one is faced with the prospect of tracking down and restoring the geometry figures for each hard drive. Programs like DUG_IDE.EXE integrated a hard drive and display the resulting details for quick reference.

- **AUTO TEST. ZIP**

Part of device testing often involves measuring performance, while performance is not a critical factor with floppy drives, poor performance (usually in conjunction with other DOS errors) may suggest a failing device or the presence of a hardware conflict. The AUTOTEST.EXE diagnostic is designed to measure floppy drive performance by putting the drive through a series of random and sequential reads. Unusually long read time may suggest problems with the hardware heads, track stepping motor, or spindle motor.

- **DFR.ZIP**

There are some occasions where the data on a floppy disk is just as valuable as the data on a hard drive. When the floppy diskette fails the loss of program on it may be catastrophic. The DFR package mark Vitt is a highly automated DFR can recover files to hardware or another floppy disk.

- **CD SPEED.ZIP**

The performance of a CD-ROM drive may eventually need to be measured and evaluated during the course of troubleshooting or upgrade procedure. The CD SPEED. ZIP package provides a tool that sustained data transfer rate of a CD ROM drive. The efficiency of the CD ROM device in services, and the overall utilization of CPU processing power (which helps determines just how much of a load the CD-ROM device actually is on the system.

- **CDTA.ZIP**

Timing and performance are vital aspects of all drives. Thus there is need to measure the performance and data throughout of new or replacement CD-ROM device. The CDTA.ZIP package is designed to test the data throughputs and access times for most CD ROM drives using conventional low-level devices and MSCDEX.

- **CHECK 136.ZIP**

CHECKSYS is a basic system diagnostic that abandons the use of menu screens and windows in favour of a command-line system. When CHECKSYS is executed, it will return a description of the whole PC system (or an error level if necessary). CHECKSYS can also be used to inspect specific systems and return one line description of a system status. The command-line diagnostic functions in the system start up files (i.e. AUTO EXEC. BAT), or other batch files.

CONF810E.ZIP

Often lies in knowing “what’s in the box” Rather than the time consuming process of disassembling the computer and examining each item by eye, a well-written up to the minute system information utility can tell everything one needs to know before even picking up a screwdriver. The long-running PC - config series of utilities by Michael Holin is one of the most recognized and respected shareware system information utilities bench marking programs available.

3.4 APPLYING ABC OF REPAIRS

The ABC of repairs simply means a three-fold rule that should be used in carrying out repairs on any PC irrespective of the manufacturer, size, speed, and standalone or in a network.

Rule I: Know the last thing that happened before the fault occurred. A computer user, system engineer or other professional should know exactly the last thing or action that preceded a break down. This will help in repairing the PC with minimum delay.

Rule II: Availability of test equipment/tools. Before repair can be carried out, test equipment/tools like the electronic digital metre ought to be available, this will assist in repair.

Rule III: Availability of replacement parts, spare parts like RAM, Motherboard, Processor, FDD, HDD etc, ought to be available at the time of any repair job.

This will reduce the number of man-hour lost and make replacement favorites past time.

3.5 GENERAL REQUIREMENT FOR SYSTEM SOFTWARE

As end-user requirements becomes more exacting, collection of programs grow in complexity and size. The following requirements were taken into consideration in the developing of the program.

1. **Correctness and Reliability:** These outweigh all the other requirements. Fault in the program were taken into consideration and corrected so that logic errors can be avoided, as the computer cannot detect it.

2. **Flexibility and Irreversibility:** The program was design in such a way that it can easily be modified.

3. **Efficiency:** As a software system grows in size and complexity, it also tends to run slower. Slow response time may be quite time consuming and strenuous but efficiency was not pursued at the expense of correctness and only in extreme cases at the expense of flexibility. The program is quite efficient as it minimize time and maximizes efficiency.

3.6 SYSTEM ANALYSIS

The object of the analysis phase is to find out exactly what the end-user wants the final program to do (of course the programmer himself may be the end-user). The problem is often that users do not know what to expect from the program. The program has been written in such a way that the end-user knows what the program is expected to perform.

3.7 DESIGN OBJECTIVES

The users were taken into consideration by making user-friendly, through the use of interactive mode. The use of custom screens for economic reason was also adopted.

Objectives of the designing of the system are:

(a) To list all the necessary files that would be used

(b) To keep logbook of all successful operation on computers

(c) To save status of user and nature of work done on system in the program using the consultation form.

(d) To develop procedures and programs

(e) To identify how all programs will work together as a system.

3.8. PROGRAM DESIGN:

The general system design includes the flowchart (procedure), a written explanation (Pseudocode). For this study, coding (programming) is used to depict the representation of the project. The output forms are in Appendix II.

3.8.1 INPUT DESIGN:

Input design, which is also known as file design, is the denominator of any system. It contains the raw material (Data) necessary to produce output.

In manufacturing, for example, it is only when the product to be made is decided upon that the raw material could be specified and ordered. In the process of developing an Information system, output requirement are first decided upon before data are specified. In a sense, output requirement can be thought of as an input database design.

Therefore, the data file design includes the entire database files used throughout the system for the proper storage of the course processing data.

Basic inputs to be used in the system are as follows:

FIELD	FIELD NAME	FIELD TYPE	WIDTH	DECIMAL PLACES
1.	TOPIC	Character	50	-
2.	Prob_Desc	Memo	4	-
3.	Who	Character	25	-
4.	When	Date	8	-
5.	How	Character	10	-
6.	Related	Logical	1	-

7.	Complaint	Memo	4	-
8.	Where	Character	20	-
9.	Periodic	Logical	1	-
10.	Components	Memo	4	-
11.	Observation	Memo	4	-
12.	Site action	Memo	4	-
13.	Date_in	Date	8	-
14.	Result	Memo	4	-
15.	Date_out	Date	8	-
16.	Prob_type	Character	10	-
17.	Code	Numeric	4	-
18.	Shop_action	Memo	4	-

3.8.2 OUTPUT DESIGN:

The output/report generated by the system includes:

(a) List of all successful operations on the computer

(b) Suggested reasons for the problem of the machine together with the observation

(c) The different components of a PC system

(d) The description of problem as reported by the client

(e) The fault as diagnosed by the engineer

(f) Report on general troubleshooting guidelines.

CHAPTER FOUR

4.0 PROGRAM IMPLEMENTATION

4.1 PROGRAM DEVELOPMENT/ IMPLEMENTATION

The process of software development is a co-operative effort of the users of the software and system analyst. While the analysts are the professionals that deals with the technology and its application to project information processing needs, the users of the software on the other side have an in depth familiarity with the respective manual functional areas.

The skill and knowledge of these two groups complement each other and can be combined to create any type of information systems during the course of the software development process.

However, because software development is a team effort, most programmers have adapted a standardized “System/Software Development Methodology”, that provides a framework for cooperation and successful development of a new system.

This step-by-step system development procedures is as illustrated below in a tabular form.

STEPS	ACTIVITIES	MEASURABLE OUTPUT
Analysis and specification	Appraisal of existing situation identification of users requirement	
Design	Design the overall program structure Design the detailed processor Processing logic	Process logic specifications, using Pseudocode or flow chart techniques.
Programming	Writing of code in the appropriate	Hand written program code

	Entry of written code into the computer	Computer printed list out of code.
Testing and debugging	Removal of syntax and logic errors. Final testing of program	Error free Program execution
Installation and maintenance	Error free program installation User training	Program documentation and up grade procedures.

However, software development process is essentially the same, be it for Inventory Management System, Currency Processing, airline reservation system etc. As members of a "Project Team" progress through the procedures outlined in a software development methodology, the result of one step provided the input for the next step and/or subsequent steps. The project team typically is made up of both users and Computer Engineers.

The methodological approach to software development is a tool information services and users employ to coordinate the effort of a variety of people engaged in a complex process. Hence for a successful program development of PGD software, the writer of this project work is part of the object team as Programmer. The above five (5) steps in the table were followed one after the other in order to design and develop a customized software package called the "TR-Guide Software".

4.2 CHOICE OF PROGRAMMING LANGUAGE

From the previous analysis, it is pertinent to say that the proposed system is going to be used to store large number of data/information, and time-to-time retrieval of record. Due to this fact, the choice of the programming language chosen for the development of the system is Database Management System (DBMS) package with special preference for Visual Fox Pro.

4.2.1 FEATURES OF THE PROGRAMMING LANGUAGE CHOSEN

The choice of Visual Fox Pro arose because of the following reasons and features posed by the application software.

- It is easy to write an interactive user interface program and also simple to understand.
- It is user's friendly.
- It reduces data redundancy.
- Data integrity can be maintained
- Provide easy and greater access to information.
- Individual database file can be designed to meet specification requirement of particular functional unit of an organisation.
- The Visual Fox Pro allows the source program to be compiled to an executable file thereby allowing the program to be run independently of the application software that was used in coding the program e.g. Visual Fox Pro.
- It has quality graphical users' interface.

4.3 PHYSICAL AND LOGICAL DESIGN OF THE SYSTEM

The physical and logical design of the system deals with the physical construction of the logical design of the proposed system. It has to do with program specification of output, input file and processing into computer software.

The design of the computer software is important to ensure that the actual program produced performs the entire task intended and to allow for future modification to be performed efficiently and with minimum destruction to the design of the system.

4.4 SYSTEM IMPLEMENTATION

System implementation is a broad term that encompasses testing and debugging, hardware and software requirement, system installation and system conversion.

It is also the coordination of the fact, which is necessary in ensuing the operation of the new system.

4.5 SYSTEM TESTING AND DEBUGGING.

The essence of program testing is to make sure that program is error free and that all the logic involved are well defined and straightforward. However, it is often seen as a means of establishing that a program is error free and that it does what is required. It is virtually impossible to test a program so thoroughly that it can be claimed to be free of error. In most cases fixing one error gives rise to host of others, which in turn have to be corrected exhaustively. It is much more realistic to think of testing as a "process of finding error". When a stage is reached, when the program appears to run perfectly, this does not mean that there are no more errors in the program, it simply mean that those errors have not been discovered.

Hence, the TR-Guide software have been developed, tested with real live data, irrespective of people and environment and it was found to be error free.

4.6 HARDWARE AND SOFTWARE REQUIREMENT

To make maximum utilization of the developed system, certain hardware and software needs to be installed.

4.6.1 HARDWARE REQUIREMENT

This comprises of all the physical component of the computer and its accessories. Therefore the choice of the computer requirement is done to suit both the current and the future needs of the organization with respect to the volume and types of data to be processed. In summary, a computer system with the following minimum requirement is required.

4.6.2 SOFTWARE REQUIREMENTS

Software requirement are the basic and other relevant application software that need to be installed on the computer in order to make maximum utilization of the computer system and the developed TR-Guide software. It is because of this fact that the following software must be installed on the system.

However, other application software apart from the ones mentioned above may be installed as may be required by the staff and organization concerned.

4.7 PROGRAM INSTALLATION

This is the process of transferring the developed software from the floppy disk to a permanent storage device called the Hard Disk.

However, due to vast improvement in recent operating systems, the installation of the software is a very simple operation.

The procedure as follows:

STEPS	PROCEDURE	RESULT
1. Go to start	Click	Start popup menu is displayed.
2. Run submenu	Click	Run dialog box displayed
3. Insert TR-Guide Software diskette into A:\	-	-
4. Type the source drive (A:\)	Browse	Content of A:\ displayed
5. Select set-up	Double – click	Installation begins
6. Follow the instruction that Follows	Click	Installation in progress
7. Destination	-	Software installed C:\ successfully

BRINGING THE SOFTWARE FROM C:\ TO WINDOW PROGRAM SUBMENU

STEPS	PROCEDURE	RESULT
1. Go to start	Click	Start popup menu is displayed
2. Setting submenu	Click	-
3. Choose taskbar and start menu	Click	Task dialog box displayed
4. Select start menu program	Click	-
5. Browse to select TR-Guide software	Double – Click	-
6. Select folder (Program)	Click	-
7. Destination	Click	TR-Guide software copied into program submenu successfully.

At the end of the installation, the floppy diskette becomes a backup and should be well safeguarded against any damages for future use.

To start the system after successful installation of the TR-Guide software, all one need to do is to go to start and run the folder name you had created.

4.8 PROGRAM LAYOUT DESIGN

The main menu of the TR-Guide software is in form of rectangular square containing pop up or pulls down menu options. The user of the system is expected to use the mouse or up and down arrow key to highlight the option. This is preceded by either enter key or double clicking to carry out the execution.

CHAPTER FIVE

5.0 SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 SUMMARY

The aims and objective of this project is to outline an effective and easily comprehensible approach to the subject of PC maintenance, troubleshooting and repair.

It was in view of the fact that customized software was suggested and undertaken to enable the users, system engineers, and other professionals carry out their routine maintenance and troubleshooting tasks. This method of computerization will completely reduce if not eliminate the burden of guesswork or suspending the system until an engineer comes when a particular error has occurs.

5.2 CONCLUSION

Without fear of being contradicted, computer technology and applications has become an indispensable factor in the world we live in. Also, advances in this field have been dynamic and not static. This astronomical change in the computer world includes a significant decrease in size, increase in speed of processing, accuracy, reliability and even in the whole technology itself.

One can now access a computer system, take a look at the error, note down the error and correct such error without necessarily looking for a system engineer.

It is in the light of this, that the project work was undertaken to alleviate the burden of maintaining, troubleshooting and repair a PC.

5.3 RECOMMENDATION

It is my humble opinion that if the steps and rules outlined in the preceding chapters were religiously adhered to, then the problem and probable solution in every PC would have been found.

REFERENCES

Barry Press (1998): PC Upgrade and Repair Bible (Professional Edition). Published by IDG books Worldwide, Inc.: England

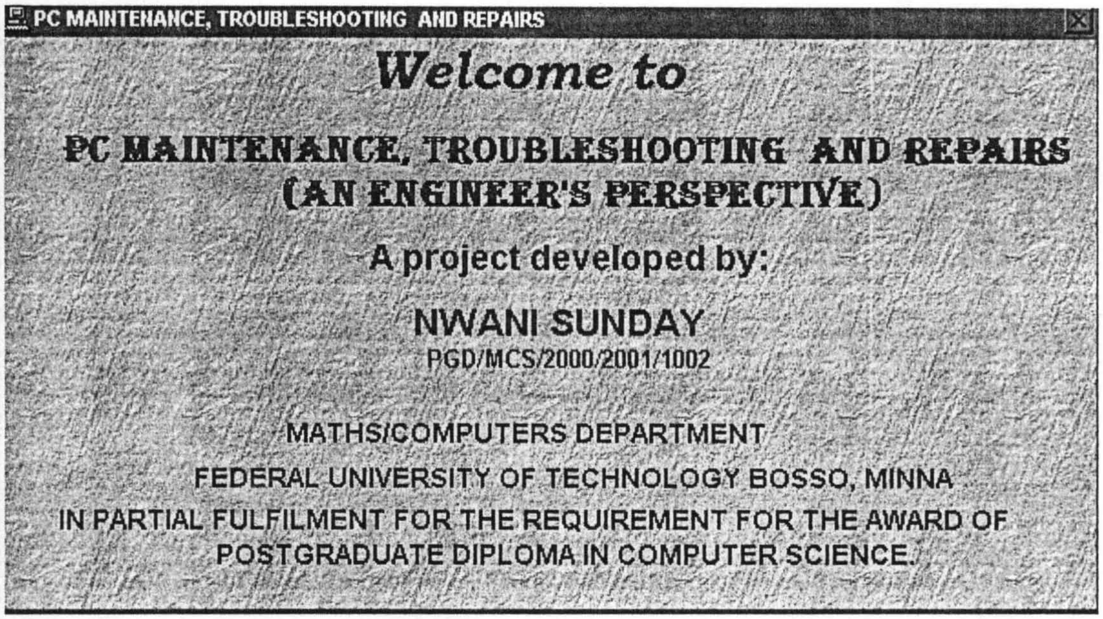
Bigelow S. J. (1998): PC Technician Troubleshooting Pocket Reference (Second Edition). Published by McGraw-Hill: New York.

Bigelow S. J. (1998): Troubleshooting, Maintaining & Repairing PC s (Second Edition) Published by McGraw-Hill: New York

Ismaila O. Nwankwo (1999): Computer Studies & Maintenance. Published and Printed by John Jacob's Classic Publishers Ltd, Okpara Avenue, Enugu.

Mark Minasi (1996): The Complete PC Upgrade and Maintenance Guide (Ninth Edition). Published Gary Masters: Illinois.

Scott Muller (1999): Upgrading and Repairing PC (Eleven Edition). Published by Que (R) Corporation: Indianapolis



PC MAINTENANCE, TROUBLESHOOTING AND REPAIRS

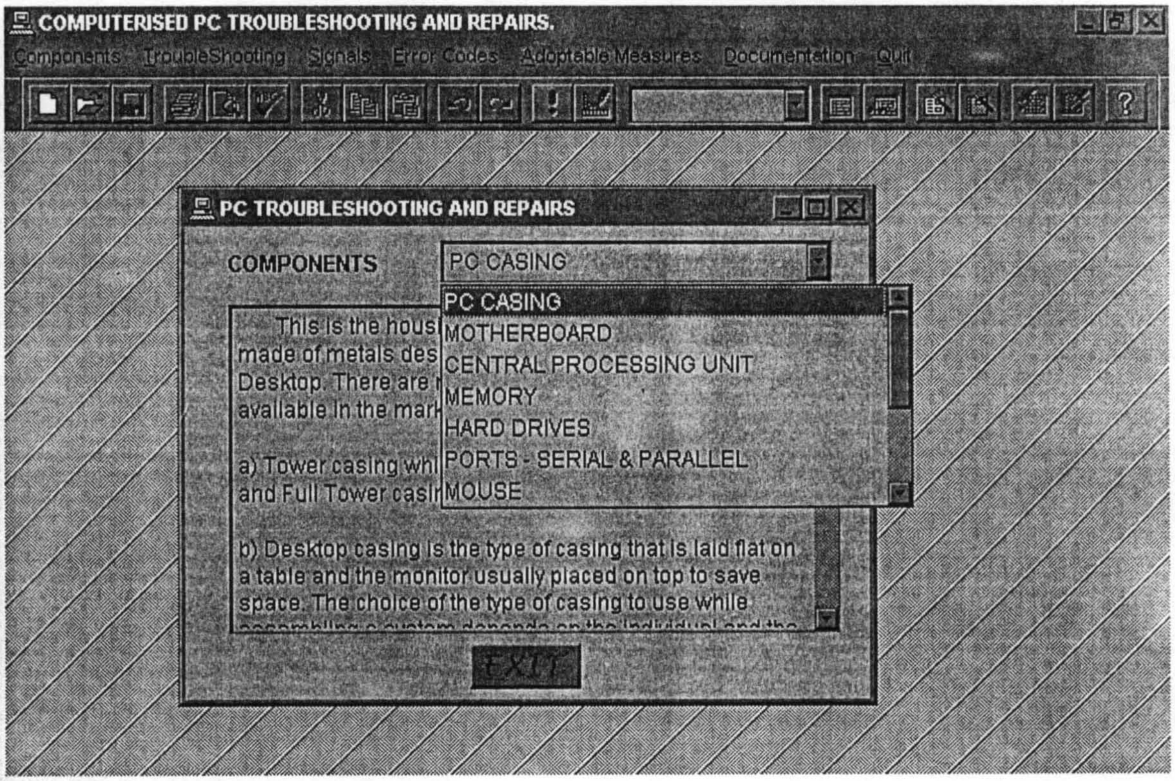
Welcome to

PC MAINTENANCE, TROUBLESHOOTING AND REPAIRS
(AN ENGINEER'S PERSPECTIVE)

A project developed by:

NWANI SUNDAY
PGD/MCS/2000/2001/1002

MATHS/COMPUTERS DEPARTMENT
FEDERAL UNIVERSITY OF TECHNOLOGY BOSSO, MINNA
IN PARTIAL FULFILMENT FOR THE REQUIREMENT FOR THE AWARD OF
POSTGRADUATE DIPLOMA IN COMPUTER SCIENCE.



COMPUTERISED PC TROUBLESHOOTING AND REPAIRS.

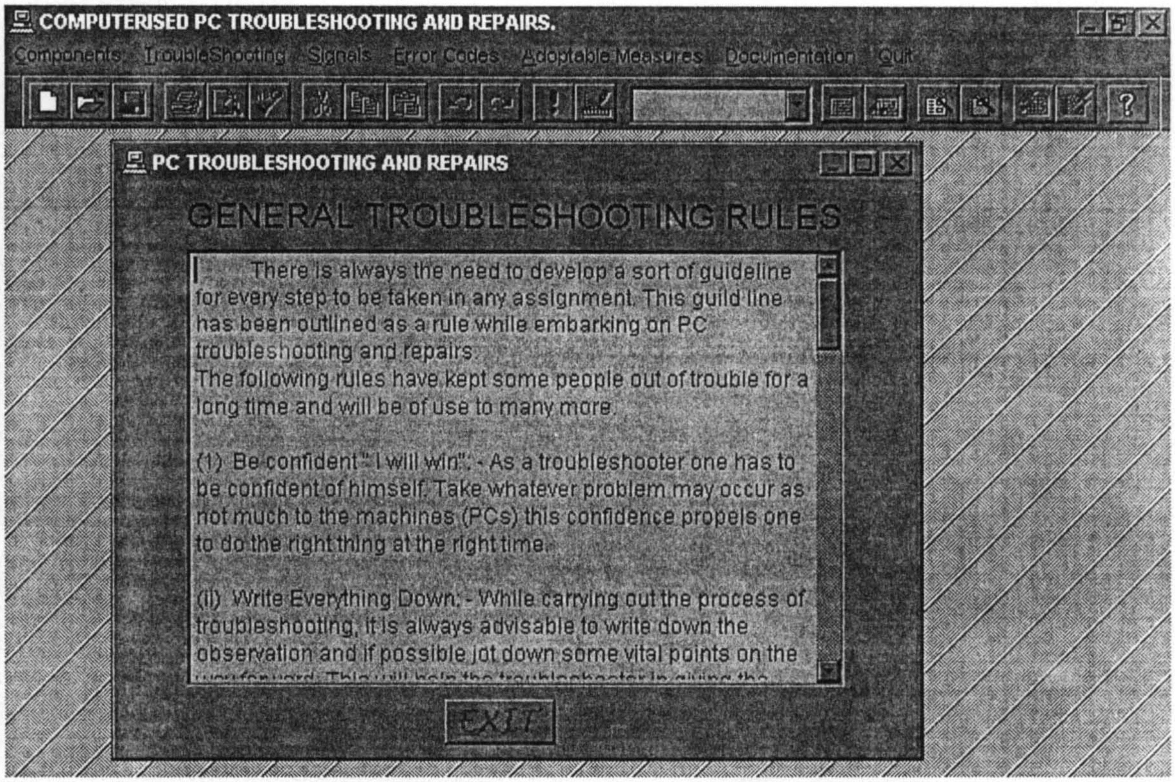
PC TROUBLESHOOTING AND REPAIRS

COMPONENTS: PC CASING

This is the hous
made of metals des
Desktop. There are
available in the mark

a) Tower casing whi
and Full Tower casi

b) Desktop casing is the type of casing that is laid flat on a table and the monitor usually placed on top to save space. The choice of the type of casing to use while assembling a custom depends on the individual and the



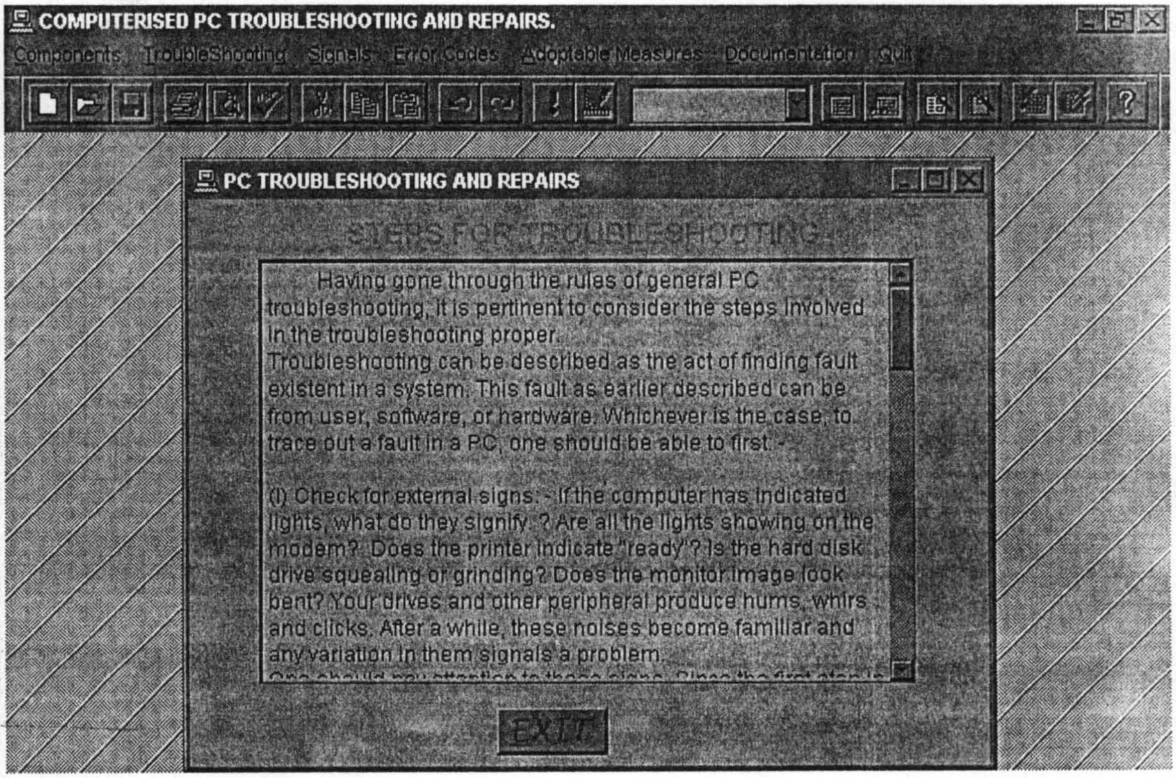
GENERAL TROUBLESHOOTING RULES

There is always the need to develop a sort of guideline for every step to be taken in any assignment. This guild line has been outlined as a rule while embarking on PC troubleshooting and repairs.

The following rules have kept some people out of trouble for a long time and will be of use to many more.

(1) Be confident " I will win". - As a troubleshooter one has to be confident of himself. Take whatever problem may occur as not much to the machines (PCs) this confidence propels one to do the right thing at the right time.

(ii) Write Everything Down: - While carrying out the process of troubleshooting, it is always advisable to write down the observation and if possible jot down some vital points on the way forward. This will help the troubleshooter in giving the



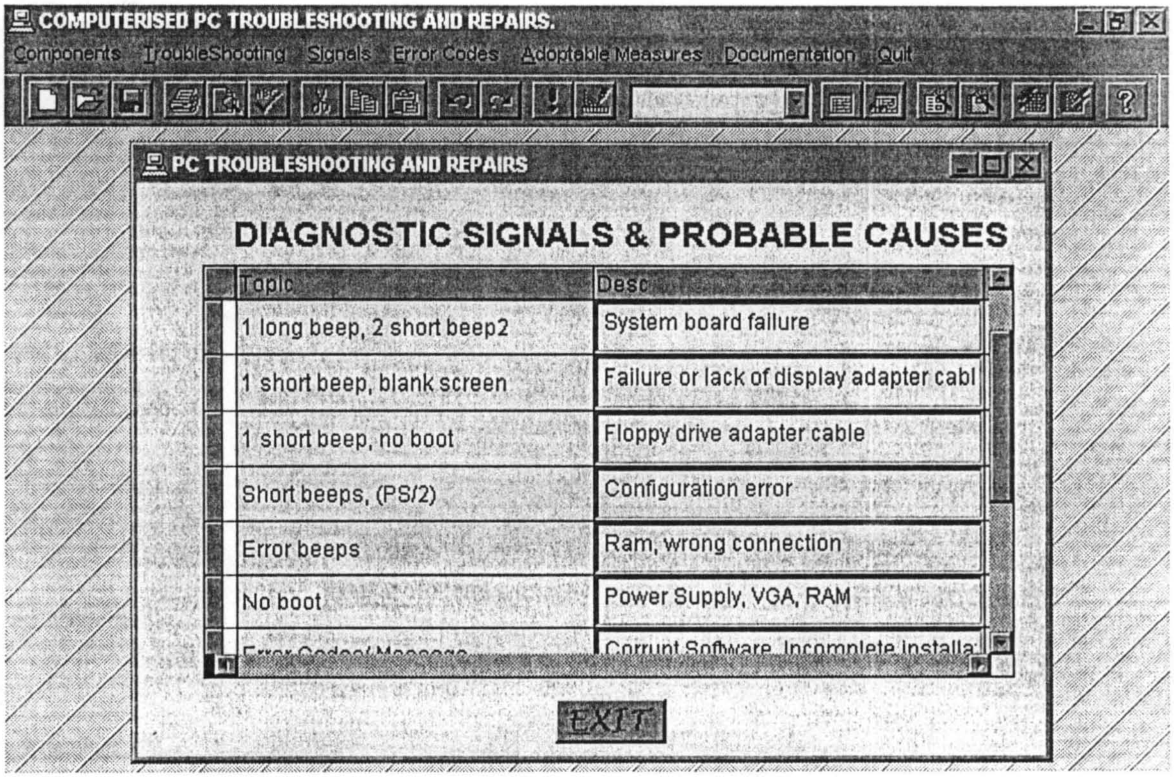
STEPS FOR TROUBLESHOOTING

Having gone through the rules of general PC troubleshooting, it is pertinent to consider the steps involved in the troubleshooting proper.

Troubleshooting can be described as the act of finding fault existent in a system. This fault as earlier described can be from user, software, or hardware. Whichever is the case, to trace out a fault in a PC, one should be able to first. -

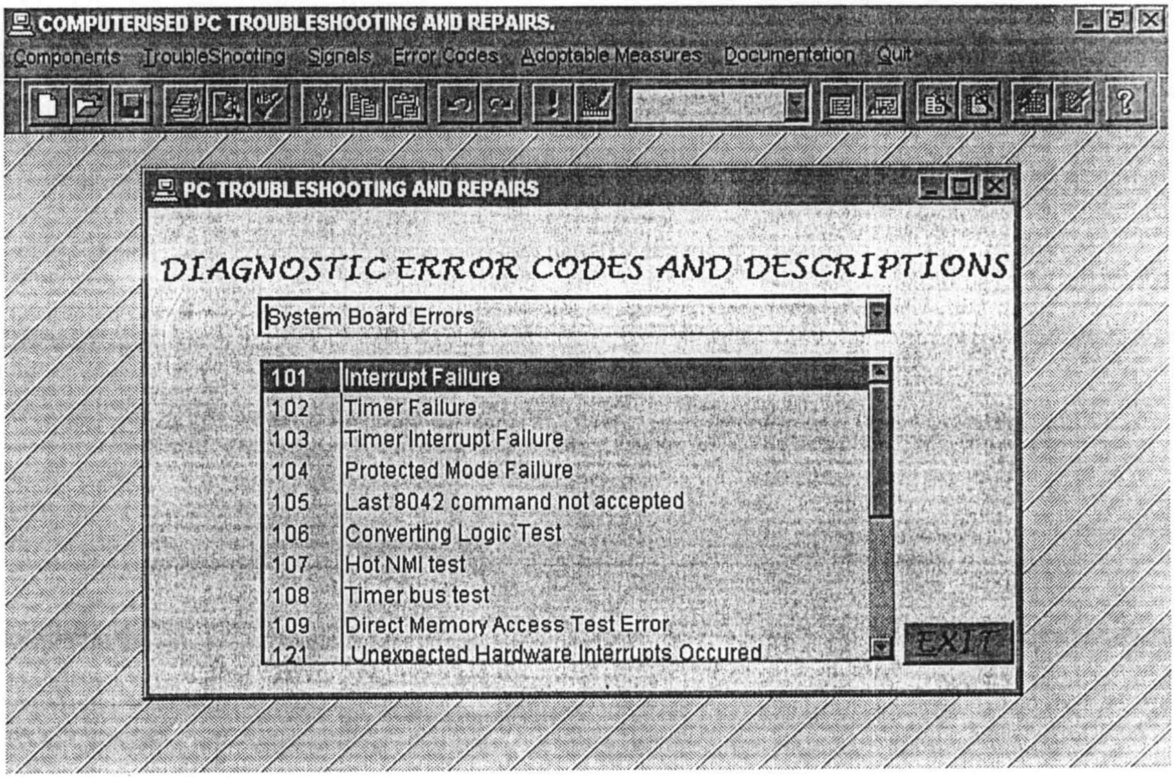
(I) Check for external signs: - If the computer has indicated lights, what do they signify. ? Are all the lights showing on the modem? Does the printer indicate "ready"? Is the hard disk drive squealling or grinding? Does the monitor image look bent? Your drives and other peripheral produce hums, whirs and clicks. After a while, these noises become familiar and any variation in them signals a problem.

One should pay attention to these signs. Since the first step is



DIAGNOSTIC SIGNALS & PROBABLE CAUSES

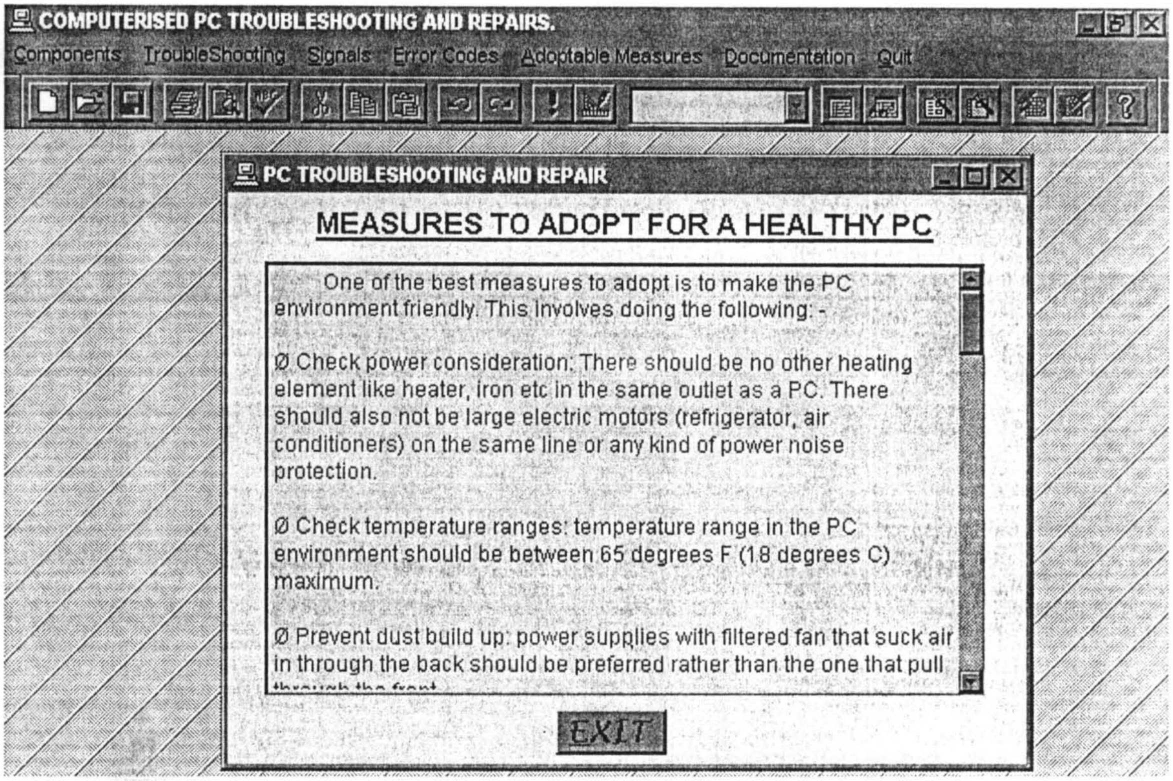
Topic	Desc
1 long beep, 2 short beep2	System board failure
1 short beep, blank screen	Failure or lack of display adapter cabl
1 short beep, no boot	Floppy drive adapter cable
Short beeps, (PS/2)	Configuration error
Error beeps	Ram, wrong connection
No boot	Power Supply, VGA, RAM
Error Codes/ Message	Corrupt Software, Incomplete Installa



DIAGNOSTIC ERROR CODES AND DESCRIPTIONS

System Board Errors

101	Interrupt Failure
102	Timer Failure
103	Timer Interrupt Failure
104	Protected Mode Failure
105	Last 8042 command not accepted
106	Converting Logic Test
107	Hot NMI test
108	Timer bus test
109	Direct Memory Access Test Error
121	Unexpected Hardware Interrupts Occured



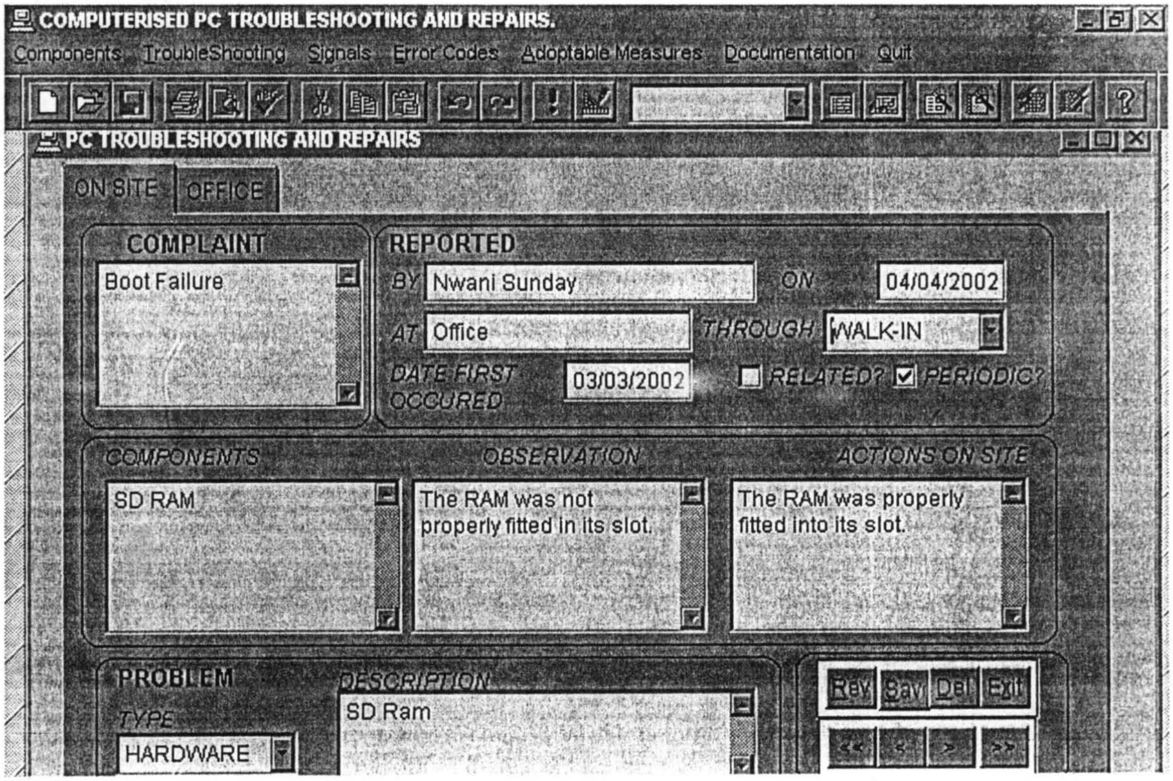
MEASURES TO ADOPT FOR A HEALTHY PC

One of the best measures to adopt is to make the PC environment friendly. This involves doing the following: -

Ø Check power consideration: There should be no other heating element like heater, iron etc in the same outlet as a PC. There should also not be large electric motors (refrigerator, air conditioners) on the same line or any kind of power noise protection.

Ø Check temperature ranges: temperature range in the PC environment should be between 65 degrees F (18 degrees C) maximum.

Ø Prevent dust build up: power supplies with filtered fan that suck air in through the back should be preferred rather than the one that pull through the front



COMPLAINT: Boot Failure

REPORTED BY: Nwani Sunday ON: 04/04/2002

AT: Office THROUGH: WALK-IN

DATE FIRST OCCURED: 03/03/2002 □ RELATED? ☑ PERIODIC?

COMPONENTS: SD RAM

OBSERVATION: The RAM was not properly fitted in its slot.

ACTIONS ON SITE: The RAM was properly fitted into its slot.

PROBLEM TYPE: HARDWARE

DESCRIPTION: SD Ram

APPENDIX II

```
************************************************
*-- Form:        form1 (c:\nwani\fchap32.scx)
*-- ParentClass:  form
*-- BaseClass:   form
*
DEFINE CLASS form1 AS form

        Top = 11
        Left = 11
        Height = 372
        Width = 617
        DoCreate = .T.
        Caption = "PC TROUBLESHOOTING AND REPAIRS"
        Icon = "pc04.ico"
        BackColor = RGB(115,185,185)
        Name = "Form1"


        ADD OBJECT pageframe1 AS pageframe WITH ;
                ErasePage = .T., ;
                PageCount = 2, ;
                TabStyle = 1, ;
                Top = 2, ;
                Left = 17, ;
                Width = 576, ;
                Height = 358, ;
                Name = "Pageframe1", ;
                Page1.Caption = "ON SITE", ;
                Page1.Name = "Page1", ;
                Page2.Caption = "OFFICE", ;
                Page2.Name = "Page2"


        ADD OBJECT form1.pageframe1.page1.shape4 AS shape WITH ;
                Top = 122, ;
                Left = 10, ;
                Height = 119, ;
                Width = 535, ;
                Curvature = 20, ;
                Name = "Shape4"


        ADD OBJECT form1.pageframe1.page1.shape1 AS shape WITH ;
                Top = 6, ;
                Left = 171, ;
                Height = 112, ;
                Width = 372, ;
                Curvature = 20, ;
                Name = "Shape1"


        ADD OBJECT form1.pageframe1.page1.shape2 AS shape WITH ;
                Top = 5, ;
                Left = 10, ;
                Height = 113, ;
        Width = 159, ;
        Curvature = 20, ;
        Name = "Shape2"


ADD OBJECT form1.pageframe1.page1.shape3 AS shape WITH ;
        Top = 247, ;
        Left = 18, ;
        Height = 74, ;
        Width = 376, ;
        Curvature = 20, ;
        Name = "Shape3"


ADD OBJECT form1.pageframe1.page1.edtcomponents AS editbox WITH ;
        Comment = "", ;
        Height = 88, ;
        Left = 23, ;
        TabIndex = 2, ;
        Top = 146, ;
        Width = 163, ;
        ControlSource = "tchap32.components", ;
        Name = "edtComponents"


ADD OBJECT form1.pageframe1.page1.lblcomponents AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "COMPONENTS", ;
        Left = 23, ;
        Top = 126, ;
        Width = 85, ;
        TabIndex = 1, ;
        Name = "lblComponents"


ADD OBJECT form1.pageframe1.page1.edtobservation AS editbox WITH ;
        Comment = "", ;
        Height = 88, ;
        Left = 191, ;
        TabIndex = 4, ;
        Top = 146, ;
        Width = 163, ;
        ControlSource = "tchap32.observation", ;
        Name = "edtObservation"


ADD OBJECT form1.pageframe1.page1.lblobservation AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "OBSERVATION", ;
        Left = 230, ;
        Top = 126, ;
        Width = 87, ;
        TabIndex = 3, ;
        Name = "lblObservation"


ADD OBJECT form1.pageframe1.page1.edtsiteaction AS editbox WITH ;
        Comment = "", ;
        Height = 88, ;
        Left = 366, ;
        TabIndex = 6, ;
        Top = 146, ;
        Width = 163, ;
        ControlSource = "tchap32.siteaction", ;
        Name = "edtSiteaction"


ADD OBJECT form1.pageframe1.page1.lblsiteaction AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "ACTIONS ON SITE", ;
        Height = 17, ;
        Left = 424, ;
        Top = 126, ;
        Width = 105, ;
        TabIndex = 5, ;
        Name = "lblSiteaction"


ADD OBJECT form1.pageframe1.page1.shape5 AS shape WITH ;
        Top = 245, ;
        Left = 402, ;
        Height = 74, ;
        Width = 149, ;
        Curvature = 20, ;
        Name = "Shape5"


ADD OBJECT form1.pageframe1.page2.shape1 AS shape WITH ;
        Top = 6, ;
        Left = 171, ;
        Height = 112, ;
        Width = 372, ;
        Curvature = 20, ;
        Name = "Shape1"


ADD OBJECT form1.pageframe1.page2.shape2 AS shape WITH ;
        Top = 5, ;
        Left = 11, ;
        Height = 113, ;
        Width = 159, ;
        Curvature = 20, ;
        Name = "Shape2"


ADD OBJECT form1.pageframe1.page2.shape3 AS shape WITH ;
        Top = 249, ;
        Left = 17, ;
        Height = 74, ;
        Width = 379, ;
        Curvature = 20, ;
        Name = "Shape3"


ADD OBJECT form1.pageframe1.page2.shape4 AS shape WITH ;
        Top = 125, ;
        Left = 11, ;
        Height = 119, ;
        Width = 535, ;
        Curvature = 20, ;
        Name = "Shape4"


ADD OBJECT form1.pageframe1.page2.txtdate_in AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tchap32.date_in", ;
        Height = 23, ;
        Left = 17, ;
        TabIndex = 2, ;
        Top = 176, ;
        Width = 71, ;
        Name = "txtDate_in"


ADD OBJECT form1.pageframe1.page2.lbldate_in AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "DATE IN", ;
        Height = 17, ;
        Left = 18, ;
        Top = 161, ;
        Width = 48, ;
        TabIndex = 1, ;
        Name = "lblDate_in"


ADD OBJECT form1.pageframe1.page2.edtshopaction AS editbox WITH ;
        Comment = "", ;
        Height = 93, ;
        Left = 98, ;
        TabIndex = 4, ;
        Top = 142, ;
        Width = 156, ;
        ControlSource = "tchap32.shopaction", ;
        Name = "edtShopaction"


ADD OBJECT form1.pageframe1.page2.lblshopaction AS label WITH ;
        AutoSize = .F., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "ACTION AT SHOP", ;
        Height = 17, ;
        Left = 98, ;
        Top = 127, ;
        Width = 108, ;
        TabIndex = 3, ;
        Name = "lblShopaction"


ADD OBJECT form1.pageframe1.page2.edtresult AS editbox WITH ;
        Comment = "", ;
        Height = 91, ;
        Left = 266, ;
        TabIndex = 6, ;
        Top = 142, ;
        Width = 156, ;
        ControlSource = "tchap32.result", ;
        Name = "edtResult"


ADD OBJECT form1.pageframe1.page2.lblresult AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "RESULT", ;
        Left = 374, ;
        Top = 127, ;
        Width = 48, ;
        TabIndex = 5, ;
        Name = "lblResult"


ADD OBJECT form1.pageframe1.page2.txtdate_out AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tchap32.date_out", ;
        Height = 23, ;
        Left = 438, ;
        TabIndex = 10, ;
        Top = 196, ;
        Width = 71, ;
        Name = "txtDate_out"


ADD OBJECT form1.pageframe1.page2.lbldate_out AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "DATE OUT", ;
        Height = 17, ;
        Left = 443, ;
        Top = 184, ;
        Width = 60, ;
        TabIndex = 9, ;
        Name = "lblDate_out"


ADD OBJECT form1.pageframe1.page2.chkreturned AS checkbox WITH ;
        Comment = "", ;
        Top = 161, ;
        Left = 426, ;
        Height = 17, ;
        Width = 93, ;
        FontItalic = .T., ;
        Caption = "RETURNED?", ;
        ControlSource = "tchap32.returned", ;
        Name = "chkReturned"


ADD OBJECT form1.pageframe1.page2.shape5 AS shape WITH ;
        Top = 246, ;
        Left = 399, ;
        Height = 74, ;
        Width = 149, ;
        Curvature = 20, ;
        Name = "Shape5"


ADD OBJECT txtwho AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tchap32.who", ;
        Height = 23, ;
        Left = 216, ;
        MaxLength = 25, ;
        TabIndex = 3, ;
        Top = 56, ;
        Width = 183, ;
        Name = "txtWho"


ADD OBJECT lblwho AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "BY", ;
        Left = 198, ;
        Top = 59, ;
        Width = 16, ;
        TabIndex = 2, ;
        Name = "lblWho"
```

```
ADD OBJECT txtwhen AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tchap32.when", ;
        Height = 23, ;
        Left = 464, ;
        TabIndex = 5, ;
        Top = 56, ;
        Width = 71, ;
        Name = "txtWhen"


ADD OBJECT lblwhen AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "ON", ;
        Left = 412, ;
        Top = 59, ;
        Width = 18, ;
        TabIndex = 4, ;
        Name = "lblWhen"


ADD OBJECT cbohow AS combobox WITH ;
        Comment = "", ;
        RowSourceType = 1, ;
        RowSource = "E-MAIL,PHONE,WALK-IN", ;
        ControlSource = "tchap32.how", ;
        Height = 24, ;
        Left = 435, ;
        TabIndex = 7, ;
        Top = 83, ;
        Width = 100, ;
        Name = "cboHow"


ADD OBJECT lblhow AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "THROUGH", ;
        Left = 369, ;
        Top = 86, ;
        Width = 61, ;
        TabIndex = 6, ;
        Name = "lblHow"


ADD OBJECT chkrelated AS checkbox WITH ;
        Comment = "", ;
        Top = 111, ;
        Left = 389, ;
        Height = 17, ;
        Width = 80, ;
        FontItalic = .T., ;
        Caption = "RELATED?", ;
        ControlSource = "tchap32.related", ;
        Name = "chkRelated"


ADD OBJECT edtcomplaint AS editbox WITH ;
        Comment = "", ;
        Height = 82, ;
        Left = 36, ;
        TabIndex = 10, ;
        Top = 55, ;
        Width = 147, ;
        ControlSource = "tchap32.complaint", ;
        Name = "edtComplaint"


ADD OBJECT lblcomplaint AS label WITH ;
        AutoSize = .T., ;
        FontBold = .T., ;
        FontSize = 10, ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "COMPLAINT", ;
        Left = 53, ;
        Top = 38, ;
        Width = 77, ;
        TabIndex = 9, ;
        Name = "lblComplaint"


ADD OBJECT lblwhere AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "AT", ;
        Left = 198, ;
        Top = 88, ;
        Width = 15, ;
        TabIndex = 11, ;
        Name = "lblWhere"


ADD OBJECT txtwhenfirst AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tchap32.whenfirst", ;
        Height = 23, ;
        Left = 293, ;
        TabIndex = 14, ;
        Top = 110, ;
        Width = 71, ;
        Name = "txtWhenfirst"


ADD OBJECT lblwhenfirst AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "DATE FIRST OCCURED", ;
        Height = 32, ;
        Left = 197, ;
        Top = 110, ;
        Width = 72, ;
        TabIndex = 13, ;
        Name = "lblWhenfirst"


ADD OBJECT chkperiodic AS checkbox WITH ;
        Comment = "", ;
        Top = 111, ;
        Left = 473, ;
        Height = 17, ;
        Width = 83, ;
        FontItalic = .T., ;
        Caption = "PERIODIC?", ;
        ControlSource = "tchap32.periodic", ;
        Name = "chkPeriodic"


ADD OBJECT label1 AS label WITH ;
        AutoSize = .T., ;
        FontBold = .T., ;
        FontSize = 10, ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "REPORTED", ;
        Height = 18, ;
        Left = 197, ;
        Top = 38, ;
        Width = 70, ;
        TabIndex = 2, ;
        Name = "Label1"


ADD OBJECT txtwhere AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tchap32.where", ;
        Height = 23, ;
        Left = 216, ;
        MaxLength = 20, ;
        TabIndex = 17, ;
        Top = 83, ;
        Width = 148, ;
        Name = "txtWhere"
```

```
ADD OBJECT cboprob_type AS combobox WITH ;
        Comment = "", ;
        RowSourceType = 1, ;
        RowSource = "HARDWARE,SOFTWARE,HUMANWARE", ;
        ControlSource = "tchap32.prob_type", ;
        Height = 24, ;
        Left = 47, ;
        TabIndex = 19, ;
        Top = 319, ;
        Width = 100, ;
        Name = "cboProb_type"


ADD OBJECT lblprob_type AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "TYPE", ;
        Left = 48, ;
        Top = 304, ;
        Width = 31, ;
        TabIndex = 18, ;
        Name = "lblProb_type"


ADD OBJECT edtprob_desc AS editbox WITH ;
        Comment = "", ;
        Height = 51, ;
        Left = 168, ;
        TabIndex = 21, ;
        Top = 296, ;
        Width = 231, ;
        ControlSource = "tchap32.prob_desc", ;
        Name = "edtProb_desc"


ADD OBJECT lblprob_desc AS label WITH ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "DESCRIPTION", ;
        Height = 17, ;
        Left = 168, ;
        Top = 284, ;
        Width = 228, ;
        TabIndex = 20, ;
        Name = "lblProb_desc"


ADD OBJECT label2 AS label WITH ;
        AutoSize = .T., ;
        FontBold = .T., ;
        FontSize = 10, ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "PROBLEM", ;
        Left = 48, ;
        Top = 281, ;
        Width = 64, ;
        TabIndex = 18, ;
        Name = "Label2"


ADD OBJECT cmdgrpeditor AS commandgroup WITH ;
        AutoSize = .T., ;
        ButtonCount = 6, ;
        BackStyle = 1, ;
        Value = 1, ;
        Height = 34, ;
        Left = 431, ;
        Top = 279, ;
        Width = 122, ;
        TabIndex = 7, ;
        BackColor = RGB(192,192,192), ;
        Name = "cmdgrpeditor", ;
        Command1.AutoSize = .F., ;
        Command1.Top = 5, ;
        Command1.Left = 5, ;
        Command1.Height = 23, ;
        Command1.Width = 28, ;
        Command1.Caption = "\<Add", ;
        Command1.Name = "cmdadd", ;
        Command2.AutoSize = .F., ;
        Command2.Top = 6, ;
        Command2.Left = 34, ;
        Command2.Height = 23, ;
        Command2.Width = 28, ;
        Command2.Caption = "\<Save", ;
        Command2.ColorScheme = 2, ;
        Command2.Name = "cmdsave", ;
        Command3.AutoSize = .F., ;
        Command3.Top = 5, ;
        Command3.Left = 61, ;
        Command3.Height = 23, ;
        Command3.Width = 28, ;
        Command3.Caption = "\<Del", ;
        Command3.Name = "cmddelete", ;
        Command4.AutoSize = .F., ;
        Command4.Top = 5, ;
        Command4.Left = 89, ;
        Command4.Height = 23, ;
        Command4.Width = 28, ;
        Command4.Caption = "E\<xit", ;
        Command4.Name = "cmdexit", ;
        Command5.AutoSize = .F., ;
        Command5.Top = 5, ;
        Command5.Left = 5, ;
        Command5.Height = 23, ;
        Command5.Width = 28, ;
        Command5.Caption = "\<Rev", ;
        Command5.Name = "cmdrevert", ;
        Command6.AutoSize = .F., ;
        Command6.Top = 5, ;
        Command6.Left = 34, ;
        Command6.Height = 23, ;
        Command6.Width = 28, ;
        Command6.Caption = "\<Mod", ;
        Command6.Name = "cmdmodify"


ADD OBJECT cmdgrpnavigator AS commandgroup WITH ;
        AutoSize = .T., ;
        ButtonCount = 4, ;
        BackStyle = 1, ;
        Value = 1, ;
        Height = 33, ;
        Left = 432, ;
        Top = 312, ;
        Width = 120, ;
        TabIndex = 8, ;
        BackColor = RGB(192,192,192), ;
        Name = "cmdgrpnavigator", ;
        Command1.AutoSize = .F., ;
        Command1.Top = 5, ;
        Command1.Left = 5, ;
        Command1.Height = 23, ;
        Command1.Width = 27, ;
        Command1.Caption = "<<", ;
        Command1.Name = "cmdtop", ;
        Command2.AutoSize = .F., ;
        Command2.Top = 5, ;
        Command2.Left = 32, ;
        Command2.Height = 23, ;
        Command2.Width = 27, ;
        Command2.Caption = "<", ;
        Command2.Name = "cmdprevious", ;
        Command3.AutoSize = .F., ;
        Command3.Top = 5, ;
        Command3.Left = 59, ;
        Command3.Height = 23, ;
        Command3.Width = 27, ;
        Command3.Caption = ">", ;
        Command3.Name = "cmdnext", ;
        Command4.AutoSize = .F., ;
        Command4.Top = 5, ;
        Command4.Left = 88, ;
        Command4.Height = 23, ;
        Command4.Width = 27, ;
        Command4.Caption = ">>", ;
        Command4.Name = "cmdbottom"


PROCEDURE Activate
```

```
        Thisform.setall("Readonly",.T.,"Textbox")
        Thisform.setall("Readonly",.T.,"Editbox")
        thisform.cmdgrpnavigator.enabled = .T.


        thisform.cmdgrpeditor.cmdadd.visible = .T.
        thisform.cmdgrpeditor.cmdmodify.visible = .T.
        thisform.cmdgrpeditor.cmdrevert.visible = .F.
        thisform.cmdgrpeditor.cmdsave.visible = .F.
        thisform.cmdgrpeditor.cmdsave.enabled = .F.

        thisform.cmdgrpeditor.cmdadd.setfocus
    ENDPROC


    PROCEDURE Load
        PUBLIC modified
        MODIFIED = .f.

        public Mwho ,mwhen ,mwhere,mhow ,mwhenfirst ,mrelated ,mperiodic,;

mprob_type,mprob_desc,mcomplaint,mcomponents,mobservation,;

msiteaction,mdate_in,mdate_out,mshopaction,mresult,mreturned

    ENDPROC


    PROCEDURE chkreturned.Click
        If this.value == .f.
            thisform.txtdate_out.enabled = .f.
        else
            thisform.txtdate_out.enabled = .t.
        endif
    ENDPROC


    PROCEDURE cboprob_type.InteractiveChange
        thisvalue = allt(this.value)
        if thisvalue == 'HARDWARE'
            Thisform.lblprob_desc.caption = 'WHICH DEVICE(S) ?'
        Endif

        if thisvalue == 'SOFTWARE'
            Thisform.lblprob_desc.caption = 'WHICH SOFTWARE/PACKAGE(S)
?'
        Endif

        if thisvalue == 'HUMANWARE'
            Thisform.lblprob_desc.caption = 'HOW DID IT HAPPEN ?'
        Endif
    ENDPROC


    PROCEDURE cmdgrpeditor.cmdadd.Click
PROCEDURE cmdgrpeditor.cmddelete.Click
        store 0 to rep1
        *rep1 = messagebox("Are you really sure ?", 36, "Want to Delete ?")
        *if rep1 = 6 then
        DELETE
        PACK
        THISFORM.REFRESH()
        *endif
        if this.parent.cmdadd.visible == .F.
                this.parent.cmdadd.visible = .T.
                this.parent.cmdrevert.visible = .F.
                this.parent.cmdsave.enabled = .F.
        endif
ENDPROC


PROCEDURE cmdgrpeditor.cmdexit.Click
        THISFORM.RELEASE()
ENDPROC


PROCEDURE cmdgrpeditor.cmdrevert.Click
        if modified == .f.
        GO BOTTOM
        DELETE
        PACK
        else
                        replace who with mwho
                        replace when with mwhen
            replace where with mwhere
                        replace how with mhow
                        replace whenfirst with mwhenfirst

                        replace related with mrelated
                        replace periodic with mperiodic
                        replace prob_type with mprob_type
                        replace prob_desc with mprob_desc
            replace complaint with mcomplaint

            replace components with mcomponents
            replace observation with mobservation
            replace siteaction with msiteaction

            replace date_in with mdate_in
            replace date_out with mdate_out
            replace shopaction with mshopaction
            replace result with mresult
            replace returned with mreturned

        endif
        Thisform.setall("Readonly",.T.,"Textbox")
        Thisform.setall("Readonly",.T.,"EDITbox")

        thisform.cmdgrpeditor.cmdadd.visible = .T.
        thisform.cmdgrpeditor.cmdmodify.visible = .T.
        thisform.cmdgrpeditor.cmdrevert.visible = .F.
        thisform.cmdgrpeditor.cmdsave.enabled = .F.


        thisform.cmdgrpnavigator.enabled = .T.

        THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpeditor.cmdmodify.Click

        Thisform.setall("Readonly",.F.,"Textbox")
        Thisform.setall("Readonly",.F.,"editbox")

        thisform.cmdgrpnavigator.enabled = .F.

        modified = .t.

                        Mwho = thisform.txtwho.value
                        mwhen = thisform.txtwhen.value
              mwhere = thisform.txtwhere.value
              mhow = thisform.cbohow.value
                        mwhenfirst = thisform.txtwhenfirst.value

                        mrelated = thisform.chkrelated.value
                        mperiodic = thisform.chkperiodic.value
                        mprob_type = thisform.cboprob_type.value
                        mprob_desc = thisform.edtprob_desc.value
              mcomplaint = thisform.edtcomplaint.value

              mcomponents = thisform.pageframe1.page1.edtcomponents.value
              mobservation = thisform.pageframe1.page1.edtobservation.value
              msiteaction = thisform.pageframe1.page1.edtsiteaction.value

              mdate_in = thisform.pageframe1.page2.txtdate_in.value
              mdate_out = thisform.pageframe1.page2.txtdate_out.value
              mshopaction = thisform.pageframe1.page2.edtshopaction.value
              mresult = thisform.pageframe1.page2.edtresult.value
              mreturned = thisform.pageframe1.page2.chkreturned.value

        thisform.cmdgrpeditor.cmdadd.visible = .F.
        thisform.cmdgrpeditor.cmdmodify.visible = .F.
        thisform.cmdgrpeditor.cmdrevert.visible = .T.
        thisform.cmdgrpeditor.cmdsave.visible = .T.
        thisform.cmdgrpeditor.cmdsave.enabled = .T.


        THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpnavigator.cmdtop.Click
        GOTO TOP
        THISFORM.REFRESH()
```

```
    *sele proj1
    APPEND BLANK

    modified = .f.
    Thisform.setall("Readonly",.F.,"Textbox")
    Thisform.setall("Readonly",.F.,"combobox")
    Thisform.setall("Readonly",.F.,"EDITbox")
    *thisform.txtyear.readonly = .T.
    thisform.cmdgrpnavigator.enabled = .F.

    thisform.cmdgrpeditor.cmdadd.visible = .F.
    thisform.cmdgrpeditor.cmdmodify.visible = .F.
    thisform.cmdgrpeditor.cmdrevert.visible = .T.
    thisform.cmdgrpeditor.cmdsave.visible = .T.
    thisform.cmdgrpeditor.cmdsave.enabled = .T.

    *
    THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpeditor.cmdsave.Click
            replace who with thisform.txtwho.value
            replace when with thisform.txtwhen.value
      replace where with thisform.txtwhere.value
            replace how with thisform.cbohow.value
            replace whenfirst with thisform.txtwhenfirst.value

            replace related with thisform.chkrelated.value
            replace periodic with thisform.chkperiodic.value
            replace prob_type with thisform.cboprob_type.value
            replace prob_desc with thisform.edtprob_desc.value
      replace complaint with thisform.edtcomplaint.value

      replace components with thisform.pageframe1.page1.edtcomponents.value
      replace observation with thisform.pageframe1.page1.edtobservation.value
      replace siteaction with thisform.pageframe1.page1.edtsiteaction.value

      replace date_in with thisform.pageframe1.page2.txtdate_in.value
      replace date_out with thisform.pageframe1.page2.txtdate_out.value
      replace shopaction with thisform.pageframe1.page2.edtshopaction.value
      replace result with thisform.pageframe1.page2.edtresult.value
      replace returned with thisform.pageframe1.page2.chkreturned.value

    thisform.cmdgrpeditor.cmdadd.visible = .T.
    thisform.cmdgrpeditor.cmdrevert.visible = .F.

    thisform.cmdgrpeditor.cmdmodify.visible = .T.

    thisform.cmdgrpnavigator.enabled = .T.
    Thisform.setall("Readonly",.T.,"Textbox")
    Thisform.setall("Readonly",.T.,"EDITbox")
ENDPROC
```

```
    this.parent.cmdprevious.enabled = .F.
    this.enabled = .F.

    this.parent.cmdnext.enabled = .T.
    this.parent.cmdbottom.enabled = .T.
ENDPROC


PROCEDURE cmdgrpnavigator.cmdprevious.Click
    IF !BOF()
        SKIP -1
        this.parent.cmdbottom.enabled = .T.
        this.parent.cmdnext.enabled = .T.
        IF BOF()
            GO TOP
            this.parent.cmdtop.enabled = .F.
            this.enabled = .F.

            this.parent.cmdnext.enabled = .T.
            this.parent.cmdbottom.enabled = .T.
        else

            this.parent.cmdtop.enabled = .T.
            this.enabled = .T.
        ENDIF
    ENDIF
    THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpnavigator.cmdnext.Click
    IF !EOF()
        SKIP
        this.parent.cmdtop.enabled = .T.
        this.parent.cmdprevious.enabled = .T.
        IF EOF()
            GO BOTTOM
            this.parent.cmdbottom.enabled = .F.
            this.enabled = .F.

            this.parent.cmdprevious.enabled = .T.
            this.parent.cmdtop.enabled = .T.
        ENDIF
    ENDIF
    THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpnavigator.cmdnext.MouseDown
    LPARAMETERS nButton, nShift, nXCoord, nYCoord
ENDPROC
```

```
        PROCEDURE cmdgrpnavigator.cmdbottom.Click
                GO BOTTOM
                THISFORM.REFRESH()


                this.parent.cmdnext.enabled = .F.
                this.enabled = .F.

                this.parent.cmdtop.enabled = .T.
                this.parent.cmdprevious.enabled = .T.
        ENDPROC


ENDDEFINE
*
*-- EndDefine: form1
**************************************************
```