ThunderBYTE

Anti-Virus Utilities

USER MANUAL

The ThunderBYTE Anti-Virus Utilities are a product of:

ESaSS B.V.
P.O. Box 1380
6501 BJ NIJMEGEN
The Netherlands

Table of Contents

Introduction

A Word (or Two) of Thanks

Congratulations! By purchasing the ThunderBYTE Anti-Virus utilities you have taken the basic step in building a massive anti-viral safety wall around your precious computer system. Setting up the appropriate defense using the TBAV utilities is a personal matter. Therefore, we recommend to read this manual thoroughly, so you are well aware of the different kinds of security measures you can take.

What Are the TBAV Utilities?

ThunderBYTE Anti-Virus (TBAV) is a comprehensive tool kit designed to protect against, and recover from, computer viruses. While TBAV focuses heavily on numerous ways to prevent a virus infection, the package would not be complete without various cleaner programs to purge a system, in the unlikely event that a virus manages to slip through. The package, therefore, consists of several programs, each of which helps you to prevent viruses from accomplishing their destructive purposes. Here is a quick overview.

TbSetup: Collecting Software Information

TbSetup is a program that collects information from all software it finds on your system. It places this information in files named ANTI-VIR.DAT and uses it for integrity checking, program validation, and cleaning infected files.

TbDriver: Enable Memory Resident TBAV Utilities

While TbDriver provides little protection against viruses by itself, you must load it in advance to enable the memory resident ThunderBYTE Anti-Virus utilities to perform properly. These utilities include: TbScanX, TbCheck, TbMem, TbFile, and TbDisk. TbDriver also provides basic protection against ANSI bombs and stealth viruses.

TbScan: Scanning for Viruses

TbScan is both a fast signature scanner and a so-called heuristic scanner. Besides its blazing speed, it has many configuration options. It can detect mutants of viruses, bypass stealth type viruses, etc. The signature file TbScan uses is a coded TBSCAN.SIG file, which you can update yourself in case of emergency.

TbScan will disassemble files. This makes it possible to detect suspicious instruction sequences and detect yet unknown viruses. As pointed out earlier, this generic detection, named heuristic analysis, is a technique that makes it possible to detect about 90% of all viruses by searching for suspicious instruction sequences rather than relying on any signature. For that purpose TbScan has a built-in disassembler and code analyzer.

Another feature of TbScan is the integrity checking it performs when it finds the ANTI-VIR.DAT files generated by TbSetup. Integrity checking means that TbScan verifies that every file it scans matches the information which was captured when the file was first analyzed by TbSetup and is maintained in the ANTI-VIR.DAT files. If a virus infects a file, the information in the ANTI-VIR.DAT file will indicate that the file has been changed, and TbScan will inform you of this. TbScan performs an integrity check automatically, and it does not have the false alarm rate other integrity checkers have. The goal is to detect viruses and NOT to detect configuration changes!

TbScanX: Automatic Scanning

TbScanX is the memory resident version of TbScan. This signature scanner remains resident in memory and automatically scans those files that are being executed, copied, de-archived, downloaded, etc. TbScanX does not require much memory. It can swap itself into expanded, XMS, or high memory, using only one kilobyte of conventional memory.

TbCheck: Check While Loading

TbCheck is a memory resident integrity checker that remains resident in memory and automatically checks every file just before it executes. TbCheck uses a fast integrity checking method, which consumes only 400 bytes of memory. You can configure it to reject files with incorrect checksums, and/or reject files that do not have a corresponding ANTI-VIR.DAT record.

TbUtil: Restoring Infected Boot-Sector, CMOS and Partition Tables

Some viruses copy themselves into the hard disk's partition table, which makes them far more difficult to remove than boot sector viruses. Performing a low-level format is an effective, but rather drastic measure.

TbUtil offers a more convenient alternative by making a precautionary backup of uninfected partition tables and the boot sector. If an infection occurs, you can use the TbUtil backup as a verifying tool and as a means to restore the original (uninfected) partition table and boot sector, without the need for a destructive disk format. TbUtil can also restore the CMOS configuration for you. If a backup of your partition table is not available, TbUtil tries to create a new partition table anyway, again avoiding the need for a low-level format.

Another important feature of TbUtil is the option to replace the partition table code with new code offering greater resistance to viruses. TbUtil executes the partition code BEFORE the boot sector gains control, enabling it to check this sector in a clean environment. The TbUtil partition code performs a CRC calculation on the master boot sector just before the boot sector code activates and issues a warning if the boot sector has been modified. The TbUtil partition code also checks and reports changes in the RAM layout. It performs these checks whenever the computer boots from the hard disk.

We should point out that boot sector verification is imperative before allowing the boot sector code to execute. A virus could easily become resident in memory during boot-up and hide its presence. TbUtil offers total security at this stage by being active before the boot sector executes. TbUtil is far more convenient than the traditional strategy of booting from a clean DOS diskette for an undisturbed inspection of the boot sector.

TbClean: Reconstructing Infected Files

TbClean is a generic file cleaning utility. It uses the ANTI-VIR.DAT files generated by TbSetup to enhance file cleaning and/or to verify the results. TbClean can also work without these files. It disassembles and emulates the infected file and uses this analysis to reconstruct the original file.

TbMem, TbFile and TbDisk: Resident Safeguards

The TBAV utilities include a set of memory resident anti-virus utilities, consisting of TbMem, TbFile and TbDisk. Most other resident anti-virus products offer you the choice to either invoke them before the network loads (thereby losing the protection after the logon procedure), or to load the anti-viral software after logging onto the network, resulting in a partially unprotected system. The TBAV utilities, on the other hand, recognize the network

software and utilize their auto-configuration capabilities to ensure their continued functionality.

TbMem: Safeguarding Memory

TbMem detects attempts from programs to remain resident in memory and ensures that no program can remain resident in memory without permission. Since most viruses remain resident in memory, this is a powerful weapon against all such viruses, known or unknown. TbMem also protects your CMOS memory against unwanted modifications. The ANTI-VIR.DAT files maintain a database of the permission information.

TbFile: Executable File Protection

TbFile detects attempts from programs to infect other programs. It also guards read-only attributes, detects illegal time-stamps, etc. It ensures that no virus succeeds in infecting programs.

TbDisk: Protecting The Disk

TbDisk is a disk guard program that detects attempts from programs to write directly to disk (that is, without using DOS), attempts to format, etc., and makes sure that no malicious program succeeds in destroying your data. This utility also traps tunneling and direct calls into the BIOS code. The ANTI-VIR.DAT files maintain permission information about those rare programs that write directly to and/or format the disk.

TbGenSig: Define Your Own Signatures

Since TBAV includes an up-to-date, ready-to-use signature file, you do not really need to maintain a signature file yourself. If, however during a crisis, you need to define your own virus signatures, then the TbGenSig utility enables you to do this. You can use either published signatures or define your own if you are familiar with the structure of computer code.

TbDel: Remove Infected Files

The DOS DEL or ERASE command does not actually erase a file. It simply deletes the first filename character in the directory listing and frees up the space by changing the disk's internal location tables (File Allocation Tables). TbDel is a small program with a single, yet all-important purpose: it overwrites every single byte

in a file with the zero character (0) before deleting it, thereby obliterating all the data and making it totally unrecoverable.

TbMon: Installed Device Checker

To check for the presence of the resident TBAV utilities (TbScanX, TbCheck, TbMem, TbFile, TbDisk or TbLog) in batch files or login scripts, you can use the TbMon utility. TbMon returns a DOS error level, depending on the installed ThunderBYTE resident programs.

The following list specifies the ThunderBYTE resident utilities and their respective error levels:

Utility Name	Error level
TbScanX	1
TbCheck	2
TbMem	4
TbFile	8
TbDisk	16
TbLog	32

The error level returned by TbMon is the cumulative sum of the error levels of the installed devices. For example, if you have TbScanX and TbMem installed, TbMon will return error level 5 (1+ 4 = 5). Another example: if you have all utilities loaded, TbMon will return error level 63 (1+2+4+8+16+32=63). If none of the resident ThunderBYTE utilities are installed, TbMon will return error level 0 (zero).

The TBAV Utilities User Interface

The DOS version of TBAV utilizes a menu-driven interface that enables you to execute the utilities easily. You can also execute many of the utilities directly from the DOS prompt. One advantage to this is that you can use the utilities in batch files.

The Microsoft Windows version of TBAV utilizes the standard Windows interface, providing you a way to protect yourself from viruses while still working in the user-friendly Windows environment. TBAV-for-Windows is not described in this document. Please refer to the TBAV-for-Windows documentation for more information.

Conventions Used in This Manual

This manual uses several special conventions:

References to the keyboard are as they appear on the 101-key enhanced keyboard. File names, DOS commands, emphasized words, and information that you are to type appears in UPPERCASE letters. The context should clearly dictate which of these is true in each case.

References to individual TBAV utilities use a combination of uppercase and lowercase letters. For example, while TBSCAN.SIG refers to a signature file, TbScan refers to the utility itself.

How To Use This Manual

This manual consists of six chapters.

Chapter 1 provides you with the fastest way to get started with the TBAV utilities. It presents the major features of the program in a step-by- step format. We recommend that you start with this chapter.

Chapter 2 contains instruction on how to prevent viruses from infecting your computer system and directions on how to handle viruses when they do strike. We recommend that you also read this chapter because it contains several useful tips.

Chapter 3 contains a detailed description of both the purpose and functionality of all the TBAV for DOS utilities.

Chapter 4 contains advanced user information for those users who are more technically oriented.

This manual also contains five appendices. Appendix A describes TBAV messages, Appendix B describes heuristic flags, Appendix C addresses some incompatibility problems, Appendix D lists various exit codes for use in batch files, and Appendix E contains information on naming viruses. Finally, the Index provides you with the means of quickly finding any major topic.

NOTE:

A complete reading of this manual is indispensable in order to become familiar with the many facets of the ThunderBYTE AntiVirus utilities; to know what steps you can, and must, take to ensure

adequate protection and be fully prepared for a complete recovery, if and when disaster strikes.

1 TBAV QuickStart

One of the problems with software manuals is they sometimes beat around the bush and don't get to the point, namely, how to use the software right now. This chapter presents the major features of TBAV and will get you up and running in the minimum amount of time.

1.1 Installing the TBAV Utilities

This section provides the initial installation instructions of the TBAV utilities for DOS. See the TBAV for Windows documentation for installing TBAV for Windows or the TBAV for Networks documentation for installing TBAV for Networks.

1.1.1 Understanding System requirements

The ThunderBYTE Anti-Virus utilities will run on any IBM or compatible PC that meets the following requirements:

At least 1 megabyte of disk space
256 kilobytes of free internal memory

DOS version 3.0 (DOS 5.0 or later recommended)

A mouse is optional

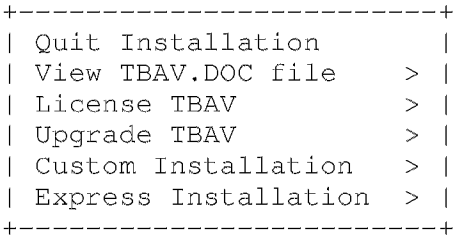
NOTE:
The TBAV utilities are compatible with networks, MS-Windows, Novell-DOS, etc.

1.1.2 Running INSTALL

You can install the TBAV utilities either by using the following installation procedure or by a fully customized procedure that you ll find in Chapter 2. To use the fast approach, follow these steps:

1. Insert the TBAV installation diskette in the diskette drive, type A: or B:, and press the ENTER key.

2. Type INSTALL and press ENTER. After a few seconds, the following window appears:



3. Since this is your first time to install the TBAV package you choose the first option, which is already highlighted, so just press ENTER. Notice also that you can always select a menu option by pressing its first letter. Install now displays the Licensing Agreement.

4. Press the cursor movement keys (up and down arrows and Page Up and Page Down) to view the Agreement. When you finish reading the agreement, press ESC. Install now asks you to acknowledge the Agreement.

NOTE:
You can exit Install at anytime by pressing the ESC key until you get to the Main Menu or even to the DOS prompt.

5. Select the Your Name field, type in your name, and press ENTER.

6. Select the company field and repeat the procedure to enter your company name.

7. Press I to select the Terms field, type in YES to accept the agreement, and press ENTER. The Install Menu now appears.

8. While you will probably accept the defaults, if you need to change the source path (the path where the installation program itself resides, usually drive A:) or the default Destination path (where Install places the TBAV program files, usually C:\TBAV), select the field, make your changes, and press ENTER.

9. Press B (or highlight Begin Installation and press ENTER) to begin the installation. Install now scans your system to ensure that it is clean (that is, no files are infected by a virus) and informs you when it is done.

10. Press any key to continue. Install now copies the TBAV files to the destination directory and makes a backup of your AUTOEXEC.BAT file before making a few modifications to it. The installation

program adds the TBAV directory to your PATH and adds a statement that will automatically run the TBSTART.BAT file.

NOTE:
The TBSTART.BAT file, which resides in the TBAV directory, contains the following commands:

```
C:\TBAV\TBDRIVER
C:\TBAV\TBSCANX
C:\TBAV\TBCHECK
C:\TBAV\TBMEM
C:\TBAV\TBFILE
C:\TBAV\TBSCAN  ONCE  ALLDRIVES
```

You can configure these commands to suit your own personal needs.

Notice:
Install now displays a message that Recommends that you create a Recovery Diskette, which you can use in the future, for example, to restore your destroyed CMOS data, or restore your hard disk's partition table after it has been tampered with.

11. Press any key to continue to the Final Menu. To create a Recovery Diskette, press M, insert a clean formatted diskette into Drive A, and press any key to continue. TBAV now copies the system files to the diskette. See the Prepare a Recovery Diskette section in Chapter 2 for more information. If you do not want to create a Recovery Diskette, press Q to Quit Install.

12. When TBAV finishes, press any key to continue. TBAV invokes TbSetup to generate an ANTI-VIR.DAT file for drive A and returns you to the Final Menu.

13. Press Q to Quit Install. Install now invokes TbSetup again to generate the ANTI-VIR.DAT reference files for your hard disk and then returns you to the DOS prompt.

CAUTION:
It is extremely likely that some of the TBAV utilities are going to display messages if you now reboot and continue using the computer as you normally would. This is because some programs perform operations that the TBAV utilities monitor. TBAV, therefore, needs to learn which programs need proper permission. Before rebooting, execute some of the programs you use regularly and respond appropriately when TBAV requests permission to authorize or deny

their use. TBAV remembers the settings and will not bother you again. Reboot the computer at the end of this test run.

14. After running some of the programs you use regularly (see Caution box above), reboot your system.

The TBAV utilities are now ready to monitor your system and will issue a warning if something suspicious (or worse!) is about to happen. The TBAV utilities also warn you if any new file contains a possible virus, well before it can do any harm.

1.1.3 Installation on a network

If a workstation does not have a hard disk, you can invoke the TBAV utilities from a login script. You create a TbStart.Bat file containing the following:

```
@echo off
x:\apps\tbav\tbdriver.exe
x:\apps\tbav\tbscanx.exe
x:\apps\tbav\tbcheck.exe
x:\apps\tbav\tbfile.exe
x:\apps\tbav\tbmem.exe
x:\apps\tbav\tbscan.exe alldrives
exit
```

In the login script add the following line:

```
#x:command.com /c /x:\apps\tbav\tbstart.bat
```

NOTE:
You need to enter the correct drive ID for 'X:'!

1.1.4 Starting And Ending TBAV

You can run TBAV in two ways: run the menu interface or run individual utilities from the DOS prompt.

Starting TBAV With the Menu Interface

You can access most of the TBAV utilities from within the TBAV menu. To start TBAV with the menu, follow these steps:

1. At the DOS prompt, type CD \TBAV and press ENTER. This places you in the TBAV directory.

NOTE:
This first step is actually optional since the TBAV directory was added to the PATH during installation. You would need this step, however, if you ever decided to remove that directory from the PATH.

2. Type TBAV and press ENTER. This starts TBAV and displays the menu interface.

3. A common task is to scan your hard disk for viruses. To do this, press S on the "Main Menu" to select the TbScan command. Press S again to select the "Start Scanning" command on the TbScan Menu. Press D on the "Path Menu" and press ENTER.

4. If TbScan finds a virus, it presents an action menu. "D)elete" deletes the infected file. "K)ill" also deletes the infected file, but in such a way that it can't be undeleted by an undelete utility (such as DOS's UNDELETE command). "R)ename" renames an EXE extension to VXE and a COM extension to VOM, preventing the execution of infected programs and thereby precluding the spread of an infection, and also enabling you to keep the file for later examination and repair. "C)ontinue scanning" continues the scan without taking action on the virus. "N)onstop continue" instructs TbScan not to stop when it detects a virus.

NOTE:
If you use C or N, we recommend that you select L on the "TbScan Menu" and then O on the "TbScan Log Menu" so that TbScan will log detected viruses. To view this log, select V from the "TbScan Menu."

5. Another common task is to scan a diskette. To scan a diskette in drive A, press A, or to scan a diskette in drive B, press B.

6. You can use one of three methods to end TBAV:

Press X to exit and save any configuration settings you have set
Press Q to exit without saving any configuration settings

Press ESC, which is the same as pressing Q

Starting TBAV Utilities from the DOS Prompt

You can also start each of the individual TBAV utilities directly from the DOS prompt by typing the command name followed by one or more options (or switches) to control special features. You can use either the full name of the option or its one- or two-letter mnemonic to shorten the command line.

For example, if you want to use TbScan to scan for viruses on your hard disk, you could execute either one of the following commands:

```
TBSCAN ALLDRIVES
TBSCAN AD
```

The advantage of being able to execute individual utilities is that you can use the utilities in batch files to create your own custom routines. A simple example of this is putting TbScan in your AUTOEXEC.BAT file so that it will scan for viruses when you boot up. To accomplish this, do the following:

1. If you are using DOS 5 or later, type CD\ and press ENTER to go to the root directory. Now type EDIT AUTOEXEC.BAT and press ENTER to load this file into the MS-DOS text editor Edit.

NOTE:
If you are using a version of DOS prior to version 5.0, consult your DOS manual on how to edit AUTOEXEC.BAT. You might have your own text editor that you can use, or you could even use a word processor to edit the file and then save it as an ASCII text file. Consult your word processor's documentation for instructions.

2. Add the following line to the beginning of the file, making sure you separate the options from the command and from each other using a space:

```
C:\TBAV\TBSCAN AllDrives Once
```

3. Press ALT, F, S to save the file again, and then press ALT, F, X to exit the editor (that is, if you are using the MS-DOS text editor EDIT; otherwise, use the commands of your favourite editor to save the file, and to exit the editor).

4. Reboot your computer so the changes will take effect.

CAUTION:

This line already exists in the TBSTART.BAT file, which runs automatically from AUTOEXEC.BAT. If you don't want to load all the TSR utilities that TBSTART.BAT loads, you could replace TBSTART.BAT with the above TBSCAN command. While this is still good protection, be aware that it doesn't fully protect your system. Refer to the Configuring TBAV section later in this chapter for more information on configuring TBAV.

Now the first time you boot your computer on a given day, TbScan will check for viruses on all fixed drives. Because of the OO option, however, if you boot again, you'll receive the Option once already used today message, meaning that since TbScan has already run once that day, it will not run again.

Another useful TBAV utility, not just for deleting infected files but any files you want destroyed, is TbDel. This utility overwrites every byte of a file with a nul character, thereby completely obliterating the file. If, for security reasons, you have files you want to destroy and prevent someone from undeleting using a file recovery program, enter the following command:

TBDEL [filename]

WARNING:
Be absolutely sure you want to destroy a file before using TbDel. Once you execute the command, the file is gone forever, and no file recovery utility can bring it back.

1.1.5 Using TBAV Commands

There are many commands in The TBAV Utilities, but most of them are available from the menu. You can select commands using either the keyboard or the mouse. To select a command, do one of the following:

Highlight an option using the arrow keys and press Enter

Press the highlighted letter of a command

Move the mouse pointer to a command and click the left button

As mentioned earlier, you can use all TBAV commands directly from the DOS prompt. You must separate the command from the first option and options from each other using a space. You can use the standard slash (/) character or hyphen (-) before an option, but it is not necessary.

The standard command line syntax for all ThunderBYTE Anti-Virus commands is:

COMMAND [<path>][<filename>] [<option>] [<option>]

where <path> and <filename> is where you want the command to execute and <option> is the specific option you want to use. For example, the following command executes a virus scan on all executable files in the root directory of drive C: and all subdirectories and skips the boot sector scan:

TBSCAN C:\ NOBOOT

1.1.6 Getting Help

TBAV enables you to get help at any time, whether you are working from the menu or the DOS prompt.

Getting Help From the Menu

To get help at anytime while working from the TBAV menu, follow these steps:

1. From the Main Menu, select Documentation.

2. From the Documentation menu, select TBAV User Manual.

3. Use the up and down arrow keys and Page Up and Page Down to move through the manual.

4. Press ESC to exit the manual.

TIP:
Instead of using the internal file viewer to view the User Manual, you can substitute your own favorite viewer. See the Configuring TBAV section later in this chapter for details.

Getting Help at the DOS Prompt

To get help about proper syntax when working with individual TBAV utilities, do one of the following:

Type the name of the command followed by a question mark (?), TBSCAN ?, for example. Some commands (TbClean, TbDel, and TbUtil) display the Help screen if you type the command name only.

Each command also displays the help screen if you issue the command with an invalid option.

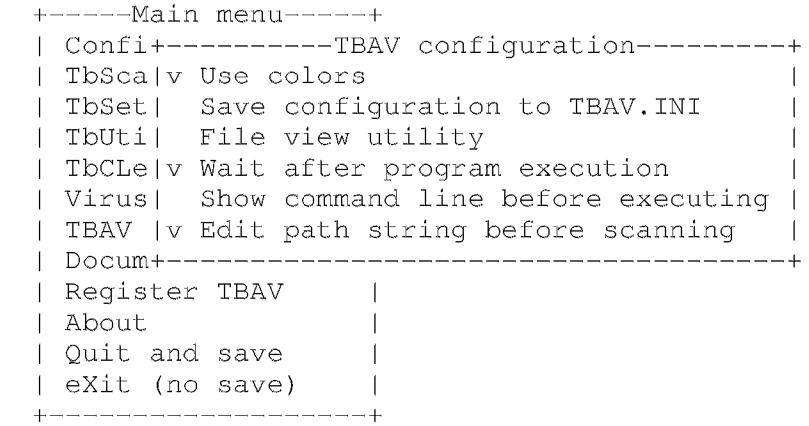
1.1.7 Configuring TBAV

The choices you made when installing the TBAV utilities might need a little fine tuning. You might want to edit AUTOEXEC.BAT, as mentioned earlier, for example, or you might want to edit TBSTART.BAT file, which AUTOEXEC.BAT executes.

Additionally, you might want to change how TBAV operates within the menu interface. This section explains how you can configure the TBAV utilities and use them the way you prefer. The following sections explain how to customize TBAV.

NOTE:
After making certain changes and then initializing and rebooting your system, TBAV needs to be "trained" as it encounters new TSR's.

NOTE:
Options that have a check mark beside them indicate that they are selected. Options may be toggled by selecting: the highlighted letter, clicking on them with your mouse or moving the highlight bar with your cursor keys and then pressing Enter.



The "Use Colors" Option

If you disable this option, that is, select it so the check mark disappears, TBAV appears in monochrome mode, which is convenient for use on laptop and notebook computers. When you select the Configure TBAV option from the Main Menu, the Configuration menu appears:

The "Save Configuration to TBAV.INI" Option

When you select this option, TBAV saves all configuration values set within the TBAV menu in the TBAV.INI file. The next time you load the TBAV utilities, these configuration values take effect. These values apply to the TBAV menu itself and the utilities TbSetup, TbScan and TbClean.

Although you can edit the TBAV.INI file manually, we recommend that you allow the TBAV menu to do it. By default, the contents of the TBAV.INI file are valid only while using the TBAV menu shell. You can, however, enable the Use TBAV.INI file options (or specify the USEINI switches in the TBAV.INI file itself) for each of the TBAV utilities.
For example, to use the settings in TBAV.INI with TbScan, you would follow these steps:

1. Select TbScan from the Main Menu. This displays the TbScan Menu.

2. From this menu, select the Options Menu option.

3. From this menu, select the Use TBAV.INI option and notice that a check mark appears beside it.

After selecting this option, TbScan also uses the TBAV.INI when you run TbScan from the DOS prompt. The same is true if you select this option for TbSetup and TbClean.

CAUTION:
Be careful, since command line options do NOT undo TBAV.INI settings. TBAV creates a TBAV.INI file when enabling this option for the first time. This file lists all valid configuration switches. Additionally, a semicolon precedes disabled switches.

The "File View Utility" Option

TbSetup and TbScan generate a data file and a log file respectively. By default, you can view these files, as well as the TBAV documentation mentioned earlier, from the TBAV menu using TBAV's internal file view utility.

If you prefer, however, you can specify your own file viewing utility. To do this, follow these steps:

1. Press F to select the File View Utility option.

2. Type in the complete path and the file name, including the extension, of the utility you want to use (e.g., C:\DIRNAME\VIEWER.EXE), and press ENTER.

The "Wait After Program Execution" Option

If you enable this option, TBAV displays the message "Press any key to return to the TBAV menu..." after executing an external utility.

The "Show Command Line Before Executing" Option

Enabling this option forces TBAV to display the DOS command that loads the external file viewing utility. This option comes in handy for enabling you to see the command(s) you specified before. After pressing ENTER, TBAV then executes the DOS commands.

The "Edit Path String Before Scanning" Option

If you enable this option, TBAV prompts you to edit or confirm the path to scan after you select Start Scanning from a scan menu.

1.2 Understanding TbSetup

By way of analogy, if you think of TbScan as being the heart of TBAV, you can think of TbSetup as being the skeleton. TbSetup collects information from all software it finds on your system and places this information in files, one in each directory, named ANTI-VIR.DAT and uses this information for integrity checking, program validation, and cleaning infected files.

WARNING:
NEVER, NEVER, NEVER use TbSetup when there is the slightest evidence of a virus on your system.

Since TbSetup was run during the installation program, it is not really necessary for you to run it again. In fact, the less you run it the better. The only time you should run TbSetup again is in directories with

new or changed program files. Assume you just added a new program to your system, which installed into a new directory called NEWPRO. To run TbSetup on that new directory, you could execute one of the following procedures:

From the TBAV Main Menu, select TbSetup, select Start TbSetup from the TbSetup Menu, type in C:\NEWPRO as the path to process, and then press ENTER.

From the DOS prompt, enter TBSETUP C:\NEWPRO and press ENTER.

See the "Using TbSetup" section in Chapter 3 for more information about using TbSetup.

WARNING:

NEVER, NEVER, NEVER use TbSetup when there is the slightest evidence of a virus on your system.

1.3 Understanding TbDriver

TbDriver is a small memory resident (TSR) program that you must load before loading any of the other five TBAV memory resident programs, which include: TbScanX, TbCheck, TbMem, TbFile, and TbDisk. Chapter 3 fully explains all of these programs, but to conclude our earlier analogy, if TbScan is the heart of TBAV, and TbSetup is the skeleton, then TbDriver and the other TSRs are the muscles. They simply wait in memory until called into action. When they detect suspicious code or other irregularities, they immediately inform you and take appropriate action.

TBAV Install places a call to TBSTART.BAT in your AUTOEXEC.BAT file so that all of these TSRs, except TbDisk, load automatically when you boot. For maximum security, we recommend that you allow these utilities to load and remain in memory.

TIP:

If you prefer, you can put the memory resident utilities listed in TBSTART.BAT in your CONFIG.SYS file. Remove the call to TBSTART.BAT from AUTOEXEC.BAT, and then use a DEVICE= command in CONFIG.SYS for each utility. Don't forget to use the full path and to specify the .EXE extension. If you are using DOS 5 or higher, you can load these utilities into upper memory using the LOADHIGH command in either TBSTART.BAT or CONFIG.SYS.

1.4 Maintaining the System

All systems need maintenance, and the TBAV utilities are no different. This section, therefore, describes how to maintain the TBAV utilities.

1.4.1 Maintaining ANTI-VIR.DAT Files

Whenever you add, update or replace programs on your system, be sure to use TbSetup to generate or update their fingerprints in the ANTI-VIR.DAT files. See the Using TbSetup section earlier in this chapter and the Using TbSetup section in Chapter 3 for more information.

1.4.2 Creating a New Recovery Diskette

There will be times when you will want to create a new recovery diskette. This will be necessary, for example, when you install a new version of DOS because this changes the boot sector. You should also do this if you change the configuration of your hard disk because this can affect the partition tables and the CMOS setup. You should prepare a new recovery diskette after all system modifications. See the Prepare a Recovery Diskette section in the next chapter for more information.

1.4.3 Getting Updates

As new viruses emerge, which is almost daily, you need to replace TbScan's signature file (TBSCAN.SIG) periodically with a more up to date one. You can get the latest signature file from your local ThunderBYTE dealer. Subscribing to the ThunderBYTE update service at your local dealer is a convenient way to guarantee the delivery of each new update.

You can also download the file directly from the ThunderBYTE support Bulletin Board Systems (BBS).

To download updates, follow these steps:

1. Using your telecommunications program, dial the BBS phone number.

2. When the modem logs on, press the ESC twice to go to the ThunderBYTE On-line Service.

3. From the File Menu select Download Latest ThunderBYTE Anti-Virus Utilities .

4. Select the File Transfer Protocol "zmodem" or "ymodem" from the Protocol Menu to select the file protocol you want to use, and then begin your download procedure.

Additionally, you can check with a local bulletin board regularly, as many of them offer updated versions of our software.

We issue the standard complete release in an archive named: TBAVxxx.ZIP, where xxx represents the three-digit version number. The archive extension might vary on local bulletin boards using a different archive method.

The release of TBAV for Windows is archived in a file named: TBAVWxxx.ZIP. Again, xxx represents the three-digit version number of TBAV for Windows. The same holds for the release of TBAV for Networks; it is distributed in a file called TBAVNxxx.ZIP.

To maintain the highest reliability, the Dutch and US ThunderBYTE support sites issue regular beta releases, also containing only the files that have changed. You can identify beta versions by a B in the filename, such as TBAVBxxx.ZIP.

The resident ThunderBYTE Anti-Virus utilities are also available in processor optimized formats. These processor optimized versions, named TBAVXxxx.ZIP, are for registered users only. You can buy these versions through your local ThunderBYTE dealer.

NOTE:
The ThunderBYTE Anti-Virus utilities currently support several languages, by means of separate language files. Check your local ThunderBYTE dealer for the availability of the TBAV support file in your language.

1.4.4 Maintaining a Network

Since you should replace the signature file TBSCAN.SIG frequently, this can turn into much work if you have to update all workstations on a network manually. Fortunately, there are several possibilities to do this job automatically.

Using the TbLoad Utility

The TbLoad utility that ships with TBAV for Windows is used to automatically update the existing ThunderBYTE Anti-Virus software

installed on your system. Please refer to the section about TbLoad in the TBAV for Windows documentation.

Using the DOS REPLACE Command

Maintain a directory \TBAV_UPD\ on a public server drive and place any new version of the TBAV utilities or any new signature file (TBSCAN.SIG) in this directory.

The workstations should execute a batch file automatically after users login on the network. This batch file should contain the following lines:

```
REM UPDATE TBAV IF A NEW RELEASE IS AVAILABLE.
REPLACE X:\TBAV_UPD\*.* C:\TBAV /U /R
REPLACE X:\TBAV_UPD\*.* C:\TBAV /A /R
```

REPLACE is a standard DOS utility. If the /U option is specified, it copies the files specified by the first parameter ONLY if they are newer than the files specified in the second parameter. The /A option makes sure that REPLACE copies files that do not yet exist in the destination directory (specified by the second parameter). Make sure REPLACE is in the current path, and that the specified paths are valid for your configuration. The x in the above example represents the drive letter of the public server drive.

Using this technique, you only have to update one drive with the new signature file or anti-virus software; all workstations will then update themselves when users login! You can also add the /S option if you want REPLACE to scan all directories on the workstations drives for matching files. Please consult your DOS Operating System manual for more details.

WARNING:
Don't forget to execute TbSetup on the new utilities in the X:\TBAV_UPD directory, thus ensuring that the REPLACE command also copies the new ANTI-VIR.DAT file.

1.4.5 Using the PKUNZIP Utility

Maintain a directory \TBAV_UPD\ on a public server drive and place any new version of the TBAV utilities or any new signature file (TBSCAN.SIG) in this directory.

The workstations should execute a batch file automatically after users login on the network. This batch file should contain the following lines:

```
REM UPDATE TBAV IF A NEW RELEASE IS AVAILABLE.
PKUNZIP -N -O X:\TBAV_UPD\TBAV???.ZIP C:\TBAV
```

Make sure the file PKUNZIP.EXE is in the current path, and that the paths specified are valid for your configuration.

Following this procedure, the PKUNZIP command comes into action only when you just updated the ZIP files in the X:\TBAV_UPD directory. Now you only have to update one drive with the new anti-virus software, and all workstations update themselves when users login.

WARNING:

If you did not create a Recovery Diskette during installation, we recommended that you do so. See the "Create a Recovery Diskette" section in Chapter 2 for instructions on how to do this. The example setups assume you have created such a recovery diskette.

2 Defining Your Anti-Virus Strategy

In this chapter, you learn how to accomplish two things: how to protect yourself against virus infection, and how to recover from virus infection. We recommend you read this chapter because it contains several useful tips.

2.1 Protecting Yourself Against Virus Infection

Maintaining a reliable safety system implies that you actively take measures to protect your system from virus infection, since some viruses can hide themselves perfectly once resident in memory.

TIP:

At least once a week you should boot from a clean and write-protected diskette and execute TbScan to check your computer for virus infections.

The tightness of your safety system really depends on two things:

1. The vitality of the appropriate computer system

2. The amount of time you want to invest to let the safety measures take place

For example, on a standalone computer containing low risk data, and in an environment with little exchange of computer software, a daily scan is usually sufficient. For company use, however, in a network environment where users exchange diskettes frequently, where disks contain highly vulnerable information, and where a network going down means the loss of an extensive amount of money, protection must be as tight as the organization can practically handle.

With this in mind, it's impossible to define one strategy for system protection that will work for everybody. It all depends on your demands and possibilities.

The TBAV utilities, however, are extremely flexible and enable you to define your own strategy, one that will work for your special needs. Although the following six basic precautions are NOT intended to be a complete protection system, they do provide a foundation on which you can build your own strategy.

1. Install TBAV on your hard disk

You can customize the installation to suit your specific needs. Be sure to use TbSetup to maintain recovery information of all executable files of your system! Refer to the Installing the TBAV Utilities section in Chapter 1 for details.

The following examples assume that all utilities reside in the default directory \TBAV. All example setups require that TbSetup is running. If your system has more hard disks or disk partitions, you should repeat the TbSetup invocation for every drive or partition.

TIP

Remember that you can use the ALLDRIVES and ALLNET options to make TbSetup process all local respectively remote non-removable drives.

Furthermore, the example setups assume you have created a recovery diskette.

2. Prepare a recovery diskette

It is imperative to have a clean recovery diskette to recover from virus infection. If you didn't create a recovery diskette during the TBAV installation, take a few minutes to prepare one now. Later, when a virus infects your system, it's too late! To create a recovery diskette, follow these steps:

1. Insert a new diskette in drive A:, and then change to the DOS directory by typing CD \DOS and pressing ENTER.

2. Type FORMAT A: /S, and press ENTER. The /S switch copies the DOS system files to the disk so you can boot the computer with it.

3. Type COPY SYS.COM A: and press ENTER. This copies the SYS.COM program, which is the program that DOS uses to copy its system files to a disk.

4. Type CD \TBAV to return to the TBAV directory.

5. Type MAKERESC A: and press ENTER to create a recovery disk in drive A.

WARNING:

If your computer has two floppy disk drives, be sure you know which one is drive A and create your recovery disk there. A PC never tries to boot from drive B.

The MAKERESC.BAT procedure creates a reliable recovery diskette by creating or copying the following:

- A backup of the boot sector, partition sector and CMOS configuration.

- A CONFIG.SYS file, containing:

```
FILES=20
BUFFERS=20
DEVICE=TBDRIVER.EXE
DEVICE=TBCHECK.EXE FULLCRC
```

- An AUTOEXEC.BAT file, containing:

```
@ECHO OFF
ECHO OFF
PATH=A:\
TBAV
CLS
ECHO WARNING!!!
ECHO IF YOU SUSPECT A VIRUS, DO NOT EXECUTE
     ANYTHING FROM THE HARD DISK!
```

- The following files:

```
TBAV.EXE
TBAV.LNG
TBSCAN.EXE
TBSCAN.LNG
TBSCAN.SIG
TBDRIVER.EXE
TBDRIVER.LNG
TBCHECK.EXE
TBCLEAN.EXE
TBUTIL.EXE
TBUTIL.LNG
```

6. Copy to the diskette any other utilities that could come in handy in an emergency, such as a simple editor to edit CONFIG.SYS and AUTOEXEC.BAT files. If your hard disk needs special device drivers to unlock added features, such as DoubleSpace or Stacker, copy the appropriate drivers to the recovery diskette and install them in the CONFIG.SYS file on drive A:, being careful to avoid statements that access the

hard disk. Be sure to check the instructions in the device driver's manual for the correct procedures.

CAUTION:
If you are using the text editor that ships with DOS 5.0 or later, be sure to not only copy the file EDIT.COM to drive A:, but also QBASIC.EXE, which EDIT.COM uses.

7. Make sure you write protect your recovery disk. Now label it "Recovery Disk" and include on the label the identification of the PC to which the diskette belongs. Store the diskette in a safe place.

TIP:
For additional security, make another recovery diskette and store it in a separate location.

3. Prevent the Installation of Unauthorized Software

Many companies do not allow employees to install or execute unauthorized software. Similarly, perhaps you want to keep family members from invading your computer with haphazard games and sundry software. TBAV provides a watchdog function that can help to enforce this. Follow these steps:

1. First you need to add the following lines to the CONFIG.SYS file:

```
DEVICE=C:\TBAV\TBDRIVER.EXE
DEVICE=C:\TBAV\TBCHECK.EXE SECURE
```

Alternately, if you are using the TBSTART.BAT file, then you would add the following two lines to it:

```
C:\TBAV\TBDRIVER
C:\TBAV\TBCHECK SECURE
```

2. Run TbSetup on the system by typing "TBSETUP ALLDRIVES" and pressing ENTER.

3. Reboot the system.

From now on, TbCheck puts an effective clamp on any user who tries to execute software that TbSetup has not duly authorized first. Whenever someone is trying to execute an unknown program, TBAV displays the following message:

```
+-----------TBAV Interception-----------+
| The requested program (GAME.EXE)      |
| is not authorized and can not be      |
| executed.                             |
| Execution cancelled! Press any key... |
+---------------------------------------+
```

4. Restrict User Access

Most of the TBAV utilities are interactive; that is, they communicate with a knowledgeable user to establish appropriate action in ambiguous situations. Many companies, however, insist that the system operator be the sole authority allowed to communicate with TBAV, and so avoid wrong doing by possibly inept employees.

It is for this very reason that most of the TBAV utilities support the SECURE option. When you specify this option, TBAV suspends all user interaction with the utilities. In other words, TBAV never asks users for permission to allow questionable operations, avoiding erroneous decisions that might well result in irreparable havoc. This option also prevents the user from disabling or unloading the TBAV utilities.

5. Never Use "Strange" Diskettes to Boot

Boot only from your hard disk or from your original DOS diskette. NEVER use someone else's disk to boot the computer. If you have a hard disk, make certain that the door to your floppy drive is open before resetting or booting the machine.

6. Run the DOS CHKDSK Command Often Use the DOS program CHKDSK frequently (without the /F switch). CHKDSK can sometimes indicate the presence of a virus simply because some viruses change the disk structure incorrectly, thereby causing disk errors in the process. Look out for changes in the behavior of your software or your PC. Any change in their behavior is suspect, unless you know its cause. Some highly suspicious symptoms are:

A decrease in the amount of available memory space. CHKDSK should report 655,360 total bytes of memory. Programs require more time to execute.

Programs do not operate as they used to, or they cause the system to crash or reboot after some time.

Data mysteriously disappears or becomes damaged.

The size of one or more programs has increased.

The screen behaves strangely or displays unusual information.

CHKDSK detects many errors.

TIP:

You can also instruct TbScan to mimic the behavior of the DOS command CHKDSK. Simply execute TbScan with the fatcheck option enabled. For example, if you want TbScan to scan your C: and D: drive once a day, and to check the integrity of those disks, place the following command in your AUTOEXEC.BAT file:

```
TbScan  C:\  D:\ FATCHECK ONCE
```

2.2 Recovering from Virus Infection

This section presents some tips on how to clean your computer system when it is has been compromised by a virus.

1. Backup Your Data

The very first thing to do when you realize that your system might be infected is to back up all important files immediately. Label the new backup as unreliable, since some of the files might be infected.

CAUTION:

Use fresh backup media and do not overwrite a previous backup set. You might need the previous set to replace lost or contaminated files.

2. Boot From a Recovery Diskette

When you become aware of a virus infection, it is imperative that you boot only from a reliable, write protected recovery system diskette.

3. Know the Symptoms of a Virus

Now execute TbScan for an indication of what is wrong, or boot from a recovery diskette and compare its system files with those on the hard disk to check for changes. During this test you should take care to stay logged onto your system diskette.

TbScan reports the virus name if it knows the virus, or it gives a summary of file changes if it can't identify the virus. If you use the command line below, for example, TbScan processes all non-removable drives and prints the results of the scan process to the printer.

```
TBSCAN  ALLDRIVES  LOGNAME=LPT1  LOG
```

Also run TbUtil, to check the boot sector, partition code and the CMOS configuration, using the following command:

```
   TBUTIL  COMPARE
```

WARNING:

To prevent a virus from invading the system's memory and possibly masking the test results, do not execute any program on your hard disk. TbCheck warns you if you accidentally try to execute an infected or unauthorized program on your hard disk.

Remember that it is in the nature of a file virus to infect as many programs as possible over a short period. You ll seldom find only a few programs on a hard disk to be infected. A TbScan virus alert that flags a mere one percent of the files on a hard-worked system is probably just a false alarm that has nothing to do with a real virus.

In other words, if the file compare test indicates that all of them are still the same, you know at least that you are not dealing with a file virus. Avoid using the same copy of the TbScan program on another system after discovering a virus. Like any other program file, TBSCAN.EXE itself can become infected!

To check infections of the TbScan program, the program performs a sanity check when it runs. Unfortunately, there is no way to make software 100% virus-proof. A sanity check does not work if a stealth-type virus is at work. A stealth virus can hide itself completely when you run a self-check.

In case you are wondering, this is not a bug in TbScan. The failure to detect stealth viruses is common to all software that performs a sanity check. We, therefore, recommend that you keep a clean version of TbScan on a write-protected diskette. Use this diskette to check other machines once you have found a virus on your system.

4. Identify Virus Characteristics

Viruses come in many different guises and have their own peculiarities. It is extremely important to know at the earliest possible stage which particular kind of virus you are dealing with. This gives you at least some indication of the nature and the amount of the damage it might have caused already.

Some viruses infect only executable files that you can easily reinstall or replace from a clean source. Others swap some random bytes anywhere on the hard disk, which could affect data files as well, although the results might not be noticeable for some time. Then there are those viruses that damage the hard disk partition table or file allocation table. Some of the even nastier viruses, the so-called multipartite viruses, operate in more than one area.

Once you isolate the virus, either contact your support BBS, consult literature on virus problems, or get in touch with a virus expert.

WARNING:

Whatever you do, DON'T PANIC! An inexperienced user, reacting in confusion, can often create more havoc than the virus itself, such as blindly eradicating important data. While an instant reformat might get rid of the virus, it will definitely destroy all your recent work as well.

5. Restore the System

Again, while recovering from a virus infection, it is particularly important to boot only from a clean write-protected system diskette. This the only way to keep a virus out of the system's memory. Never execute a program from the hard disk.

Using the SYS command on the system or recovery diskette, restore the master boot sector and the DOS system files to the hard disk. If the boot sector or partition code contains a virus, you can also use the following command to get rid of it by restoring clean sectors:

```
TBUTIL  RESTORE
```

WARNING:

Many modern hard disks, notably IDE or AT drives using advanced pre-formatting methods, are low-level formatted by the manufacturer, ready for partitioning and a DOS format. NEVER try to low-level format these drives yourself. Doing so can ruin the drive. It is always better to back up the partition table with a utility such as TbUtil, which restores the partition table for you without reformatting.

If TBAV identifies the virus as a file virus, the safest course is to remove the infected files (using TbDel) and to copy or reinstall all executables from a CLEAN source. A virus cleaning utility, such as TbClean, won't always be able to fully restore the original program code, so use this only as a last resort, such as when you don't have a reliable backup. It might be necessary to replace data files as well if the virus is known to cause damage in that area.

CAUTION:

After reassuring yourself that the system is absolutely clean again, run a careful check on all diskettes and backups to remove every single trace of the virus. Keep in mind that it takes only one infected diskette to reacquire the problem.

3 Using the TBAV utilities

This chapter fully describes each of the TBAV utilities. For quick reference, we will present each utility using at least three sections: Understanding the utility, Working with the utility, and Maximizing the utility. Most discussions also include a fourth section: Understanding the utility's operating process.

3.1 Using TbSetup

Even though TbSetup does not take an active part in actual virus detection or cleaning, it is nonetheless an indispensable tool in adding support to the rest of the ThunderBYTE Anti-Virus utilities. TbSetup organizes control and recovery information, thereby giving extra power to the other utilities. It gathers information, mainly from program files, into a single ANTI-VIR.DAT reference file, one in each directory.

NOTE:
See the "Understanding ANTI-VIR.DAT Files" section at the end of this chapter for a fuller explanation of these files.

3.1.1 Understanding TbSetup

Although the ThunderBYTE utilities can work perfectly well without the ANTI-VIR.DAT files, we recommend that you have TbSetup generate these files. TBAV uses these files for several purposes:

TbScan and the memory resident TbCheck program perform an integrity check while scanning if it can detect the ANTI-VIR.DAT file. If a file becomes infected by a virus, the information in the ANTI-VIR.DAT file will not match the actual file contents, and TbScan and TbCheck will inform you that the file has been changed.

The TbSetup program recognizes some files that need special treatment. An example of such a file is a disk image file of a network remote boot disk. You should completely scan such a file, which actually represents a complete disk. TbSetup puts a mark in the ANTI-VIR.DAT file to ensure that TbScan scans the entire file for all viruses.

Once a file becomes infected, TbClean can reconstruct the original file. The information in the ANTI-VIR.DAT file will be of great help

to TbClean. TbClean can cure some infected programs only if there is information about the program in the ANTI-VIR.DAT file.

TbCheck (a tiny resident integrity checker) has no purpose if there are no ANTI-VIR.DAT files on your system. The resident TBAV utilities need the ANTI-VIR.DAT files to maintain permission information. Without ANTI-VIR.DAT files you can't prevent false alarms other than by disabling a complete feature.

NOTE:
Be aware that the ANTI-VIR.DAT directory entries have by default the attribute hidden and therefore do not show up when you use standard directory commands. You can see the filenames only with the help of special utilities or with the DOS 6 command DIR AH.

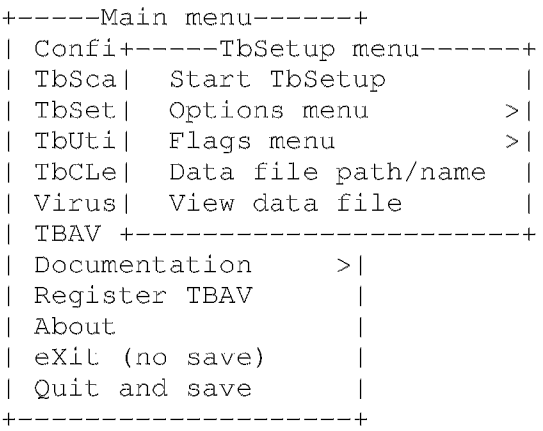
3.1.2 Working with the TbSetup Menu

This is the one program where the rule applies: The less you use the program, the better your protection against viruses! Why? Keep in mind that an ANTI-VIR.DAT file stores vital information needed to detect a virus, as well as data for subsequent recovery and for cleaning. Consider, then, what would happen if you were to execute TbSetup after a virus entered the system. The information in the ANTI-VIR.DAT file would be updated to the state of the infected file, wiping out all traces of data needed to reconstruct the original file to its uninfected state.

WARNING:
NEVER, NEVER, NEVER, use TbSetup when there is the slightest evidence of a virus on your system. Once TbSetup generates ANTI-VIR.DAT files as part of the initial setup, you should confine any subsequent use of TbSetup to directories with new or changed program files.

Now we will explore these menu options.

Selecting the "TbSetup" option from the Main Menu displays the following menu:

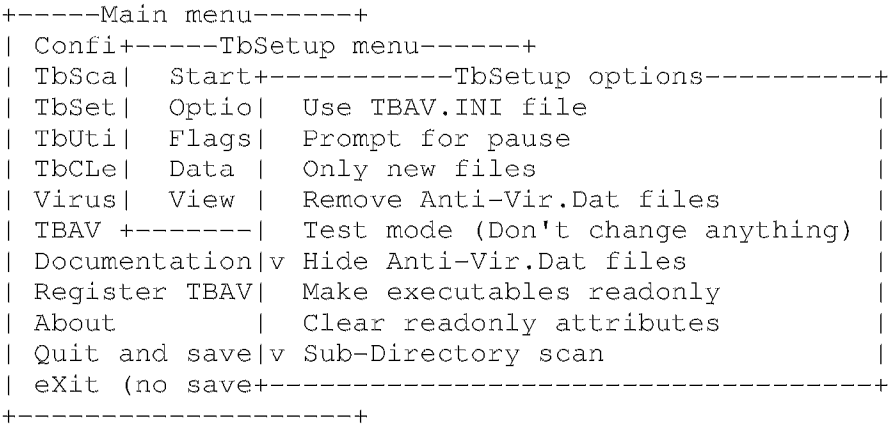


The "Start TbSetup" Option

Select this option only after you complete your selection of other options on this menu and other sub-menus. When you select this option, the "Enter disk / path / file(s) to process:" window appears. Type in the drive and directory you want to setup and press ENTER.

The "Options Menu" Option

Selecting this option displays the following menu:



Use TBAV.INI file.

By enabling this option, the TbSetup configuration values, saved in the TBAV.INI file, will also apply when loading TbSetup from the command line.

CAUTION:
If you specify options in the TBAV.INI file, you cannot undo them on the command line.

Prompt for pause.

When you specify this option, TbSetup stops after it processes the contents of one window. This enables you to examine the results.

Only new files.

Use this option if you want to add new files to the ANTI-VIR.DAT database but prevent the information of changed files from being updated. Updating the information of changed files is dangerous because if the files are infected, the information to detect and cure the virus is overwritten. This option prevents the information from being overwritten but still allows adding information of new files to the database.

Remove ANTI-VIR.DAT files.

If you want to stop using the ThunderBYTE utilities you do not have to remove all the ANTI-VIR.DAT files yourself. By using this option TbSetup neatly removes all ANTI-VIR.DAT files from your system.

Test mode (Don't change anything).

Use this option if you want to see the effects of an option without the risk of activating something you don't want to activate. This option instructs the program to behave as it normally would but not change or update anything on your hard disk.

Hide ANTI-VIR.DAT files.

The ANTI-VIR.DAT files are normally not visible in a directory listing. If you prefer them to be visible, disable this option.

NOTE:
Be aware that this option applies only for new ANTI-VIR.DAT files

Make executables read-only.

Since TbFile permanently guards the read-only attribute, we recommend that you make all executable files read-only to prevent any modifications on these files. TbSetup automatically does this job for you if you enable this option. TbSetup recognizes files that you should not make read-only.

Clear read-only attributes.

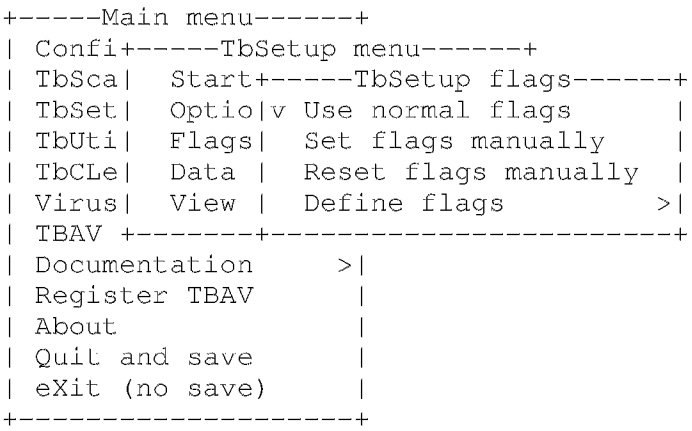
Use this option to reverse the "Make executables read-only" operation. If you enable this option, TBAV clears all read-only attributes on all executable files.

Sub-Directory scan.

By default, TbSetup searches sub-directories for executable files, unless you specify a filename (wildcards allowed). If you disable this option, TbSetup will not process sub-directories.

The "Flags Menu" Option

Selecting this option displays the following menu:



NOTE:
"Flags" refer to internal indicators, created by ThunderBYTE to signal internal file attributes.

This menu contains the following options:

Use normal flags.

This is the default setting for TbSetup.

Set flags manually.

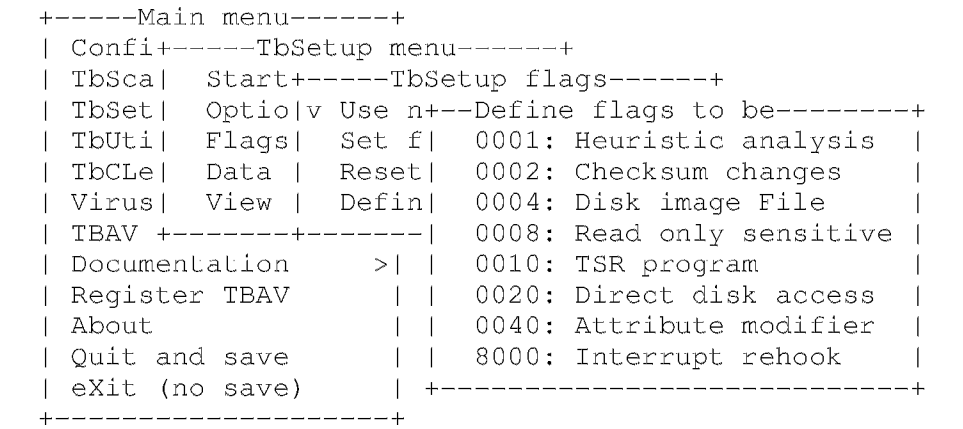
This option is for advanced users only. Using this option, you can manually set permission flags in the ANTI-VIR.DAT record. This option requires a hexadecimal bit mask for the flags to set; you can specify this bit mask by selecting one of more of the items listed in the "Define flags" sub-menu, which appears below.

Reset flags manually.

This option is for advanced users only. Using this option, you can manually reset permission flags or prevent flags from being set in the ANTI-VIR.DAT record. This option requires a hexadecimal bit mask for the flags to reset; you can specify this bit mask by selecting one or more of the items listed in the "Define flags" sub-menu, which appears below.

Define flags.

Selecting this option displays the changed following menu:



Selecting one or more of these options accomplishes the following:

0001: Heuristic analysis.

Programs with the 0001 flag will not be heuristically scanned.

0002: Checksum changes.

Programs with the 0002 flag will not be checked for file changes.

0004: Disk image File.

Files with this flag contain a disk layout and are checked completely.

0008: Read only sensitive.

Files with this flag cannot be changed to read-only.

0010: TSR program.

Programs with this flag have permission to stay resident in memory.

0020: Direct disk access.

Programs with this flag have permission to write directly to the disk.

0040: Attribute modifier.

Programs with this flag have permission to change program attributes.

8000: Interrupt rehook.

After a program with this flag starts, TbDriver should rehook interrupts.

The "Data File Path Name" Option

TbSetup searches for "special" files by using a file named TBSETUP.DAT. You can use this option to specify another path or filename that contains a list of special files. Select the option, and then enter the name (and path if necessary) of the data file you want to use.

The "View Data File" Option

Selecting this option displays the TBSETUP.DAT file on the screen for your viewing. Use the cursor movement keys to move through the file.

TIP:

Instead of using the internal file viewer to view the User Manual, you can substitute your own favorite viewer. See the "Configuring TBAV" section in Chapter 1 for details..

3.1.3 Maximizing TbSetup

Now that you know how to use TbScan's menus, you can more easily understand how to maximize its performance by using command line options. The following table summarizes these options:

option parameter	short	explanation
help	he	help
pause	pa	enable "Pause" prompt
mono	mo	force monochrome output
nosub	ns	skip sub-directories
newonly	no	do not update changed records
alldrives	ad	process all local fixed drives
allnet	an	process all network drives
remove	rm	remove ANTI-VIR.DAT files
test	te	do not create / change anything
nohidden	nh	do not make ANTI-VIR.DAT files hidden
readonly	ro	set read-only attribute on executables
nordonly	nr	remove / do not set read-only attribute
set=<flags>	se	set flags
reset=<flags>	re	reset flags / do not set flags
datfile=<filename>	df	specify the data file to be used

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

help (he).

Specifying this option displays a short list of available options, as listed above.

pause (pa).

Specifying this option stops after processing the contents of one window. This enables you to examine the results.

mono (mo).

This option enhances the screen output on some LCD screens or color-emulating monochrome systems.

nosub (ns).

By default, TbSetup searches sub-directories for executable files, unless you specify a filename (wildcards allowed). If you specify this option, TbSetup will not process sub-directories.

newonly (no).

Use this option if you want to add new files to the ANTI-VIR.DAT database but prevent the information of changed files from being updated. Updating the information of changed files is dangerous because if the files become infected, the information to detect and cure the virus is overwritten. This option prevents the information from being overwritten but still allows adding information of new files to the database.

alldrives (ad).

If you want TbSetup to process all local non-removable drives you can specify this option. Except for the initial execution, it isn't a good idea to use this option.

allnet (an).

Specify this option if you want TbSetup to process all network drives.

WARNING:
Except for the initial execution of the TBAV utilities, it isn't a good idea to use the "allnet" option

remove (rm).

If you want to stop using the ThunderBYTE utilities, you do not have to remove all the ANTI-VIR.DAT files manually. By using this option, TbSetup neatly removes all ANTI-VIR.DAT files from your system.

test (te).

Use this option if you want to see the effects of an option without the risk of activating something you don't want to activate. If you specify this option, the program behaves as it would normally but does not change or update anything on your hard disk.

nohidden (nh).

The ANTI-VIR.DAT files are normally not visible in a directory listing. If you prefer the ANTI-VIR.DAT files to be visible, use this option.

NOTE:
Be aware that the "nohidden" option applies only for new ANTI-VIR.DAT files

readonly (ro).

Since, TbFile permanently guards the read-only attribute, we recommend that you make all executable files read-only to prevent any modifications on these files. TbSetup automatically does this job for you if you use this option. TbSetup recognizes files that you should not make read-only.

nordonly (nr).

This option reverses the operation of READONLY option. If you use this option, TbSetup clears the read-only attribute from all executable files.

set (se).

This option is for advanced users only. Using this option you can manually set permission flags in the ANTI-VIR.DAT record. This option requires a hexadecimal bit mask for the flags to set. For information about the bit mask consult the TBSETUP.DAT file. Option format: Set =<flags>; for example: Set = 0001.

reset (re).

This option is for advanced users only. With this option you can manually reset permission flags or prevent flags from being set in the ANTI-VIR.DAT record. This option requires a hexadecimal bit mask for the flags to reset. For information about the bit mask consult the TBSETUP.DAT file. Option format: Reset =<flags>; for example: Reset = 0001.

datfile (df).

After the datfile option you can specify the name of the data file to use.

For the initial installation of TBAV, you could use the following command:

```
TBSETUP ALLDRIVES
```

Using the following command, you could specify which drives (C: and D:, for example) you want TbSetup to process:

```
TBSETUP C:\ D:\
```

Since you did not specify a filename in the above command, TbSetup assumes that the specified path to be the top-level path. In other words, TbSetup processes all its sub-directories. If you do specify a filename, TbSetup processes only that path, not any subdirectories. You can use wildcards (the asterisk [*] or the question mark [?]) in the filename.

You can use the NEWONLY option to prevent TbSetup from overwriting existing information. To help you remember that you need to run TbSetup again, the next time you run TbScan it displays either a small 'c' after the file to indicate a new file or a capital 'C' if a file has simply been changed.

If you add a new file called TEST.EXE to your directory C:\TESTING, you should execute the following command:

```
TBSETUP C:\TESTING\TEST.EXE
```

If you install a new product in a new directory, C:\NEW, you should use the following command:

```
TBSETUP C:\NEW
```

3.1.4 Understanding TbSetup's Operation

TbSetup divides the screen into three windows: an information window displaying data file comments across the top of the screen, a scanning window on the left, and a status window on the right.

The lower left window lists the names of the files being processed, along with file specific information in the following way:

```
TEST.EXE 01234  12AB23CD   Added    * 0001
|        |      |          |        | |
|        |      |          |        | |
|        |      |          |        | 'flags' set for this file
|        |      |          |        indicates 'special' file
|        |      |          action performed
|        |      32-bit CRC (checksum)
|        file size in hexadecimal number
name of file in process
```

Do not be concerned if the information flies too fast for you to read, or if it puzzles you. These details are provided purely for diagnostic use.

The scanning window also displays an action performed field, which indicates whether an entry in the ANTI-VIR.DAT was added, changed or updated:

Added.

Means that there was no previous entry for this file in the ANTI-VIR.DAT record and that a new entry was added.

Changed.

Means that there was an existing entry but the file has been changed and ANTI-VIR.DAT information was updated.

Updated.

Means that there was an ANTI-VIR.DAT record and the file was found to be unchanged. TbSetup did, however, change some of the program's permission flags, due to either an entry in the TBSETUP.DAT file or in compliance with a SET or RESET option.

TIP:
You can abort TbSetup at any time by pressing Ctrl+Break.

3.1.5 Understanding TBSETUP.DAT Files

Although the ThunderBYTE utilities perform well on almost every file without extra help, there are some files that need special attention. TbSetup uses information collected in the TBSETUP.DAT data file, to flag

these special files in the ANTI-VIR.DAT file. The other ThunderBYTE utilities then use this information to determine how they should treat such a "special" file.

Some programs maintain configuration information inside the executable file (EXE, COM) itself. Whenever you change the configuration of these programs, the executable file changes as well, along with its checksum. As a result, the new checksum no longer matches the one stored in the TBSETUP.DAT file. Since some TBAV utilities use this checksum information to verify integrity or cleanup results, they need to know when a file's checksum is allowed to change. TbScan can use generic detection methods such as "heuristic" analysis to detect unknown viruses. Since heuristic analysis implies inevitable false alarms when a file looks like a virus, TbScan might have to decide not to do a heuristic analysis on such a program.

Some of the TBAV utilities guard the read-only attribute and ensure that it can be removed only with the user's explicit permission. A few programs, however, refuse to run properly with the read-only attribute set.

TbScan's default scanning method performs perfectly well with just about any file, but there are some that need special analysis. Such a file is the Novell NET$DOS.SYS file, which is not a device driver as the filename extension suggests, but a disk image of the bootable disk. You should, therefore, scan it completely for all signatures, including COM and BOOT. The resident monitoring utilities of the TBAV package detect all sorts of virus-specific behavior. Some programs, even though they might act like a virus, are still perfectly normal and should be permitted to execute without TBAV interference.

You need not worry if you discover that a few files will be excluded from heuristic analysis. TBAV still scans these files in the conventional way for signatures. Furthermore, TBAV will not grant heuristic exclusion unless a file exactly matches its entry in the TBSETUP.DAT file, including its name, size, and 32-bit CRC checksum.

This safety feature eliminates security holes effectively, since if a listed file is already infected, its checksum won't match the 32-bit CRC in the TBSETUP.DAT file and the exclusion does not apply. By the same token, if a program becomes infected at a later date, the result is a change in at least one of its characteristics, so the record in the ANTI-VIR.DAT file no longer matches and the file will be subject to full heuristic analysis like any other.

3.2 Using TbScan

TbScan is the program you will most likely use the most detect virus infections.

3.2.1 Understanding TbScan

TbScan is a scanner which has been specifically designed to detect viruses, Trojan Horses and other such threats to your valuable data. Most viruses consist of a unique sequence of instructions, called a signature. By checking for the appearance of such signatures in a file we can find out whether a program has been infected. Scanning all program files for the signatures of all known viruses helps you to find out quickly whether your system has been infected and, if so, by which virus.

Understanding TbScan involves understanding three main features of the program.

Fast Scanning

TbScan is the fastest scanner on the market today. It, therefore, invites you to use it from within your AUTOEXEC.BAT file every morning. Thanks to its design, TbScan does not slow down if the number of signatures increases. It doesn't matter whether you scan a file for 10 or a 1000 signatures.

TbScan even checks itself upon launching. If it detects infection, it aborts and displays an error. This minimizes the risk of the TbScan program itself transferring a virus to your system.

Heuristic Scanning

TbScan can detect unknown viruses. The built-in disassembler is able to detect suspicious instruction sequences and abnormal program layouts. This feature is called "heuristic scanning" and is partially enabled by default. TBAV performs heuristic scanning on files and boot sectors.

NOTE.

Virus scanners can only tell you whether your system has been infected. By that time only a non-infected backup or a recovery program such as TbClean can properly counter a virus infection.

Scan Scheduling

Every PC owner should use a virus scanner frequently. It is the least one should do to avoid damage caused by a virus. We recommend that you devise your own schedule for a regular scan of your system. See Chapter 2 for details.

We recommend the following scan sessions, to be used in combination with each other:

Execute TbScan from write-protected bootable diskette once a week. Boot from this diskette before invoking the scanner. Booting from a clean diskette is the only way to make sure that no stealth virus can become resident in memory.

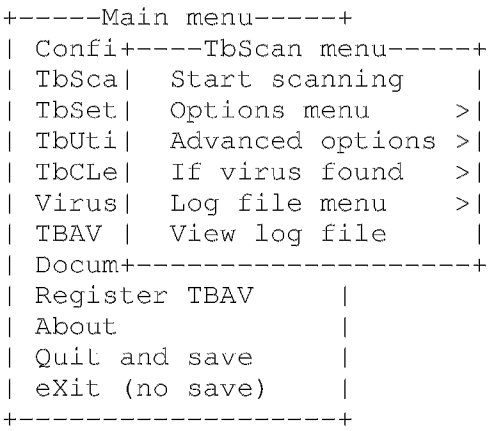
Invoke a daily scan. You can invoke TbScan with the ONCE option from within the AUTOEXEC.BAT file to perform the daily scan session automatically, which is the default if you used the standard installation procedure for TBAV (see Chapter 1). It is not necessary to boot from the bootable TbScan diskette to perform the daily scan.

Scan each new diskette. You should scan EVERY diskette you receive from a friend or acquaintance for viruses to ensure that a virus hasn't been included along with a copy of "a great game!"

3.2.2 Working with the TbScan Menus

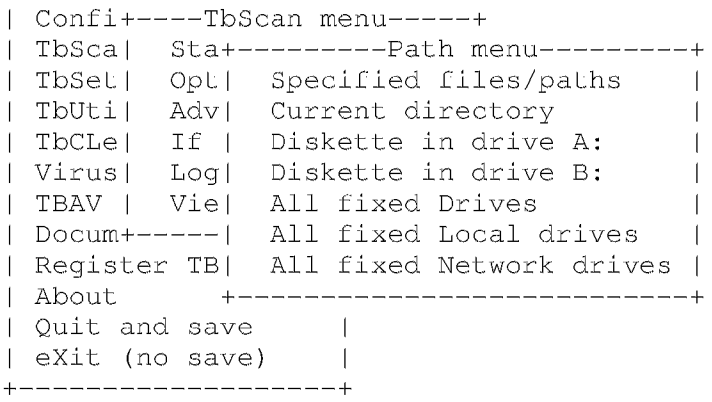
For daily use you can activate TbScan by loading the program from the DOS command line (e.g., in the AUTOEXEC.BAT file), or through the TBAV menu. For weekly use, when scanning from the TbScan diskette, you could use the DOS command. The Maximizing TbScan section of this chapter lists the TbScan DOS options. This section describes the use of the TbScan Menu, which is part of the TBAV menu. Taking each menu item in order, we ll explore the function of each.

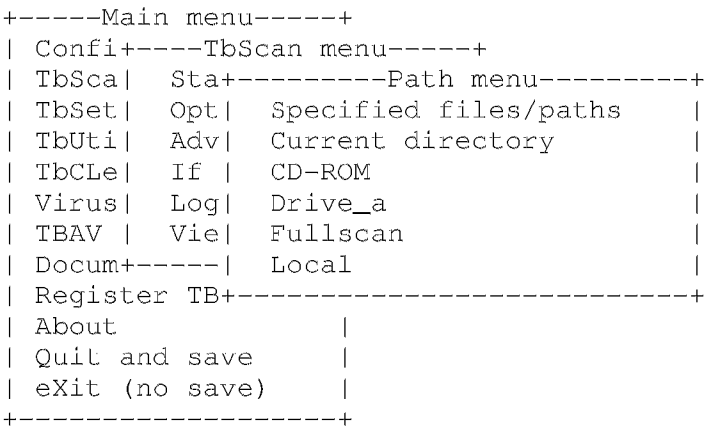
Selecting the "TbScan" option from the TBAV menu displays the following menu:



The "Start Scanning" Option

Selecting the "Start Scanning" option from the TbScan Menu displays one of the following "Path Menu" configurations:





The first menu configuration includes scan targets such as CD-ROM, Drive_a, etc. Primarily, TBAV for Windows uses these scan targets, but TbScan for DOS can also use them. If the TBAV menu finds one or more of these scan targets (the targets are really files with the filename extension SCN), the Path Menu will then display the list of available targets. If no such scan targets exist, the second Path Menu configuration will appear.

NOTE:
Please be aware that the actual menu items you come across in the Path menu might differ slightly, depending on your system configuration.

The Path Menus list the following options:

Specified files/paths.

This option always presents you with a small prompt window in which you can specify the drives, paths, or even files you want to scan. You can specify multiple path specifications by separating each with spaces. This specification automatically initializes with the last path you scanned before you saved the configuration.

Current directory.

Select this option if you want to scan only the directory from which you started the TBAV menu shell.

Diskette in drive A: or Diskette in drive B:.

If you want to scan multiple diskettes, you might wish to activate the Repeat option of TbScan. See the TbScan Options Menu for more information.

All fixed drives.

This option instructs TbScan to scan all available drives (except the removable ones) completely. Depending on the settings in the TBAV configuration menu, TbScan prompts you to confirm the selected drives.

All fixed Local drives.

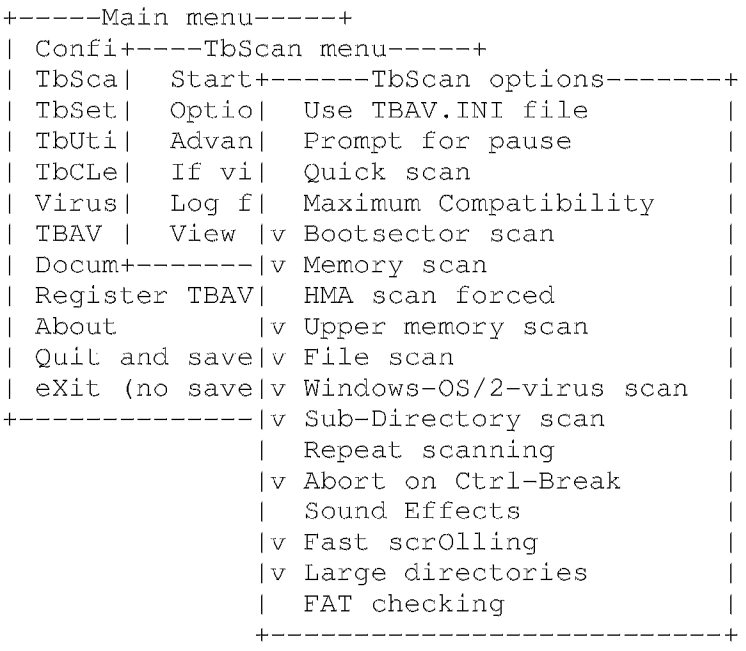
If you are on a network, you probably don't want to scan the entire network. Using this option you can scan just the drives that reside in your machine. Depending on the settings in the TBAV configuration menu, TbScan prompts you to confirm the selected drives.

All fixed Network drives.

Using this option you can scan all network drives. Depending on the settings in the TBAV configuration menu, TbScan prompts you to confirm the selected drives.

The "Options Menu" Option

Selecting the "Options Menu" option displays the following menu:



Taking each menu item in order, we ll explore the function of each.

Use TBAV.INI file.

TbScan searches for a file named TBAV.INI in the TBAV directory. By enabling this option, the TbScan configuration values, saved in the TBAV.INI file, will also be valid when loading TbScan from the command line.

CAUTION:
Be aware that if you specify options in the TBAV.INI, you cannot undo them when running TbScan from the command line.

Prompt for pause.

When you activate this option, TbScan stops after it checks the contents of each window. As each window fills with files, a "[More]" prompt appears at the bottom of the screen. Simply press any key to view the next list of files. Using this feature enables you to examine the results of the scan without having to consult a log file afterwards.

Quick scan.

This option instructs TbScan to use the ANTI-VIR.DAT files to check for file changes since the last scan. TbScan scans only those files that have changed (CRC change) or are not yet listed in ANTI-VIR.DAT. The other files are just checked for matching ANTI-VIR.DAT records. By default, TbScan always scans files (the quick scan option is not enabled by default).

Maximum compatibility.

If you select this option, TbScan attempts to be more compatible with your system. Use this option if the program does not behave as you would expect or if it halts the system. Be aware, however, that this option slows down the scanning process. Therefore, use it only when necessary. Be aware also that this option does not affect the results of a scan.

Boot sector scan.

Enabling this option forces TbScan to scan the boot sector. A boot sector is a certain part of a disk, which is used by the operating system to initialize itself. A special class of viruses (boot sector viruses) use this special part of a disk to infect your system.

Memory scan.

Enabling this option forces TbScan to scan the memory of the PC.

HMA scan forced.

By default, TbScan automatically detects the presence of an XMS-driver and scans the HMA. If you are using an HMA-driver that is not compatible with the XMS standard, you can use this option to force TbScan to scan HMA.

Upper memory scan.

By default, TbScan identifies RAM beyond the DOS limit and scans that memory. This means that it scans video memory and the current EMS. You can use this option to enable the scanning of non-DOS memory.

File scan.

By default, TbScan checks files for viruses. Removing the check mark disables file scanning. This option is particularly useful if, for example, you have been struck by a boot sector virus. In order to scan only boot sectors of your floppy disks, you can disable file scan using this option.

Windows-OS/2-virus scan.

By default, TbScan scans Windows and OS/2 files for viruses. Removing the check mark disables Windows and OS/2 file scanning.

Subdirectory scan.

By default, TbScan searches sub-directories for executable files, unless you specify a filename (wildcards allowed). If you disable this option, TbScan does not scan sub-directories.

Repeat scanning.

This option is very useful if you want to check a large number of diskettes. TbScan does not return to DOS after checking a disk, rather it prompts you to insert another disk in the drive.

Abort on Ctrl-Break.

If you don't want to be able to abort the scanning process by pressing Ctrl+Break, you can disable this option.

Sound Effects.

Checking this option enables an audible sound when TbScan detects a virus.

Fast scrolling.

TbScan displays processed files in a scrolling window, which scrolls in one of two methods: fast scrolling, in which the files appear on

top of the previous ones if the window becomes full, and the conventional slow method of scrolling, in which the files at the bottom "push up" the previous ones. By default TbScan uses the faster but less attractive method of scrolling.

Large directories.

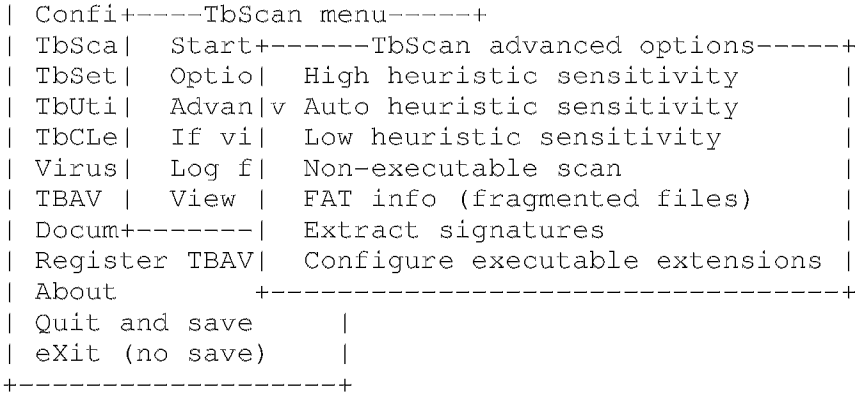
If TbScan's directory table runs out of space, which is very unlikely, you can use this option to allocate a large directory table.

Fat checking.

If this option is specified, and TbScan is able to use its internal file system, it will check the disks for lost clusters, cross linked clusters, invalid cluster numbers, and invalid allocation sizes. These errors often indicate system problems and need to be corrected as soon as possible. Because TbScan needs to read the FAT and all directories anyway, it can perform this important check without using additional time.

The "Advanced Options" Option

When you select the Advanced Options option, the following menu is displayed:



Let's now explore these options.

High heuristic sensitivity.

While TbScan always performs a heuristic scan on the files being rocessed, it reports a file as infected only if it is very probable that the file is infected. If you select this option, TbScan is somewhat more sensitive. In this mode, TbScan detects 90% of the new, unknown viruses without any signature. Be aware, however, that some false alarms might occur.

Auto heuristic sensitivity.

By default, TbScan automatically adjusts the heuristic detection level after it finds a virus. In other words, when TbScan finds a virus, it then goes on as if you had selected High heuristic sensitivity. This option provides you maximum detection capabilities in case you need it, while at the same time keeps false alarms at a minimum.

Low heuristic sensitivity.

In this mode TbScan almost never issues a false alarm. It still, however, detects about 50% of the new, unknown viruses.

Non-executable scan.

This option instructs TbScan to scan non-executable files (files with an extension other than COM, EXE, SYS, OV? or BIN) as well as executables. If TbScan finds out that such a file does not contain anything that the processor can execute, it skips the file. Otherwise TbScan searches the file for COM, EXE and SYS signatures. Be aware that TbScan does not perform heuristic analysis on non-executable files. Since viruses normally do not infect non-executable files, it is not necessary to scan non-executable files too. We recommend, in fact, that you NOT use this option unless you have a good reason to scan all files. Again, you must execute a virus before it can do what it was programmed to do, and since you do not execute non-executable files, a virus in such a file cannot do anything. For this reason viruses do not even try to infect such files. Some viruses, however, do write to non-executable files, but this is a result of "incorrect" programming. And even though these non-executable files contain corrupted data, they still won't harm other program or data files.

FAT info (fragmented files).

If this option is specified, TbScan displays the number of fragmented files after it has finished scanning. If the number of fragmented files is high, you can increase the system performance by using a disk optimizer. This option is only valid if the option 'fatcheck' has been specified, and TbScan is using its internal file system.

Extract signatures.

This option is available to registered users only. See the Using TbGenSig section in Chapter 4 for more information.

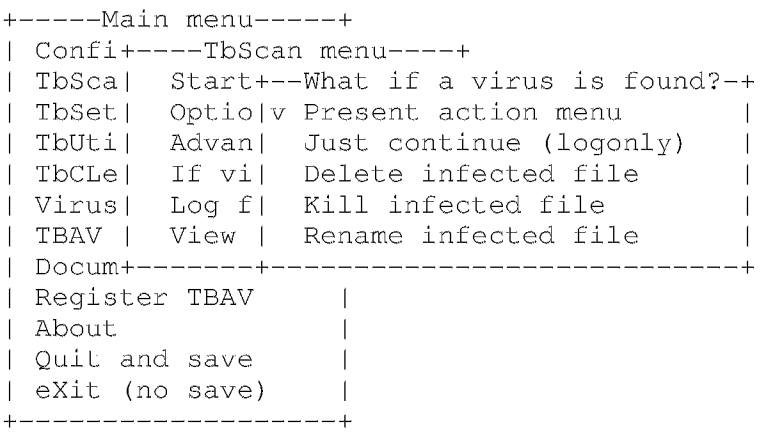
Configure executable extensions.

By default, TbScan scans only those files that have a filename extension that indicates that the file is a program file. Viruses that do not infect executable code simply do not exist. Files with the extension EXE, COM, BIN, SYS, and OV? (note the wildcard: the OV? specification includes files such as OVR and OVL) are considered executable. There are, however, some additional files that have an internal layout that makes them suitable for infection by viruses. Although it is not likely that you will ever execute most of these files, you might want to scan them anyway. Some filename extensions that might indicate an executable format include: .DLL (MS-Windows Dynamic Link Library), .SCR (MS-Windows screen saver file), .MOD (MS-Windows file), .CPL (MS-Windows Control Panel application), .00? and .APP. While infection of such files is not likely, you might want to scan them once in while. To force TbScan to scan these files by default, select this option and fill out the extensions you want TbScan to scan. For example, you can specify .DLL.SCR.CPL (with no spaces in between). You can also use the question mark wildcard.

WARNING:
Be careful which extensions you specify. Scanning a non-executable file, for example, causes unpredictable results, and might result in false alarms.

The "If Virus Found" Option

Selecting this option displays the following menu:



Let's explore these options.

Present action menu.

This option (the default) instructs TbScan to display a menu listing four possible actions if it detects a virus: just continue, delete, kill or rename the infected file.

Just continue (logonly).

By default, if TbScan detects an infected file, it prompts you to delete or rename the infected file, or to continue without action. If you select this option, however, TbScan always continues. We recommend that you use a log file in such situations, since a scanning operation does not make much sense if you don't read the return messages (see the Log File Menu option below for further information).

Delete infected file.

By default, if TbScan detects a virus in a file it prompts you to delete or rename the infected file, or to continue without action. If you select this option, however, TbScan deletes the infected file automatically, without prompting you first. Use this option if you know your computer is infected by a virus and you want to erase all files the virus has infected. Make sure you have a clean backup and that you really want to get rid of all infected files at once.

Kill infected file.

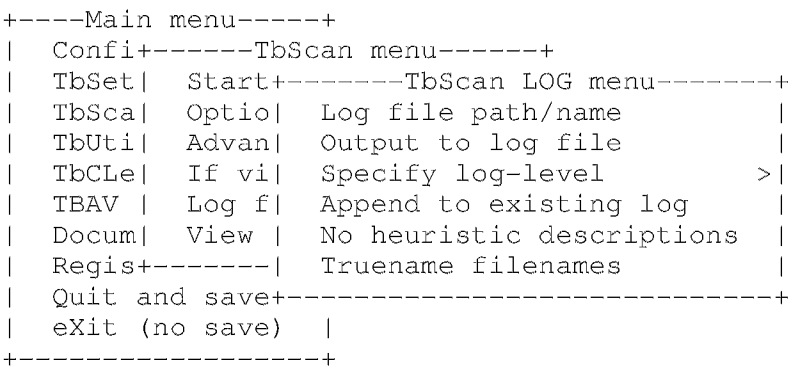
This option is almost the same as the "Delete infected file" option with one major difference. The DOS UNDELETE command enables you can recover a deleted file, but if you delete the infected file using this "Kill" option, recovery is no longer possible.

Rename infected file.

By default, if TbScan detects a file virus it prompts you to delete or rename the infected file, or to continue without action. If you select this option, however, TbScan renames the infected file automatically, without prompting you first. By default, TbScan replaces the first character of the file extension by the character 'V'. It names an .EXE file, to .VXE, for example, and a .COM file to .VOM. This prevents the execution of infected programs and thereby spreading the infection. This also enables you to keep the files for later examination and repair.

The "Log File Menu" Option

You can use the "TbScan Log Menu" to handle the results of the scan process (write them to a file or to a printer, for example). The menu appears below, followed by a description of the options.



Log file path/name.

Using this option you can specify the name of the log file you want to use. TbScan creates the file in the current directory unless you specify a path and filename. If the log file already exists, TbScan

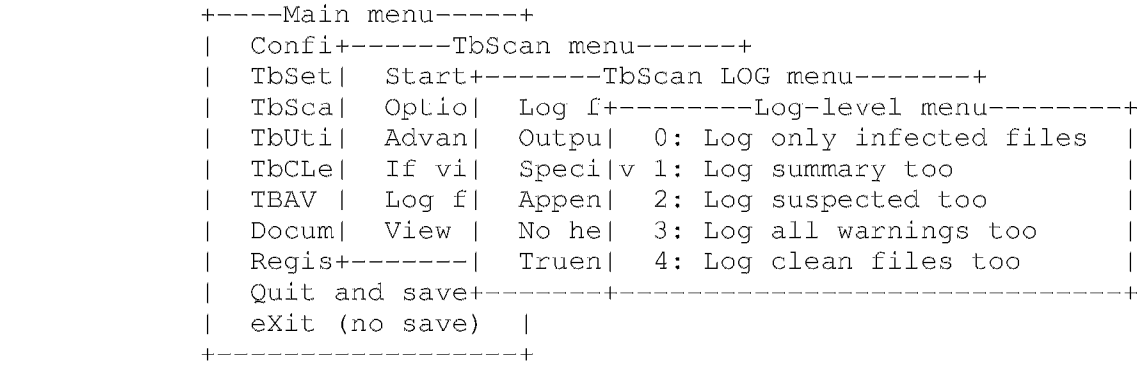
overwrites the file (unless you selected the "Append to existing log" option. If you want to print the results, you can specify a printer device name rather than a filename (LPT1 instead of C:\TBAV\TBSCAN.LOG, for example).
CAUTION:
To create the log file, you must select the "Output to log file" option.

Output to logfile.

When you select this option, TbScan creates a log file. The log file lists all infected program files, specifying heuristic flags (see Appendix B) and complete pathnames.

Specify log-level.

This option enables you to configure the actual contents of the log file using the following menu:



These levels determine what kind of file information TbScan notes in the log file. The default log level is 1, but you can select one of five levels:

0: Logonly infected files.

Specifies that if there are no infected files, do not create or change the log file.

1: Log summary too.

Places a summary and time stamp in the log file, and specifies that TbScan put only infected files in the log file.

2: Log suspected too.

This is almost the same as level 1, but TbScan also logs suspected files, files that would trigger the heuristic alarm if you specify the "High heuristic" sensitivity option.

3: Log all warnings too.

This level is an extension of the previous level. It specifies that TbScan log all files that have a warning character printed behind the filename.

4: Log clean files too.

This places the information of all files being processed into the log file.

Append to existing log.

If you select this option, TbScan appends new information to the existing log file instead of overwriting it. If you use this option often, we recommended that you delete or truncate the log file once in a while to avoid unlimited growth.

CAUTION:
To create the log file, you must select the "Output to log file" option.

No heuristic descriptions.

If you enable this option, TbScan does not specify the descriptions of the heuristic flags in the log file. See Appendix B for the heuristic flag descriptions.

Truename filenames.

If this option is specified, TbScan uses 'truenames' rather than DOS filenames. If you process a file on a network, accessed by DOS as F:\USER\FILE.EXE then TbScan will use the fully expanded filename (like \\SERVER2\PUBLIC\USER\FILE.EXE) on the screen and in the log file.

The "View Log File" Option

If you activate one of the above log file options, you can then select this option to view and study the log. Otherwise, this option is not available.

TIP:

See the "Configuring TBAV" section in Chapter 1 for how you can specify your own file viewer using the "Configure TBAV, File view utility" command.

3.2.3 Maximizing TbScan

Now that you know how to use TbScan's menus, you can more easily understand the power of using it from the command line.

When you run TbScan from the DOS command line, it recognizes command line options (often called "switches" in DOS terms). These options appear as "key-words" or "key-letters." The words are easier to memorize, so we will use these in this manual for convenience.

When you run TbScan, it looks for a file named TBAV.INI in the TBAV directory. If the keyword USEINI appears in the [TbScan] section of the TBAV.INI file, the other options listed in the [TbScan] section will be includede when you run TbScan from the command line.

CAUTION:

Be aware that if you specify options in the TBAV.INI file, you cannot undo them when you run TbScan from the command line.

The following table lists the TbScan command line options:

option parameter	short	explanation
help	he	help
pause	pa	enable Pause prompt
mono	mo	force monochrome output
quick	qs	quick scan (use ANTI-VIR.DAT)

allfiles	af	scan non-executables too
alldrives	ad	scan all local non-removable drives
allnet	an	scan all network drives
heuristic	hr	enable heuristic alerts
extract	ex	extract signatures (registered users only)
once	oo	scan only once a day
slowscroll	ss	enable conventional (slow) scrolling
secure	se	disable "user abort" (registered users only)
compat	co	maximum compatibility mode
ignofile	in	ignore no-file error
largedir	ld	use large directory table
fatcheck	fc	check the FAT for errors
fatinfo	fi	display amount of fragmented files
old	ol	disable the "This program is old" message
noboot	nb	skip boot sector check
nofiles	nf	skip scanning of files
nomem	nm	skip memory check
hma	hm	force HMA scan
nohmem	nh	skip UMB/HMA scan
nosub	ns	skip sub-directories
noautohr	na	auto heuristic level adjust
nowin	nw	do not scan for Windows-OS/2 viruses
repeat	rp	scan multiple diskettes
audio	aa	make noise if virus found
batch	ba	batch mode - no user input
delete	de	automatically delete infected files
kill	ki	automatically kill infected files
truename	tn	use true name instead of DOS name
log	lo	output to log file
append	ap	log file append mode
expertlog	el	no heuristic descriptions in log
logname=<filename>	ln	set path/name of log file
loglevel=<0...4>	ll	set log level
wait=<0...255>	wa	amount of timer-ticks to wait
rename[=<text-mask>]	rn	rename infected files
exec=.<ext-mask>	ee	specify executable extensions

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

TIP:

Remember that you can display these options from the command line by entering TBSCAN ?.

help (he).

If you specify this option TbScan displays the help as listed above.

pause (pa).

When you specify the PAUSE option, TbScan stops after it checks the contents of one window. This enables you to examine the results without having to consult a log file later.

mono (mo).

This option prevents TbScan from using colors in the screen output. This might enhance the screen output on some LCD screens or color-emulating monochrome systems.

quick (qs).

This option instructs TbScan to use the ANTI-VIR.DAT files to check for file changes since the last scan. TbScan scans only those files that have changed (CRC change) or do not appear in ANTI-VIR.DAT. By default, TbScan always scans files.

allfiles (af).

If you specify this option, TbScan also scans non-executable files (that is, files without a .COM, .EXE, .SYS or .BIN extension). If TbScan finds that such a file does not contain executable code, it "skips" that file. Otherwise, TbScan searches the file for COM, EXE and SYS signatures. Be aware that TbScan does not perform heuristic analysis on non-executable files. Since viruses normally do not infect non-executable files, it is not necessary to scan them. We recommend, in fact, that you do not use this option unless you have a good reason to scan all files since a file infected with a virus must normally be executed before it can perform what it is programmed to do, and since you can't execute a non-executable file, a virus in such a file cannot do anything. Some viruses write to non-executable files, but this is simply a result of "incorrect" programming or a specific targeted attack-- the result of which may be corrupted data, which will not likely harm other program or data files.

alldrives (ad).

This option instructs TbScan to scan all local non-removable disks.

allnet (an).

This option instructs TbScan to scan all network drives.

heuristic (hr).

While TbScan always performs a heuristic scan on the files being processed, if you select this option TbScan increases it's level of sensitivity. In this mode, TbScan detects 90% of the unknown viruses without any signatures. Be aware, however, that some false alarms might occur. See the "Understanding Heuristic Scanning" section later in this chapter for more information.

extract (ex).

This option is available to registered users only. See the "Using TbGenSig" section in Chapter 4 for more information.

once (oo).

If you specify this option, TbScan "remembers" whether it has run that day, and that if it has, it will not run again. In other words, this instructs TbScan to run only once a day, regardless of how many times you actually enter the command from the DOS prompt or a batch file. This command is very useful in your AUTOEXEC.BAT file, for example: TBSCAN @EVERYDAY.SCN ONCE RENAME. TbScan now scans the list of files and/or paths specified in the file EVERYDAY.SCN during the first boot-up of the day. If the systems boots more often that day, TbScan returns to the DOS prompt immediately. This option does not interfere with the regular use of TbScan. If you invoke TbScan without this option, it always runs, regardless of a previous run with the ONCE option set.

NOTE:
If TbScan cannot write to TBSCAN.EXE because it is flagged "read-only" or is located on a write-protected diskette, the ONCE option fails and the scanner executes without it.

slowscroll (ss).

If you specify this option, TbScan scrolls the files in the files window conventionally. This method is slower but looks more attractive.

secure (se).

This option is available to registered users only. If you use it, it is no longer possible to cancel TbScan by pressing Ctrl+Break, or to respond to a virus alert window.

compat (co).

If you select this option, TbScan attempts to be more compatible with your system. Use this option if the program does not behave as you would expect, or if it even halts the system. This option slows down the scanning process, so you should use it only if necessary. This option in no way affects the results of a scan.

ignofile (in).

If you specify this option and TbScan doesn't find any files, TbScan does not display the no files found message, nor does it exit with ERRORLEVEL 1. You might use this option for automatic contents scanning.

largedir (ld).

If TbScan's directory table runs out of space, which is very unlikely, you can use this option to allocate a large directory table.

fatcheck (fc).

If this option is specified, and TbScan is able to use its internal file system, it will check the disk(s) for lost clusters, cross linked clusters, invalid cluster numbers, and invalid allocation sizes. These errors often indicate system problems and need to be corrected as soon as possible. Because TbScan needs to read the FAT and all directories anyway, it can perform this important check without using additional time.

fatinfo (fi).

If this option is specified, TbScan displays the amount of fragmented files after it finished scanning. If the amount of fragmented files is high, you can increase the system performance by using a disk optimizer. This option can only be used in combination with option "fatcheck", and if TbScan is using its internal file system.

old (ol).

This option suppresses the message that appears if TbScan is 6 months old.

noboot (nb).

If you specify this option, TbScan does not scan the boot sector.

nofiles (nf).

This option disables the scanning of files. This can be useful if you are the victim of a boot sector virus and want to scan a large stack of diskettes as fast as possible.

nomem (nm).

If you specify this option, TbScan does not scan memory.

hma (hm).

By default, TbScan automatically detects the presence of an XMS-driver and scans HMA. If you have an HMA-driver that is not compatible with the XMS standard, you can use this option to force TbScan to scan HMA.

nohmem (nh).

By default, TbScan identifies RAM beyond the DOS limit and scans it. This means that it scans video memory and the current EMS pages. You can, therefore, use this option to disable the scanning of non-DOS memory.

nosub (ns).

By default, TbScan searches sub-directories for executable files, unless you specify a filename (wildcards allowed). If you enable this option, TbScan does not scan sub-directories.

noautohr (na).

TbScan automatically adjusts the heuristic detection level after it locates a virus. In other words, when TbScan finds a virus, it continues as if you used the HEURISTIC option. This provides you maximum detection capabilities in case you need it, while keeping the amount of false alarms to a minimum. If you don't want this, you can specify option NOAUTOHR.

nowin (nw).

By default, TbScan scans Windows and OS/2 files for viruses. Removing the checkmark disables Windows and OS/2 file scanning.

repeat (rp).

This option is very useful if you want to check a large amount of diskettes. Instead of returning to DOS after checking a disk, TbScan prompts you to insert another disk in the drive.

audio (aa).

This enables an audible alarm sound when TbScan finds a virus.

batch (ba).

By enabling this option, TbScan scans without displaying any messages. If you use this option, we recommend that you use a log file (see the LOG option below).

delete (de).

By default, if TbScan detects a virus in a file, it prompts you to delete or rename the infected file, or to continue without action. If you specify this option, however, TbScan deletes the infected file automatically, without prompting you first. Use this option if you know there is a virus infection. Make sure that you have a clean backup, and that you really want to get rid of all infected files at once.

kill (ki).

By default, if TbScan detects a virus in a file it prompts you to delete or rename the infected file, or to continue without action. If you specify the DELETE option, TbScan deletes the infected file automatically, without prompting you first. Unlike the DELETE option, however, KILL prevents files from being undeleted. Be careful if you use this option. Make sure you have a clean backup!

truename (tn).

This option instructs TbScan to use "truenames" rather than DOS names. For example, if you process a file on a network that DOS accesses using the name F:\USER\FILE.EXE, TbScan uses the full name \\SERVER\PUBLIC\USER\FILE.EXE on the screen and in the log.

log (lo).

When you use this option, TbScan creates a log file. The log file lists all infected program files, specifying heuristic flags (see Appendix B) and complete pathnames.

append (ap).

If you use this option, TbScan appends new information to an existing log file rather than overwriting it. If you use this option often, we recommend that you delete or truncate the log file occasionally to avoid unlimited growth.

NOTE:
If you use this option, you must also use the LOG option.

expertlog (el).

If you enable this option, TbScan does not specify the descriptions of the heuristic flags in the log file. Appendix B lists the heuristic flag descriptions

logname =<filename> (ln).

Using this option, you can specify the name of the log file you want to use. TbScan creates the file in the current directory unless you specify a path and filename after selecting this option. If the log file already exists, TbScan overwrites it. If you want to print the results, you can specify a printer device name rather than a filename (for example, you can specify LOGNAME=LPT1).

NOTE:
If you use this option, you must also use the LOG option.

loglevel =<0..4> (ll).

These levels determine what kind of file information the log file stores. The default log level is 1, but you can select one of five log levels:

0 : Log only infected files.

This specifies that if there are no infected files, do not create or change the log file.

1 : Log summary too.

This places a summary and time stamp in the log file, and specifies that TbScan put only infected files in the log file.

2 : Log suspected too.

This is almost the same as level 1, but TbScan also logs "suspected files," files that would trigger the heuristic alarm if you specify the "High heuristic" sensitivity option.

3 : Log all warnings too.

This level is an extension of the previous level. It specifies that TbScan log all files that have a warning character printed behind the filename.

4 : Log clean files too.

This places the information of all files being processed into the log file.

NOTE:
If you use this option, you must also use the LOG option.

wait =<0..255> (wa).

Use this option to delay TbScan. This might be handy if you want to scan a very busy network but don't want to occupy the network too heavily. You have to specify the amount of timer ticks you want to insert between scanned files.

rename [=<ext-mask>] (rn).

By default, if TbScan detects a file virus, it prompts you to delete or rename the infected file, or to continue without action. If you select this option, TbScan renames the infected file automatically, without prompting you first. Also by default, TbScan replaces the first character of the file extension with the character 'V.' It renames an .EXE file to .VXE, for example, and a .COM file to .VOM. This prevents the execution of infected programs and thereby prevents spreading the infection. This option also enables you to keep the infected files for later examination and repair. You can also add a parameter to this option specifying the target extension. This parameter should always contain three characters; you can use question marks. The default target extension is "V??."

exec =.<ext-mask> (ee).

Using this option you can add filename extensions that indicate what files are executable. If you want to use this option, you probably want to put it in the configuration file. Refer to the "Advanced

Options" Option section earlier in this chapter for an explanation of configuring executable extensions.

Here are a few examples using TbScan from the DOS command line.

1. This command:

   TBSCAN C:\ NOBOOT

   scans all executable files in the root directory and its subdirectories and skips the boot sector scan.

2. This command:

   TBSCAN *.*

   scans all files in the root directory but does not process subdirectories.

3. This command:

   TBSCAN C:\ LOG LOGNAME=C:\TEST.LOG LOGLEVEL=2

   scans all executable files on drive C: and creates a LOG file named C:\TEST.LOG that contains all infected and suspected files.

4. This command:

   TBSCAN \ LOG LOGNAME=LPT1

   scans the root directory and its subdirectories and then redirects the results to the printer instead of a log file.

3.2.4 Understanding the Scanning Process

This section adds to your knowledge of TbScan by explaining a little more about the scanning process. TbScan starts scanning immediately whenever you run it from the DOS command line or select the Start Scanning option in the TbScan Menu. As TbScan begins its scan, your screen will look similar to the following:

TbScan divides the screen into three windows: an information window (at the top), a scanning window (the bottom-left window) and a status window (to the right of the scanning window). The information window initially displays the vendor information only.

```
+-----------------------------------------------------------------+
|Thunderbyte virus detector          (C) 1989-95, Thunderbyte B.V. |
|                                                                 |
| TBAV is upgraded every two months. Free hotline support is      |
| provided for all registered users via telephone, fax and        |
| electronic bulletin board. Read the comprehensive documentation |
| files for detailed info.                                        |
|                                                                 |
| C:\DOS\                                                         |
|                                                                 |
| ANSI.SYS       scanning..>         OK    signatures:        986  |
| COUNTRY.SYS    skipping..>         OK                            |
| DISKCOPY.COM   tracing...>         OK    file system:       OWN  |
| DISPLAY.SYS    scanning..>         OK                            |
| DRIVER.SYS     scanning..>         OK    directories:        01  |
| EGA.CPI        skipping..>         OK    total files:       17  |
| FASTOPEN.EXE   looking...>         OK    executables:       12  |
| FDISK.EXE      looking...>         OK    CRC verified:      10  |
| FORMAT.COM     tracing...>    E    OK    changed files:     00  |
| GRAFTABL.COM   tracing...>         OK    infected items:    00  |
| GRAPHICS.COM   tracing...>         OK                            |
| GRAPHICS.PRO   skipping..>         OK    elapsed time:   00:05  |
|                                          Kb /second:        57  |
+-----------------------------------------------------------------+
```

If TbScan detects infected files, it displays the names of the file and the virus in the upper window. The lower left window displays the names of the files being processed, the algorithm in use, information and heuristic flags, and finally an OK statement or the name of the virus detected.

Notice the following example:

```
     NLSFUNC.EXE     checking..>    FU          OK
        |              |            |           |
        |              |            |           result of scan
        |              |            heuristic flags
        |              algorithm being used to process file
        name of file in process
```

You will see comments following each file name, such as: "looking," "checking," "tracing," "scanning," or "skipping." These refer to the various algorithms being used to scan files.

Other comments that TbScan displays here are the heuristic flags. Consult the Understanding Heuristic Flags section later in this chapter and Appendix B for more information on these warning characters.

The lower right window is the status window. It displays the number of files and directories encountered as well as the number of viruses found. It also displays which file system is being used: either DOS or OWN. The latter means that TbScan is able to bypass DOS. If this is the case, TbScan reads all files directly from disk for extra security and speed.

You can abort the scanning process by pressing the two keys Ctrl+Break simultaneously (that is, if you didn't specify the "SECURE" option).

When TbScan detects an infected program, it displays the name of the virus. If you did not specify the BATCH, RENAME or DELETE options, TbScan prompts you to specify the appropriate action. If you choose to rename the file, TbScan replaces the first character of the file extension with the character 'V.' This prevents you or someone else from accidentally executing the file before you can investigate it more thoroughly.

If TbScan detects an infected file, it displays one of the following messages:

[Name of file] is infected by [name of virus] virus.

The file is infected by the virus mentioned.

[Name of file] is Joke named [name of Joke].

Some programs simulate that the system is infected by a virus; such a program is a "joke." A joke is completely harmless; however it causes confusion and might cause people to stop using the computer, and should therefore be removed..

[Name of file] is Trojan named [name of Trojan].

The file is a Trojan Horse. A Trojan Horse is a program that pretends to be a harmless program (like a game) but it is designed to do something harmful like erasing a disk. Some Trojan Horses also

install viruses on your system. Do not execute the program, but delete it instead.

[Name of file] damaged by [name of virus].

Unlike an infected file, which carries the virus itself, a damaged file has only been damaged by the virus.

[Name of file] dropper of [name of virus].

A "dropper" is a program that has not been infected itself, but which does contain a boot sector virus and is able to install it into your boot sector.

[Name of file] garbage: (not a virus) [name of garbage].

A "garbage" program is a file that does not work because it is badly damaged or may have been overwritten with "garbage." Some virus collections (i.e. a CD-ROM based virus collection) contain "garbage-like" program code that was designed specifically to trigger virus detection programs (and fool them), which is exactly why ThunderBYTE identifies them as "garbage."

It is also possible for TbScan to encounter a file that appears infected by a virus, although it could not find a signature. In this case TbScan displays the prefix "Probably" before the message.

If TbScan finds a file to be suspicious and displays a virus alert window, you can avoid future false alarms by pressing V (Validate program). Note that this works only if there is an ANTI-VIR.DAT record of the file available. Once TbScan validates a program, the program is no longer subject to heuristic analysis, unless the program changes and no longer matches the ANTI-VIR.DAT record. This will be the case if such a file becomes infected at a later time. In such a case, TbScan still reports infections on these files.

NOTE:
Be aware that a validated program is still subject to the conventional signature scanning.

If you specify the HEURISTIC or the HIGH HEURISTIC SENSITIVITY option, it is likely that TbScan will find some files that look like a virus. In

this case, TbScan uses the prefix "Might be" to inform you about it. So, if TbScan displays:

[Name of file] Probably infected by an unknown virus

or:

[Name of file] Might be infected by an unknown virus

it does not necessarily mean that the file is infected. There are a lot of files that look like a virus but are not.

It is extremely important to understand that false alarms are part of the nature of heuristic scanning. In its default mode, it is very unlikely that TbScan will issue a false alarm. If you specify the HEURISTIC option, however, some false alarms might occur.

How should you deal with false alarms? If TbScan thinks it has found a virus, it tells you the reason for this suspicion. In most cases you will be able to evaluate these reasons when you consider the purpose of the suspected file.

NOTE:
Viruses infect other programs. It is, therefore, unlikely that you will find only a few infected files on a hard disk you use frequently. You should ignore the result of a heuristic scan if only a few programs on your hard disk trigger it. If, on the other hand, your system behaves "strangely" and several programs trigger the TbScan alarm with the same serious flags, your system could very well be infected by a (yet unknown) virus.

3.2.5 Understanding Heuristic Flags

Heuristic flags consist of single characters that appear behind the name of the file that just scanned. There are two kinds of flags: the informative ones, which appear in lower-case characters, and the more serious flags, which appear in upper-case characters.

The lower-case flags indicate special characteristics of the file being scanned, whereas the upper-case warnings might indicate a virus. If the loglevel is 3 or above, the important warnings not only appear as a warning character, but TbScan also adds a description to the log file.

How should you treat the flags? You can consider the less important lower-case flags to be informational only; they provide file information

you might find interesting. The more serious uppercase warning flags MIGHT (we repeat, MIGHT) indicate a virus. It is quite normal that you have some files in your system that trigger an uppercase flag.

NOTE:
Appendix B lists the heuristic flag descriptions.

3.3 Using TbDriver

TbDriver is a small memory-resident (TSR) program that you must load before any of the other TBAV memory-resident utilities. This brief section explains the use of TbDriver.

3.3.1 Understanding TbDriver

By itself, TbDriver does not provide much protection against viruses, rather its use is to enable the memory resident ThunderBYTE Anti-Virus utilities, such as TbScanX, TbCheck, TbMem, TbFile, and TbDisk, to perform properly. It is the source for some of the routines these utilities have in common, including: support to generate the pop-up window routines, driving the translation unit that enables the possibility of displaying messages in your native language, and support for networks. Additionally, TbDriver also contains basic protection against "stealth" viruses and against "ANSI bombs."

NOTE:
See the NOFILTER option below for an explanation of an ANSI bomb.

3.3.2 Working with TbDriver

You must load TbDriver before loading any of the other memory-resident TBAV utilities. If you ran the TBAV Install program, TbDriver is already set up to load automatically when you boot. Your AUTOEXEC.BAT file calls the TBSTART.BAT file, which in turn loads TbDriver.

If you prefer, you can load TbDriver directly from the command line or from an individual line in AUTOEXEC.BAT by using this command:

```
<PATH>TBDRIVER
```

If TbDriver resides in the TBAV directory on drive C:, for example, you could enter C:\TBAV\TBDRIVER.

An even more secure way to load TbDriver, and the other TBAV memory-resident utilities (which we ll examine in more detail in the Using TbScanX section later in this chapter), is to load it via the CONFIG.SYS file. After removing the call to TBSTART.BAT in AUTOEXEC.BAT, you could put the following command in CONFIG.SYS:

```
DEVICE=<PATH>TBDRIVER.EXE
```

If TbDriver resides in the TBAV directory on drive C:, for example, you could enter DEVICE=C:\TBAV\TBDRIVER.EXE.

TIP:

If you want protection against ANSI-bombs, you should load TbDriver AFTER the ANSI.SYS driver. Also, if you install TbDriver on a machine that boots from a boot ROM, specify the message file with the drive and path where it resides AFTER the machine boots. The default message file will no longer be accessible after the machine boots.

3.3.3 Maximizing TbDriver

This section describes how to use TbDriver's option to maximize its performance and how to get foreign language support for the TBAV utilities.

When you run TbDriver from the DOS command line, it recognizes command line options (often called "switches" in DOS terms). These options appear as "key-words" or "key-letters." The words are easier to memorize, so we will use these in this manual for convenience.

TbDriver enables you to specify loading options on the command line. It treats a filename specification as a language file specification (see the following "Getting Language Support" section).

The first three options in the following table are always available. The other options are available only if TbDriver is not already memory resident. The command-line syntax is as follows:

```
TBDRIVER [<PATH>][<FILENAME>]... [<OPTIONS>]...
```

TbDriver recognizes the following options:

option parameter	short	explanation
help	?	help
net	n	force LAN support
remove	r	remove TbDriver from memory
mode=<m|c>	m	override video mode (mono|color)
freeze	j	freeze the machine after an alert
lcd	l	enhance output on LCD screens
noavok=<drives>	o	assume permission for specified drives when ANTI-VIR.DAT record is missing
quiet	q	do not display activity
secure	s	do not allow permission updates
notunnel	t	do not detect tunneling
nofilter	f	do not filter dangerous ANSI codes
nostack	ns	do not install a stack

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

TIP:
Remember that you can display these options from the command line by entering TBDRIVER ?.

help (?).

If you specify this option, TbDriver shows you the valid command line options as listed above.

net (n).

TbDriver cooperates well with most networks. In normal situations you will not need the NET option at all. You should use it only if both the following conditions are true at the same time:

1. You make a connection to a Novell network and TBDRIVER.EXE before using the logon command.

2. There is no valid ANTI-VIR.DAT record in the directory where the NET?.COM program resides or after renaming the NET?.COM file.

remove (r).

This option disables TbDriver and attempts to remove the resident part of its code from memory and return this memory space to the system.

Unfortunately, this works only if you loaded TbDriver last. An attempt to remove a TSR after you load another TSR leaves a useless gap in memory and could disrupt the interrupt chain. TbDriver checks whether it is safe to remove its resident code; if not, it simply disables itself.

mode (m).

On dual video systems TbDriver uses the currently active screen. It might be forced to use the alternate screen with the MODE=M option for monochrome or the MODE=C option for color systems.

lcd (l).

This option enhances the output on LCD screens.

freeze (f).

This option freezes the computer when there is a virus alert.

noavok (o).

We don't recommend this option for normal use. You might need it to grant permission automatically for programs without an ANTI-VIR.DAT record. The option requires a parameter specifying the drives to which the default permission applies. If, for example, you do not want TbMem to display a message when a TSR without ANTI-VIR.DAT executes from drive E: and F:, you could specify NOAVOK=EF on the TbDriver command line. Additionally, if you want to exclude network drives, you should use an asterisk [*]. For example, if you want to grant permission for all files without ANTI-VIR.DAT records on drive A:, your ram disk F: and your remote network drives, specify NOAVOK=AF*.

quiet (q).

Some resident TBAV utilities display an activity status. TbScanX, for instance, displays a rectangle with the word Scanning in the upper left corner of your screen while scanning a file. The QUIET option disables this message.

secure (s).

Some ThunderBYTE utilities can store permission flags in the ANTI-VIR.DAT files. You can use this option if you don't want these flags changed. It has no effect on flags already set, so you can use the option after installing new programs or packages.

notunnel (t).

"Tunneling" is a technique viruses apply to determine the location of the DOS system code in memory, and to use that address to communicate with DOS directly. This inactivates all TSR programs, including resident anti-virus software. TbDriver is able to detect these tunneling attempts, and informs you about it. Some other anti-virus products also rely on tunneling techniques to bypass resident viruses, thereby causing false alarms. If you are currently executing other anti-viral products, the NOTUNNEL option disables TbDriver's tunneling detection.

nofilter (f).

The original ANSI driver has a feature to assign text strings to keys. Years ago people used this feature, for example, to assign the DIR /W command to the F10 key. Such reprogramming can be done simply by embedded ANSI codes in batch files. Almost no one uses this feature nowadays. Some misguided people, however, use this feature, for example, to make a text file that reprograms the Enter key to execute the DEL *.* command or something even worse. Such a file is an "ANSI-bomb." TbDriver protects you against ANSI-bombs by filtering out the keyboard reprogramming codes. All other ANSI codes pass without interference. If you don't want this protection, or if you want to use this obsolete ANSI feature, you can use the NOFILTER option.

nostack (ns).

By default, TbDriver maintains a stack for the resident TBAV utilities. For most systems, however, this isn't necessary. If you

use this option, TbDriver uses the application stack, saving a few hundred bytes of memory. If the system hangs or becomes unstable, however, discontinue use of this option.

You can use the optional filename specification to direct TbDriver to the location of the language file you want to use. TbDriver retrieves pop-up window messages from a TBDRIVER.LNG file, which it expects to find in its own home directory. The default English language file is TBDRIVER.LNG, which you can replace with a file in your local language. You can order separate language support packages at your local ThunderBYTE dealer, or download the language file from a ThunderBYTE support BBS. See the Maintaining the System section in Chapter 1 for more information about the ThunderBYTE support BBS.

To load a language file, either rename it to the default (TBDRIVER.LNG), or specify the full path and filename following the command. You can also switch to another language by calling TbDriver again with a different message file. This will not take up any extra memory.

3.4 Using TbScanX

TbScanX is virtually identical to TbScan, with one important difference: TbScan is memory-resident. This section describes TbScanX in detail.

3.4.1 Understanding TbScanX

To implement real-time or on-the-fly virus protection, the TBAV for DOS utilities include the TbScanX program, a memory-resident (TSR) program that tracks all file operations. If you copy an infected file from a diskette to your hard disk, for example, TbScanX recognizes the virus hidden in the file and informs you about it, BEFORE the virus becomes active.

Why use TbScanX? Let's assume you have a virus scanner that automatically runs from your AUTOEXEC.BAT file. If it doesn't find any viruses, your system should be uninfected. Right? Not necessarily. To be sure that no virus infects your system, you need to execute the scanner every time you copy a file to your hard disk, after downloading a file from a bulletin board system, or after unarchiving an archive such as a ZIP file. Now be honest, do YOU invoke your scanner every time you introduce a new file into the system? If you don t, you take the risk that within a couple of hours all files will become infected by a virus.

Once you load TbScanX, it remains resident in memory and automatically scans all files you execute and all executable files you copy, create, download, modify, or unarchive. It uses the same approach to protect against boot sector viruses; every time you put a diskette into a drive, TbScanX scans the boot sector. If the disk is contaminated with a boot sector virus, TbScanX warns you in time!

NOTE:
TbScanX is fully network compatible. It does not require you to reload the scanner after logging onto the network.

3.4.2 Working with TbScanX

Since TbScanX is memory resident, you can execute and configure the program from the command line or from within a batch file. It is important to load TbScanX as early as possible after the machine boots. We therefore recommend that you execute TbScanX from within the CONFIG.SYS file.

CAUTION:
TbScanX requires that you load TbDriver first! See the previous section on "Using TbDriver" for details.

There are three possible ways to load TbScanX:

1. From the DOS prompt or within the AUTOEXEC.BAT file:

   <PATH>TBSCANX

2. From the CONFIG.SYS files as a TSR (DOS 4+ and above):

   INSTALL=<PATH>TBSCANX.EXE

   The INSTALL= CONFIG.SYS command is NOT available in DOS 3.xx.

3. From the CONFIG.SYS as a device driver:

   DEVICE=<PATH>TBSCANX.EXE

NOTE:
Using TbScanX as a device driver does not work in all OEM versions of DOS. If it does not work, use the INSTALL= command or load TbScanX from within the AUTOEXEC.BAT. TbScanX should always work correctly if you run it from AUTOEXEC.BAT.

Unlike other anti-virus products, you can load the ThunderBYTE Anti-Virus Utilities before starting a network without losing the protection afterwards.

In addition to the three loading possibilities, you can also load TbScanX into an available UMB (upper memory block) if you are using DOS version 5 or higher. To accomplish this from AUTOEXEC.BAT, use the following command:

LOADHIGH <PATH>TBSCANX

Alternately, to accomplish this from CONFIG.SYS, use the following command:

DEVICEHIGH=<PATH>TBSCANX.EXE

If you are using Microsoft Windows, you should load TbScanX BEFORE starting Windows. When you do this, there is only one copy of TbScanX in memory regardless of how many DOS windows you might open. Every DOS

window (that is, every virtual machine) has a fully functional copy of TbScanX running in it.

TbScanX automatically detects if Windows is running, and switches itself in multitasking mode if necessary. You can even disable TbScanX in one window without affecting the functionality in another window.

NOTE:
TBAV for Windows includes a full-featured resident scanner. Please refer to the TBAV for Windows documentation for more information.

3.4.3 Maximizing TbScanX

When you run TbScanX from the DOS command line, it recognizes command line options (often called "switches" in DOS terms). These options appear as "key-words" or "key-letters." The words are easier to memorize, so we will use these in this manual for convenience.

You can maximize TbScanX's performance by using one or more command line options. The first four options in the following table are always available. The other options are available only if TbScanX is not already resident in memory.

option parameter	short	explanation
help	?	display on-line help
off	d	disable scanning
on	e	enable scanning
remove	r	remove TbScanX from memory
noexec	n	never scan at execute
allexec[=<drives>]	a	always scan at execute
noboot	b	do not scan boot sectors
wild	w	only search viruses which appear "in the wild"
ems	me	use expanded memory (EMS)
xms	mx	use extended memory (XMS)
secure	s	deny all suspicious operations
lock	l	lock PC when a virus is detected
api	i	load TbScanX's Application Program Interface
compat	c	increase compatibility

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

TIP:
Remember that you can display these options from the command line by entering TBSCANX ?.

help (?).

This option displays the command line options as shown above. Once you load TbScanX, however, this option does not display all the options.

off (d).

This option disables TbScanX, but leaves it in memory.

on (e).

This option re-enables TbScanX after you disable it with the OFF option.

remove (r).

This option disables TbScanX and attempts to remove the resident part of its code from memory and return this memory space to the system. Unfortunately, this works only if you loaded TbScanX last. An attempt to remove a TSR after you load another TSR leaves a useless gap in memory and could disrupt the interrupt chain. TbScanX checks whether it is safe to remove its resident code; if not, it simply disables itself.

noexec (n).

TbScanX normally scans files located on removable media just before they execute. You can use this option to disable this feature completely.

allexec (a).

TbScanX normally scans executable files only if they reside on removable media. It "trusts" files on the hard disk, since these files must have been copied or downloaded before, and since by this time TbScanX has already scanned them automatically. If you want to

scan every file before it executes, however, regardless of whether it is on the hard disk or removable media, you should use this option. It is possible to explicitly specify drives from which you want executed files to be scanned. For example, if you specify option ALLEXEC=DF, then TbScanX will only scan files being executed that reside on either drive D: or drive F:.

noboot (b).

TbScanX automatically monitors the disk system. Every time DOS reads the boot sector, TbScanX scans the disk for boot sector viruses. If you change a disk, DOS first reads the boot sector; otherwise it does not know what kind of disk is in the drive. As soon as DOS reads the boot sector, TbScanX checks it for viruses. If you don't like this feature, or if it causes problems, you can switch it off using the NOBOOT option.

wild (w).

TbScanX can distinguish viruses that do not appear "in the wild" from frequently appearing viruses. In order to reduce the memory requirements of TbScanX, you can specify option WILD, which makes TbScanX load and use the viruses signatures from viruses that frequently appear "in the wild." This option is disabled by default.

ems (me).

If you specify this option, TbScanX uses expanded memory (such as that provided by the LIM/EMS expansion boards or 80386 memory managers) to store the signatures and part of its program code. Since conventional memory is more valuable to your programs than expanded memory, we recommend the use of EMS memory. TbScanX can use up to 64Kb of EMS memory. (Refer to the XMS option also.)

xms (mx).

If you specify this option TbScanX uses extended memory to store the signatures and part of its program code. An XMS driver (such as DOS's HIMEM.SYS) must be installed to be able to use this option. XMS memory is not directly accessible from within DOS, so every time TbScanX has to scan data it has to copy the signatures to conventional memory. To be able to save the original memory contents, TbScanX needs a double amount of XMS memory. Swapping to

XMS is slower than swapping to EMS memory, so if you have EMS memory available, we recommend swapping to EMS. Swapping to XMS might conflict with some other software, so if you experience problems try using TbScanX without the XMS option.

secure (s).

TbScanX normally asks you to continue or to cancel when it detects a virus. In some business environments, however, employees should not make this choice. By using the SECURE option, you can disallow suspicious operations.

NOTE:
This option also disables the OFF and REMOVE options.

lock (l).

If you are a system operator, you can use this option to instruct TbScanX to lock the system when it detects a virus.

api (i).

This option is for advanced users only. It enables TbScanX's Application Program Interface (API), which is necessary if you want to call TbScanX from within your application. Consult the ADDENDUM.DOC file for detailed programming information.

compat (c).

In most systems TbScanX performs trouble free. Another TSR program, however, might conflict with TbScanX. If you load the other TSR first, TbScanX normally detects the conflict and uses an alternate interrupt. If, on the other hand, you load the other TSR after TbScanX, and it aborts with a message telling you that it is already loaded, you can use the COMPAT switch of TbScanX (when installing it in memory). It is also possible for TbScanX to conflict with other resident software that is using EMS or XMS. In this case, the system will hang. Again, the COMPAT option solves this problem, but be aware that due to extensive memory swapping, TbScanX's performance will slow down.

TIP:

> If you are using DOS version 5 or above and have extended memory (XMS) on your system, you can use EMM386.SYS to treat a portion or extended memory as expanded memory (EMS). See your DOS manual for details.

Here is one example of loading TbScanX:

```
DEVICE=C:\TBAV\TBSCANX.EXE XMS NOBOOT
```

In this example, the memory resident portion of TbScanX loads into extended memory (XMS) and will not scan boot sectors for viruses.

3.4.4 Understanding the Scanning Process

This section adds to your knowledge of TbScanX by explaining a little more about the scanning process.

Whenever a program tries to write to an executable file (files with the extensions .COM and .EXE), you will briefly see the text "*Scanning*" in the upper left corner of your screen. As long as TbScanX is scanning, this text appears. Since TbScanX takes very little time to scan a file, the message appears very briefly. The text "*Scanning*" also appears if you execute a program directly from a diskette, and if DOS accesses the boot sector of a diskette drive.

If TbScanX detects a suspicious signature that is about to be written into a file, a window appears similar to the one displayed below:

```
+---------TBAV interception---------+
|              WARNING!              |
| TbScanX detected that COMMAND.COM  |
| is infected with                   |
| Yankee_Doodle {1}                  |
| Abort? (Y/N)                       |
+------------------------------------+
```

Whenever this message appears, you should press N to continue, or any other key to abort. If TbScanX detects a suspicious signature in a boot sector, it displays a message like the following:

```
+------------TBAV interception-----------+
|                 WARNING!                |
| TbScanX detected that the bootsector    |
| of disk in drive A: is infected with    |
```

```
| Form                                    |
| Do NOT attempt to boot with that disk! |
+-----------------------------------------+
```

Although a virus seems to be in the boot sector of the specified drive, the virus cannot do anything since it has not yet executed. If you reboot the machine with the contaminated diskette in the drive, however, the virus copies itself into memory and onto your hard disk.

NOTE:

To display the name of a virus, TbScanX needs access to the virus signature file (TBSCAN.SIG). If for any reason TbScanX cannot access this file, it still detects viruses, but no longer displays the name of the virus. It displays "[Name unknown]" instead.

3.5 Using TbCheck

This section describes another one of TBAV's memory resident (TSR) utilities, TbCheck.

3.5.1 Understanding TbCheck

TbCheck is a memory-resident integrity checker that comes into action whenever the system is about to execute a file. It uses the ANTI-VIR.DAT records TbSetup generates to detect file changes, which is often the first sign of a virus infection. These records contain information, such as file sizes and checksums, of every executable file in a directory. By comparing this information with the actual file status, it is possible to detect automatically any changes, including infections caused by viruses.

Assume your AUTOEXEC.BAT file automatically loads a conventional integrity checker. If no files appear changed, your system should be uninfected, but to be sure that no virus can infect your system, you have to execute the checker frequently. In contrast, once you load TbCheck, it remains resident in memory, and automatically checks all programs you try to execute.

NOTE:
TbCheck is fully network compatible. It does not require you to reload the checker after you are logged onto the network.

3.5.2 Working with TbCheck

Since TbCheck is a memory resident program, you can execute and configure it from the DOS command line or from within a batch file. You should, however, load TbCheck automatically when the computer boots, preferably during the execution of AUTOEXEC.BAT, or better yet, CONFIG.SYS.

CAUTION:
Be sure to load TbDriver before trying to load TbCheck. TbCheck will refuse to load without it.

There are three possible ways to start TbCheck:

1. From the DOS prompt or within the AUTOEXEC.BAT file:

   <PATH>TBCHECK

2. From CONFIG.SYS as a TSR (DOS 4 or above):

```
INSTALL=<PATH>TBCHECK.EXE
```

The INSTALL= CONFIG.SYS command is NOT available in DOS 3.xx.

3. From CONFIG.SYS as a device driver:

```
DEVICE=<PATH>TBCHECK.EXE
```

NOTE:
Executing TbCheck as a device driver does not work in all OEM versions of DOS. If it doesn't work, use the INSTALL= command or load TbCheck from AUTOEXEC.BAT. TbCheck should always work correctly if you load it from AUTOEXEC.BAT. Also, unlike other anti-virus products, you can load the ThunderBYTE Anti-Virus utilities before starting a network without losing the protection after the network is started.

In addition to the three loading possibilities, if you are using DOS version 5 or above, you can load TbCheck into an available UMB (upper memory block) from AUTOEXEC.BAT using this command:

```
LOADHIGH <PATH>TBCHECK
```

You can also load TbCheck into high memory from within the CONFIG.SYS using this command:

```
DEVICEHIGH=<PATH>TBCHECK.EXE
```

If you are using Microsoft Windows, you should load TbCheck BEFORE starting Windows. When you do this, there is only one copy of TbCheck in memory regardless of how many DOS windows you might open. Every DOS window (that is, every virtual machine) has a fully functional copy of TbCheck running in it.

TbCheck automatically detects if Windows is running, and switches itself into multi-tasking mode if necessary. You can even disable TbCheck in one window without effecting the functionality in another window.

NOTE:
TBAV for Windows comes with a full-fledges Windows-based version of TbCheck. Please refer to the documentation of TBAV for Windows for more information.

3.5.3 Maximizing TbCheck

When you run TbCheck from the DOS command line, it recognizes command line options (often called "switches" in DOS terms). These options appear as "key-words" or "key-letters." The words are easier to memorize, so we will use these in this manual for convenience.

You can maximize TbCheck's performance by using it's various options. The first four options in the following table are always available. The other options are available only if TbCheck is not yet memory resident.

option parameter	short	explanation
help	?	display on-line help
remove	r	remove TbCheck from memory
off	d	disable checking
on	e	enable checking
noavok [=<drives>]	o	do not warn for missing ANTI-VIR.DAT record
fullcrc	f	calculate full CRC (slow!)
secure	s	do not execute unauthorized files

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

TIP:
Remember that you can display these options from the command line by entering TBCHECK ?.

help (?).

Specifying this option displays the above options list.

remove (r).

This option disables TbCheck and attempts to remove the resident part of its code from memory and return this memory space to the system. Unfortunately, this works only if you loaded TbCheck last. An attempt to remove a TSR after you load another TSR leaves a useless gap in memory and could disrupt the interrupt chain. TbCheck checks whether it is safe to remove its resident code; if not, it simply disables itself.

off (d).

This option disables TbCheck, but leaves it in memory.

on (e).

This re-enables TbCheck after having been disabled with the OFF option.

noavok (o).

TbCheck looks in the ANTI-VIR.DAT file for checksum information on the file you want to check. TbCheck displays a message if it finds no checksum information or if the specific checksum is incorrect. This ensures that you will receive a warning whenever a malicious program deletes the ANTI-VIR.DAT file. Although we recommend that you maintain ANTI-VIR.DAT files on all drives, this might not always be practical with floppy disks, RAM disks, or CD-ROM disks. This option, therefore, tells TbCheck not to look for an ANTI-VIR.DAT on specific drives. For example, if you don't want TbCheck to alert you about the absence of an ANTI-VIR.DAT record on floppy disks A: and B: or on your RAM disk E:, you should load TbCheck using the following command line:

```
<PATH>TBCHECK NOAVOK=ABE
```

If you don't want a message when an ANTI-VIR.DAT record is missing on network drives, you should specify an asterisk (*) instead of a drive letter. If you don't specify a drive to the NOAVOK option, TbCheck never issues a warning if an ANTI-VIR.DAT record is missing on any drive.

CAUTION:

This presents a security hole for viruses: by deleting the ANTI-VIR.DAT file you will not be able to detect file changes caused by a viral infection. Also, please note that the NOAVOK option does not prevent the detection of infected programs if the ANTI-VIR record is available. If a program has changed and the ANTI-VIR record is available, you will still get an alarm regardless of how you implement the NOAVOK option.

fullcrc (f).

By default, TbCheck verifies only that part of the file near the program's entry point. If a virus infects the file, this area will definitely change, so this is perfectly adequate to detect all

infections. Other file changes, notably configuration variations, will not trigger the alarm. If, however, you should ever desire a full check that detects ANY file changes, this option takes care of it. Be aware that this option slows down the system considerably, so we don't recommend its use in normal circumstances.

secure (s).

TbCheck normally asks whether you want to continue or cancel when a file has been changed or when there is no checksum information available. In a business environment it may be unwise to leave such decisions to employees. Option SECURE makes it impossible to execute new or unknown programs, or programs that have been changed.

NOTE:
Be aware that the SECURE option also disables the OFF and REMOVE options.

3.5.4 Understanding the Scanning Process

This section adds to your knowledge of TbCheck by explaining a little more about the scanning process.

Whenever a program wants to execute, TbCheck steps in to see if it really has the authority to do so. During that time it displays the message "*Checking*" in the upper left hand corner of the screen. TbCheck operates at lightning speed, so the message appears only momentarily.

TbCheck quickly checks a program when the program loads. If TbCheck detects that a file has changed, a notification message appears. At this point, you can choose to either continue, or to abort the program's execution.

If there is no information in the ANTI-VIR.DAT file about the program, TbCheck also informs you of this. You can either choose to continue without checking, or to abort the program's execution.

TIP:
You can prevent users from executing unauthorized software by using the SECURE option.

3.5.5 Testing TbCheck

Understandably, many users wish to test the product they are using. In contrast to a word processor, for example, it is very difficult to test a smart integrity checker like TbCheck. You cannot change a random 25 bytes of an executable file just to find out whether TbCheck detects the file change. On the contrary, it is very likely that TbCheck will NOT detect it because the program checks only the entry area of the file, whereas the changed bytes might reside in another location within the file. But again, if a virus infects the file, this entry area will definitely change, so this is perfectly adequate to detect all infections.

3.6 Using TbClean

In case a virus infects one or more files, and you wish to remove the virus from those files (for example, in case you do not have a clean backup of the files), you can use TbClean. TbClean is the program that can remove viruses from infected files, even without knowing the virus itself. This section explores TbClean.

3.6.1 Understanding TbClean

TbClean isolates viral code in an infected program and removes it. It is then safe to use the program again, since TbClean securely eliminates the risk of other files becoming infected or damaged.

Understanding the Repair Cleaner

TbClean works differently from conventional virus cleaners because it does not actually recognize any specific virus. TbClean's disinfection scheme is unique, employing ThunderBYTE's heuristic (learn as you go) technology so that it works with almost any virus.

Actually, the TbClean program contains two cleaners: a "repair" cleaner, and a "heuristic" cleaner. The repair cleaner needs an ANTI-VIR.DAT file generated by the TbSetup program before the infection occured. This ANTI-VIR.DAT file contains essential information such as the original file size, the bytes at the beginning of the program, a cryptographic checksum to verify the results, etc. This information enables TbClean to disinfect almost every file, regardless of the specific virus that has infected it, even if it is unknown.

Understanding the Heuristic Cleaner

In the heuristic cleaning mode TbClean does not need any information about viruses either, but it has the added advantage that it does not even care about the original, uninfected state of a program. This cleaning mode is very effective if your system becomes infected with an unknown virus and you neglected to let TbSetup generate the ANTI-VIR.DAT files in time.

In the heuristic mode, TbClean loads the infected file and starts emulating the program code to find out which part of the file belongs to the original program and which belongs to the virus. The result is

successful if TbClean restores the functionality of the original program, and reduces the functionality of the virus to zero.

NOTE:
This does not imply that the cleaned file is 100% equal to the original. Please read on.

When TbClean uses heuristic cleaning to disinfect a program, the file most likely will not be exactly the same as in its original state. This does not imply a failure on TbClean s part, nor does it mean the file is still infected in some way.

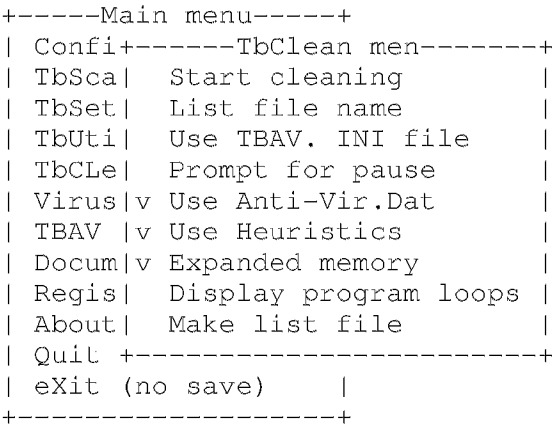
It is actually normal that the heuristically cleaned file is still larger than the original. This is normal because TbClean tries to be on the safe side and avoids removing too much. The bytes left at the end of the file are dead code, that is, instructions that will never execute again since TbClean removes the jump at the beginning of the program. If the cleaned file is an EXE type file, it is likely that some bytes in front of the program (the EXE-header) are different. There are several suitable solutions for reconstructing the EXE-header, so TbClean cannot, of course, know the original state of the program. The functionality of the cleaned file will nevertheless be the same.

NOTE:
This applies only to heuristic cleaning. If there is a suitable ANTI-VIR.DAT record available, the cleaned program will normally be exactly the same as the original clean file.

It's also possible for a virus to infect a file with multiple viruses, or multiple instances of the same virus. Some viruses keep on infecting files, and in such cases the number of infected files keeps growing. If TbClean used its heuristic cleaning mode, it is very likely that TbClean removed only one instance of the virus. In this case, it is necessary to repeat the cleaning process until TbClean reports that it cannot remove anything else.

3.6.2 Working with the TbClean Menus

Selecting TbClean from TBAV's Main Menu displays the following menu:



We'll now explore these menu options.

The "Start Cleaning" Option

After tracking one or more viruses, all you should do is select the Start cleaning option. After specifying the relevant filename, TbClean goes into action. Before beginning, however, you can select various parameters. We will explore these in the following sections.

The "List File Name" Option

By selecting this option you can specify a filename to use as a list file (see also the Make list file option below).

The "Use TBAV.INI File" Option

If you enable this option, the TbClean configuration values, saved in the TBAV.INI file, will also be valid if you run TbClean from the DOS command line. Be careful, however, since if you specify options in the TBAV.INI file, you cannot undo them on the command line. See the "Configuring TBAV" section of Chapter 1 for details about TBAV.INI.

The "Prompt For Pause" Option

This option instructs TbClean to stop disassembling information after each full screen, enabling you to examine the results.

The "Use ANTI-VIR.DAT" Option

If you turn this option off, TbClean acts as if there were no ANTI-VIR.DAT records available and therefore performs heuristic cleaning.

The "Use Heuristics" Option

If you turn this option off, TbClean does not try to apply heuristic cleaning, even when there are no ANTI-VIR.DAT records available.

The "Expanded Memory" Option

If you select this option, TbClean detects the presence of expanded memory and uses it in heuristic mode. You might want to disable EMS usage if it is too slow or if your expanded memory manager is not very stable.

The "Show Program Loops" Option

By default TbClean keeps track of looping conditions to prevent repetitive data from appearing on your screen thousands of times. If you select this option, TbClean "works out" every loop.

CAUTION:

> Using this option drastically reduces TbClean's performance speed. Also, do not combine this option with the "Make list file" option, because the list file might grow too big

The "Make List File" Option

Selecting this option instructs TbClean to generate an output file with a chronological disassembly of the virus being removed.

Maximizing TbClean

Now that you know how to use TbClean's menus, you can more easily understand the power of using it from the command line.

3.6.3 Using TbClean Command Line Options

When you run TbClean from the DOS command line, it recognizes command line options (often called "switches" in DOS terms). These options appear as "key-words" or "key-letters." The words are easier to memorize, so we will use these in this manual for convenience.

You can maximize TbClean's performance by using its command line options. The following table lists these options:

option parameter	short	explanation
help	he	display on-line help
pause	pa	enable pause prompt
mono	mo	force monochrome display output
noav	na	do not use ANTI-VIR.DAT records
noheur	nh	do not use heuristic cleaning
noems	ne	do not use expanded memory
showloop	sl	show every loop iteration (slow!)
list[=<filename>]	li	create list file

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

TIP:
Remember that you can display these options from the command line by entering TBCLEAN ?.

help (he).

Specifying this option displays the above options list.

pause (pa).

This option instructs TbClean to stop disassembling information after each full screen, enabling you to examine the results. The PAUSE option is available for registered users only.

mono (mo).

This option enhances the screen output on some LCD screens or color-emulating monochrome systems.

noav (na).

If you specify this option, TbClean acts as if there were no ANTI-VIR.DAT records available and therefore performs heuristic cleaning.

noheur (nh).

If you specify this option, TbClean does not try to apply heuristic cleaning, even when there are no ANTI-VIR.DAT records available.

noems (ne).

If you specify this option, TbClean does not detect the presence of expanded memory and use it in heuristic mode. You might want to disable EMS use if it is too slow, or if your expanded memory manager is not very stable.

showloop (sl).

By default TbClean keeps track of looping conditions to prevent repetitive data from appearing on your screen thousands of times. If you select this option, TbClean "works out" every loop.

CAUTION:
Using this option drastically reduces TbClean's performance speed. Also, do not combine this option with the "Make list file" option, because the list file might grow too big

list [=<filename>] (li).

This option instructs TbClean to generate an output file with a chronological disassembly of the virus being removed. The LIST option is available for registered users only.

Here are two examples of using TbClean from the command line:

1. This command:

```
TBCLEAN VIRUS.EXE
```

instructs TbClean to make a backup of the file VIRUS.EXE using the name filename VIRUS.VIR, and then disinfect VIRUS.EXE.

2. This command:

```
TBCLEAN VIRUS.EXE TEST.EXE
```

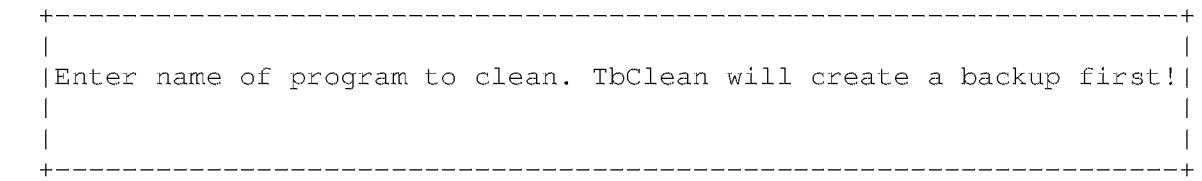
instructs TbClean to copy the file called VIRUS.EXE to the new filename TEST.EXE and then disinfect TEST.EXE.

3.6.4 Understanding the Cleaning Process

TbClean's cleaning process is extremely important. To better illustrate it, let's look at a sample file cleaning.

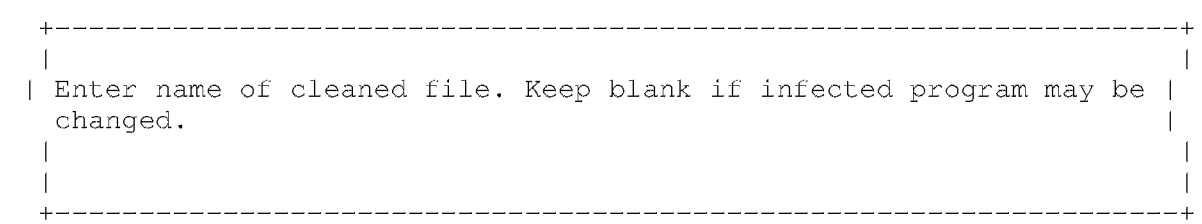
Assume you want to clean a file called COMMAND.COM, which resides in the TMP directory on drive G. To do so, you would follow these steps:

1. Select the "Start cleaning" option on the TBAV menu. The following window appears:



The ThunderBYTE utility cleans on a file-by-file approach; that is, it cleans one file, verifies the result, and continues on to the next file. This helps you keep track of which file is clean, which file is damaged and should be restored from a backup, and which file is still infected.

2. Specify the name of the file. In this case, you would type G:\TMP\COMMAND.COM and press ENTER. The following window appears:



3. Type a new file name and press ENTER. In this case, we'll use G:\TMP\TEST.EXE. TbClean now begins the cleaning process.

By specifying a different name you ensure that the cleaned file cannot overwrite the original file. In this example TbClean copies COMMAND.COM to TEST.COM and disinfects TEST.COM.

If you do not specify a backup filename, TbClean creates a backup with the .VIR extension. In this example, the TbClean would copy the original file to COMMAND.VIR and then clean COMMAND.COM.

During the cleaning process, TbClean displays as much information as possible about the current operation, as illustrated below. All the major actions appear in the emulation window at the lower half of the screen, which displays a disassembly and the register contents of the program under scrutiny, as well as a progress report. The top-left and top-right status windows reveal useful details of the infected file and (if TbClean can find a suitable ANTI-VIR.DAT file) the file's original status. You can abort the cleaning process by pressing Ctrl+Break.

```
+-----------------------------------------------------------------+
|  Thunderbyte clean utility       (C) 1992-95 Thunderbyte B.V.   |
+---------Infected state----------++---------Original state-------+
| Entry point (CS:IP)   34BF:0012 || Entry point (CS:IP) 34BF:0012|
| File length                     || File length         UNKNOWN! |
| Cryptographic CRC     9F90F52A  || Cryptographic CRC   UNKNOWN! |
+---------------------------------++--------------------------------+
|                                                                   |
| Starting clean attempt. Analyzing infected file...                |
| Anti-Vir not found: original state unknown. Trying emulation...   |
| Emulation terminated:                                             |
|                                                                   |
| G:\VIRUS\COMMAND.COM                                              |
| CS:IP    Instruction     AX  BX  CX  DX  DS  SI  ES  DI  SS  SP  |
| 9330:0101  mov ah,40    FFFE9330FFFFEFFFD382FFEDEFFEFFFF9520007E|
| 9330:0103  mov bx,0002  40FE9330FFFFEFFFD382FFEDEFFEFFFF9520007E|
| 9330:0106  mov cx,0016  40FE0002FFFFEFFFD382FFEDEFFEFFFF9520007E|
| 9330:0109  mov dx,cs    40FE00020016EFFFD382FFEDEFFEFFFF9520007E|
| 9330:010B  mov ds,dx    40FE000200169330D382FFEDEFFEFFFF9520007E|
| 9330:010D  mov dx,0117  40FE0002001693309330FFEDEFFEFFFF9520007E|
| 9330:0110  int 21       40FE0002001601179330FFEDEFFEFFFF9520007E|
| 9330:0112  mov ax,4CFF  40FE0002001601179330FFEDEFFEFFFF9520007E|
| 9330:0115  int 21       4CFF0002001601179330FFEDEFFEFFFF9520007E|
| 9330:0115  <End of emulation>                                     |
+-----------------------------------------------------------------+
```

A successful purge is not the end of the story! Your job is only partially complete. Some viruses damage data files. They could randomly change bytes on your disks, swap sectors, or perform other nasty tricks. A cleaning utility can never repair data!

4. Check your data files thoroughly and consult a virus expert to find out what the virus is capable of doing. If there is any doubt, restoring the data is definitely the most reliable option.

WARNING:

Under no circumstances should you continue to use cleaned software! Cleaning is a temporary solution that simply enables you to delay a large restore operation until a more practical time. You should never rely on a cleaned program for any length of time. This is not a criticism of anti-viral cleaning agents. If your data is valuable to you, you should care for it as much as possible, and sticking to original software is simply an elementary precaution. In other words, restore the original programs as soon as possible!

3.6.5 Understanding Cleaning Limitations

Although TbClean has a very high success rate and is able to clean programs that other cleaners refuse to process, it simply cannot remove all viruses and cannot clean every file. Examples of computer viruses that TbClean (or other virus cleaners) cannot clean include:

Overwriting viruses. This type of virus does not add itself to the end of the original program, rather it copies itself over the original file. Further, it does not attempt to start the original program but simply hangs the machine or returns you to DOS after it activates. Since it overwrites the original file, no cleaner can restore the file.

Some encrypted viruses. TbClean is usually able to decrypt the virus. However, some viruses use anti-debugger features that TbClean cannot yet cope with (but we re working on it!).

The construction of some program files makes them impossible to clean, making reinstallation the only option. Some of these file types include:

EXE-programs with internal overlays. TbScan marks these files with an "i" flag. Any infection is sure to cause major damage to these files. Some viruses recognize such programs and do not

infect them, but most viruses infect these programs anyway and corrupt them. No cleaner can repair this kind of damage.

Programs with sanity check routines. Some programs (mostly anti-virus software or copy-protected programs) perform their own kind of sanity check. Heuristic cleaning of an infected program normally results in a program that is not physically identical to the original. So, although TbClean removes the virus from the program and the program is functionally identical to the original, the program's internal sanity check usually detects the slight changes and aborts the program.

Cleaning Multiple Files

TbClean has no provisions for cleaning multiple programs in one run. There are two reasons for this omission:

1. TbClean cannot search for viruses automatically since it does not know any virus.

2. We recommend that you clean the system on a file-by-file basis. Clean one file, verify the result, and go on to the next file. Again, this helps you keep track of which files are clean, which files are damaged and should be restored from a backup, and which files are still infected.

3.7 Using TbMem

TBAV provides three extra utilities that help you build a massive security wall around your computer system. This set includes: TbMem, TbFile and TbDisk. In this section, we'll introduce these three utilities collectively as a set and then examine each individual utility.

3.7.1 Introducing the TbMem, TbFile & TbDisk Utilities

As the old saying goes, An ounce of prevention is worth a pound of cure, and the computer virus threat gives this old saying new meaning. TBAV is the best product on the market for removing viruses, but if this is all it did, it would be of little use. It's much wiser to prevent virus infection than wait until you get one and remove it.

This is where a set of three small memory-resident (TSR) programs come in. These utilities are shipped with TBAV for DOS; they monitor specific areas of your system and protect against virus infection. These three utilities are:

TbMem.

This program detects attempts by programs to remain resident in memory and ensures that no program can remain resident in memory without permission.

TbFile.

This program detects attempts by programs to infect other programs.

TbDisk.

This program detects attempts by programs to write directly to the disk (bypassing DOS), attempts to format disks, and other such destructive actions.

3.7.2 Loading TbMem, TbFile and TbDisk

The TbMem, TbFile and TbDisk programs load in the same way. The following sections contain specific information on each of the programs, but here we present loading information that is common to all of them.

CAUTION:
You must load TbDriver before you can load any of the TbMem, TbFile or TbDisk utilities. These utilities will refuse to load without it.

There are three possible ways to load TbMem, TbFile or TbDisk. Please note that we call the programs TbXXX here. Naturally, you will replace the XXX with either Mem, File, or Disk when you load each utility.

1. From the DOS prompt or within the AUTOEXEC.BAT file:

   <PATH>TBXXX

2. From the CONFIG.SYS file as a TSR (DOS 4 or higher):

   INSTALL=<PATH>TBXXX.EXE

   The INSTALL= CONFIG.SYS command is NOT available in DOS 3.xx.

3. From the CONFIG.SYS as a device driver:

   DEVICE=<PATH>TBXXX.EXE

NOTE:
Executing one of the utilities TbMem, TbFile or TbDisk as a device driver does not work in all OEM versions of DOS. If it doesn't work, use the INSTALL= command or load the desired program from within the AUTOEXEC.BAT. TbMem, TbFile and TbDisk should always work correctly after being started from within the AUTOEXEC.BAT file. Also, unlike other anti-virus products, you can load the ThunderBYTE Anti-Virus utilities before starting a network without losing the protection after the network starts.

In addition to the three loading possibilities, if you are using DOS version 5 or above, you can load the TbMem, TbFile or TbDisk programs in an available UMB (upper memory block) from AUTOEXEC.BAT using the following command:

```
LOADHIGH <PATH>TBXXX.EXE
```

You can load TbMem, TbFile or TbDisk high from within the CONFIG.SYS using the following command:

DEVICEHIGH=<PATH>TBXXX.EXE

If you are using Microsoft Windows, you should load the resident TBAV programs BEFORE starting Windows. When you do this, there is only one copy of the program in memory regardless of how many DOS windows you might open. Every DOS window (that is, every virtual machine) has a fully functional copy of the program running in it.

Each of the programs automatically detects if Windows is running, and switches itself into multitasking mode if necessary. You can even disable each of the programs in one window without affecting the functionality in another window.

3.7.3 Using Command Line Options

You can load all the TbMem, TbFile or TbDisk utilities using several command line options. See the description of each individual utility for further information.

3.7.4 Understanding TbMem

Once they execute, most viruses remain resident in memory. While resident in memory, they might have many opportunities to infect other files in the background, interfere with the system operation, hide themselves from virus scanners or checksumming programs, and/or perform other nasty tasks.

On the other hand, because so many viruses remain resident in memory, most of them are easy to detect by monitoring the process of becoming memory resident.

TbMem monitors the system and ensures that no program can remain resident in memory without permission. This brings to your attention any software that attempts to remain resident, thereby reducing the likelihood of a virus going unnoticed.

TbMem also protects CMOS (a small area of memory that stores vital information concerning your computer).

NOTE:

What exactly is a memory-resident program? Most programs run by executing a command at the DOS command line, perform some task, and then terminate, placing you back where you started. Some programs, however, continue to operate after you terminate them. These

programs load themselves into memory, remain resident in memory, and perform some task in the background. Programs in this category include: disk caches, print spoolers and network software. These programs are often referred to as TSR (Terminate and Stay Resident) programs.

Like a TSR program, most viruses also remain resident in memory, and it is for this reason that TbMem should be usedto control the process of becoming resident in memory.

If a program attempts to become resident, TbMem offers you the option to abort the attempt. It does this by guarding the DOS TSR function calls while also monitoring important interrupts and memory structures. TbMem uses the ANTI-VIR.DAT records to determine whether it will allow a specific program to remain resident in memory.

TbSetup recognizes many common TSRs. If it doesn't recognize a TSR, however, TbMem asks your permission for the TSR to load. It then maintains permission information in the ANTI-VIR.DAT files to prevent TbMem from bothering you when an approved TSR is loading.

TbMem also checks the contents of the CMOS configuration memory after each program termination to ensure that programs have not changed. TbMem offers you the option of restoring the CMOS configuration when it changes. Once you teach TbMem which programs are TSRs and which are not on a PC, you can use TbSetup to set the permission flag of these files on other machines.

TbMem also installs a hot key that you can use to escape from nearly all programs.

TbMem is fully network compatible. It does not require you to reload the checker after logging onto a network.

3.7.5 Working with TbMem

Since TbMem is a memory resident program, you can execute and configure it from the command line or from within a batch file. It is more efficient, however, to load TbMem at boot up from either CONFIG.SYS or AUTOEXEC.BAT. See the "Introducing the TbMem, TbFile and TbDisk Utilities" section earlier in this chapter for details.

CAUTION:

You must load TbDriver before you can load TbMem. TbMem will refuse to load without it.

3.7.6 Maximizing TbMem

You can maximize the performance of TbMem by using its command line options. The first four options in the table below are always available. The other options are available only if TbMem is not yet memory resident.

option parameter	short	explanation
help	?	display on-line help
remove	r	remove TbMem from memory
on	e	enable checking
off	d	disable checking
secure	s	do not execute unauthorized TSRs
hotkey<=keycode>	k	specify keyboard scancode for the program cancel hotkey
nocancel	n	do not install the cancel hotkey
nocmos	m	do not protect CMOS memory

The explanations in the above table serve as a quick reference, but the follow descriptions provide more information about each option.

TIP:
Remember that you can display these options from the command line by entering TBMEM ?.

help (?).

Specifying this option displays the brief help as shown above.

remove (r).

This option disables TbMem and attempts to remove the resident part of its code from memory and return this memory space to the system. Unfortunately, this works only if you loaded TbMem last. An attempt to remove a TSR after you load another TSR leaves a useless gap in memory and could disrupt the interrupt chain. TbMem checks whether it is safe to remove its resident code; if not, it simply disables itself.

on (e).

This option reactivates TbMem after you disable it using the OFF option.

off (d).

Specifying this option disables TbMem but leaves it in memory.

secure (s).

TbMem normally asks the user to continue or to cancel when a program tries to remain resident in memory. In some business environments, however, employees should not make this choice. If you use this option, it is no longer possible to execute new or unknown resident software. It is also no longer possible to use the REMOVE or OFF options.

hotkey (k).

TbMem offers you a reliable way to escape from any program by pressing a special key combination. You can not only use this feature to escape from programs that "hang," but also from software that seems to be malicious (although we recommend powering down and rebooting from a write-protected system disk). Instead of the default combination (Ctrl+Alt+Insert), you can specify another keyboard combination using the HOTKEY=<KEYCODE> option. You must specify the scancode using a 4-digit hexadecimal number; the first two digits specify the shift-key mask, and the last two digits specify the keyboard scancode. Consult your PC manual for a list of "scan codes." For example, the default scan code is 0C52, but you can change this to another code, such as 0C01, the code for Ctrl+Alt+Esc.

nocancel (n).

TbMem normally installs the program cancel hot key (Ctrl+Alt+Insert). If you do not want to use the program cancel hot key, specify this option, since this saves a few bytes of memory.

nocmos (m).

TbMem normally protects the CMOS memory if available. If you do not want TbMem to do this, you can specify this option.

The following command loads TbMem as a device driver in the CONFIG.SYS, configures the "program cancel hot key" as Ctrl+Alt+Esc, and cancels protection of CMOS memory:

```
DEVICE=C:\TBAV\TBMEM.EXE HOTKEY=0C01 NOCMOS
```

To achieve the same functionality, you could execute TbMem from the DOS command line rather then specifying the TbMem command line in the CONFIG.SYS by entering the following command at the DOS command line:

```
C:\TBAV\TBMEM.EXE HOTKEY=0C01 NOCMOS
```

3.7.7 Understanding TbMem's Operation

If TbMem detects that a program tries to remain resident in memory, it displays a pop-up window displaying a message to that effect. You can either choose to continue, or to abort the program's loading. If you answer "NO" to the question "Remove program from memory?" the program continues undisturbed, and TbMem places a mark in the ANTI-VIR.DAT file about this program. Next time you invoke the same resident program, TbMem will not disturb you again.

There are many programs that normally remain resident in memory, such as: disk caches, print spoolers, and others. How, then, does TbMem distinguish between these programs and viruses?

TbMem uses the ANTI-VIR.DAT records generated by TbSetup to keep track of which files are normal TSRs and which are not. It marks most common resident software as being common so you don't have to worry about these files.

If TbMem pops up with the message that a program tries to remain resident in memory, you have to consider the purpose of the program mentioned. For example, is the program supposed to continue to operate in the background? The answer is obviously yes if the program is a disk cache, print spooler, pop-up utility or system extension software.

If, on the other hand, the message appears after you have exited your word processor, database, spreadsheet application, something is definitely wrong! You ought to terminate the program immediately and use a virus scanner to check the system. The same applies when software that

operates normally without staying resident in memory suddenly changes its behavior and tries to remain resident in memory.

3.8 Using TbFile

This section concerns another resident TBAV utility, TbFile, which checks programs for virus infections as they begin to load.

3.8.1 Understanding TbFile

The two most dangerous virus categories are the boot sector and the file variants. File viruses all have a common purpose, namely, to infect programs. Infecting a program involves very unusual file manipulations that are quite dissimilar to normal file handling procedures, so in order to detect viral activity it is essential to keep an eye out for program file changes involving peculiar actions.

TbFile monitors the system and detects attempts by programs to infect other programs. Unlike other file guards, TbFile monitors the system only for virus specific file modifications. TbFile doesn't generate an alarm when a program modifies itself for configuration purposes, nor does it bother you when you update a program or create one yourself. On an average system, configurations should never cause a false alarm. TbFile has a very sophisticated infection detector and will not give a false alarm when you perform standard file operations. In normal configurations you will never get a false alarm!

TbFile not only detects attempts to infect programs, it also offers you the option of aborting the infection process and continuing a program's execution.

TbFile also detects other suspicious activities, including setting the seconds value of time stamps to an illegal value.

TIP:

> As many users know, you can protect files against unwanted modifications by means of the read-only attribute. Without TbFile, however, someone can easily circumvent this standard DOS protection. TbFile detects any attempts to sabotage the read-only attribute. This gives you added security by enabling you to use this uncomplicated method to fully protect your files against destruction and infection.

TbFile is fully network compatible. It does not require you to reload the checker after logging onto a network. In contrast, other resident anti-virus utilities force you to choose between protection BEFORE you start the network, or protection AFTER you start network, but not both.

3.8.2 Working with TbFile

Since TbFile is a memory resident program, you can execute and configure it from the command line or from within a batch file. It is more efficient, however, to load TbFile at boot up from either CONFIG.SYS or AUTOEXEC.BAT. See the "Introducing the TbMem, TbFile and TbDisk Utilities" section earlier in this chapter for details.

CAUTION:
You must load TbDriver before you can load TbFile. TbFile will refuse to load without it.

3.8.3 Maximizing TbFile

You can maximize the performance of TbFile by using its command line options. The first four options in the table below are always available. The other options are available only if TbFile is not yet memory resident.

option parameter	short	explanation
help	?	display on-line help
remove	r	remove TbFile from memory
on	e	enable checking
off	d	disable checking
secure	s	all permissions denied
allattrib	a	readonly check on all files
compat	c	allow CPM-style file I/O calls

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

TIP:
Remember that you can display these options from the command line by entering TBFILE ?.

help (?).

Specifying this option displays the brief help shown above.

remove (r).

This option disables TbFile and attempts to remove the resident part of its code from memory and return this memory space to the system. Unfortunately, this works only if you loaded TbFile last. An attempt to remove a TSR after you load another TSR leaves a useless gap in memory and could disrupt the interrupt chain. TbFile checks whether it is safe to remove its resident code; if not, it simply disables itself.

on (e).

This option reactivates TbFile after you disabled it using the OFF option.

off (d).

Specifying this options disable TbFile, but leaves it in memory.

secure (s).

TbFile normally asks you to continue or to cancel when a program tries to perform a suspicious operation. In some business environments, however, employees should not make this decision. If you use the SECURE option, it is no longer possible to allow suspicious operations. It is also no longer possible to use the OFF and REMOVE options.

allattrib (a).

TbFile normally protects only the read-only attribute of executable files (program files with the extension COM and EXE). If you want to have the read-only check on all files, add this option. In this case you always get an alarm when something attempts to remove the read-only attribute of any file.

compat (c).

DOS still contains some CPM (an earlier operating system) internal functions, even though DOS programs no longer use these functions. Some viruses, however, use these functions to bypass anti-virus software. TbFile closes these backdoors by default, but you can prevent this by specifying this option.

The following command loads as a device driver in CONFIG.SYS and it guards the read-only attribute of all files:

```
DEVICE=C:\TBAV\TBFILE.EXE ALLATTRIB
```

To achieve the same functionality, you could execute TbFile from the DOS command line rather then specifying the TbFile command line in the CONFIG.SYS by entering the following command at the command line:

```
C:\TBAV\TBFILE.EXE ALLDRIVES
```

3.9 Using TbDisk

This section deals with TbDisk, which prevents viruses from damaging data on your hard disk.

3.9.1 Understanding TbDisk

Many viruses try to damage the data on disk. They accomplish this by various actions, such as, formatting the disk, overwriting the FAT, and swapping disk sectors, among others. Almost anything is possible!

Another category of malicious software, known as boot sector virus droppers, install a boot sector virus on the disk. The program itself is not a virus, so detection with virus scanners and other anti-viral software is very difficult. The only way to detect such a program is by monitoring its behavior.

The main problem in all this lies in the way these programs manage to avoid the usual DOS procedures: they go directly to the BIOS (Basic Input/Output System). This is the reason you need TbDisk, to monitor the system and to ensure that no program can write directly to disk without permission. TbDisk draws attention to any software that attempts to write directly to disk, thereby reducing the likelihood of a virus remaining unnoticed. TbDisk prevents viruses from damaging data on your disk and stops boot sector virus droppers in their tracks.

TbDisk not only informs you when a program tries to write directly to the disk, it also offers you the option to abort the program before it can cause any damage.

TbDisk is able to detect stealth techniques, that is, attempts to single step through the BIOS software, and even monitors the use of undocumented calls that could cause disk damage. For example, TbDisk is able to distinguish whether DOS or an application makes direct write attempts via Int 13h (a system call implemented in the BIOS of your computer). Direct writes are perfectly legal for DOS, but unusual for application software.

TbDisk does require a little maintenance. TbDisk uses the ANTI-VIR.DAT records to determine if it should allow a program (including popular disk utilities, which TbSetup recognizes) to write directly to the disk. In the absence of an ANTI-VIR.DAT record, TbDisk asks your permission first and, if granted it, updates the record accordingly to avoid repeated warnings about the same program.

TbDisk is fully network compatible. It does not require you to reload the program after logging onto a network. Other resident anti-virus utilities force you to choose between either protection BEFORE the network is started, or protection AFTER it starts, but not both..

TIP:
TbDisk also comes in handy if you ever need to write protect a hard disk. This bonus feature often helps when testing new software.

3.9.2 Working with TbDisk

Since TbDisk is a memory resident program, you can execute and configure it from the command line or from within a batch file. It is more efficient, however, to load TbFile at boot up from either CONFIG.SYS or AUTOEXEC.BAT. See the "Introducing the TbMem, TbFile and TbDisk Utilities" section earlier in this chapter for details.

CAUTION:
You must load TbDriver before you can load TbDisk. TbDisk will refuse to load without it.

In addition to all this, there are several special considerations in using TbDisk.

Loading TbDisk

Improper installation of TbDisk can cause excessive false alarms! If you want to install TbDisk in your CONFIG.SYS or AUTOEXEC.BAT file, we recommend that you use the INSTALL option of TbDisk first. If the system continues to behave normally and TbDisk does not give false alarms when you copy files on your hard disk, TbDisk is installed correctly and you can remove the INSTALL option from the command.

WARNING:
Failure to use the Install option when you install TbDisk in CONFIG.SYS or AUTOEXEC.BAT file might cause loss of data! Please read on.

While the INSTALL option instructs TbDisk to allow all disk accesses, it also displays a message as it would do in normal mode. If no false alarms occur when you copy files on your hard disk, TbDisk is installed correctly and you can remove the INSTALL option.

If TbDisk causes false alarms, load TbDisk further ahead in your CONFIG.SYS or AUTOEXEC.BAT file until it works as it should.

CAUTION:
Unlike the other TBAV utilities, we recommend that you load TbDisk after other resident software! Failure to do so can cause false alarms!

TbDisk detects if Windows is running and automatically switches into multitasking mode if necessary. You can even disable TbDisk in one window without affecting the functionality in another. If you are using Windows fast 32-bit disk access, you might need to use TbDisk's WIN32 option if Windows displays an error-message.

3.9.3 Maximizing TbDisk

You can maximize TbDisk's performance by using its command line options. The first four options are always available. The other options are available only if TbDisk is not yet memory resident.

option parameter	short	explanation
help	?	display on-line help
remove	r	remove TbDisk from memory
on	e	enable checking
off	d	disable checking
wrprot	p	makes hard disk write protected
nowrprot	n	allow writes to hard disk
win32	w	allow Windows 32-bit disk access
secure	s	deny access without asking first
notunnel	t	do not detect tunneling
nostealth	a	do not detect stealth disk access
install	i	installation test mode

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

TIP:
Remember that you can display these options from the command line by entering TBDISK ?.

help (?).

Specifying this option displays the brief help as shown above. After loading TbDisk into memory, not all options appear.

remove (r).

This option disables TbDisk and attempts to remove the resident part of its code from memory and return this memory space to the system. Unfortunately, this works only if you loaded TbDisk last. An attempt to remove a TSR after you load another TSR leaves a useless gap in memory and could disrupt the interrupt chain. TbDisk checks whether it is safe to remove its resident code; if not, it simply disables itself.

on (e).

This option activates TbDisk after you disabled it using the OFF option.

off (d).

Specifying this option disables TbDisk but leaves it in memory.

wrprot (p).

Hard disks are more difficult to protect against writing than floppies, which adds considerable risk when doing such things as testing new software. Sometimes you might want to find out what this software does to your hard disk and how this could possibly affect your valuable data. Using the "WRPROT" option makes this safer to do. Whenever a program wishes to write to a protected disk, you will see a message such as:

```
Write protect error writing drive C: A)bort, R)etry, I)gnore?
```

You can then take the appropriate action.

CAUTION:

Software write protection is not absolutely reliable. Some viruses can bypass this protection, but fortunately they are few and far between. Despite its shortcomings, this option can be a valuable shield against most malicious software.

nowrprot (n).

Use this option to undo the WRPROT option.

win32 (w).

Windows 386 Enhanced Mode uses some undocumented DOS calls to retrieve the original BIOS disk handler when you enable 32-bit disk access. Since TbDisk guards these calls, 32-bit disk access will no longer be possible, unless you specify the WIN32 option when you initialize TbDisk.

CAUTION:
Use this option only in Windows 386 Enhanced Mode with fast 32-bit disk access enabled as it reduces anti-viral security to some extent.

secure (s).

TbDisk normally asks whether the user wants to continue or cancel when a program tries to perform direct disk access. In some business environments, however, employees should not make this decision. This option disables direct disk access permission to new or unknown software. It also disables the OFF and REMOVE options.

notunnel (t).

"Tunneling" is a technique viruses apply to determine the location of the DOS system code in memory, and to use that address to communicate with DOS directly. This inactivates all TSR programs, including resident anti-virus software. TbDisk is able to detect these "tunneling" attempts, and informs you about it. Some other anti-virus products also rely on tunneling techniques to bypass resident viruses, thereby causing false alarms. If you are currently executing other anti-viral products, the NOTUNNEL option disables TbDisk's tunneling detection.

nostealth (a).

TbDisk tries to detect direct calls into the BIOS. If such an attempt occurs, TbDisk pops up with a message that something is

accessing the disk in an unusual way. If this feature causes false alarms, you can use this option to turn it off.

install (i).

Incorrect installation can result in a large number of false alarms. You should use this option when installing TbDisk because it reduces the risk of canceling a valid disk write operation as a result of false alarms.

3.9.4 Understanding TbDisk's Operation

What is Direct Disk Access? Programs usually access files through the operating system (DOS). Whenever a program wants to update a file, for example, it asks DOS to write the data to disk. It is also possible, however, to write to a disk without using DOS. This is called direct disk access.

While normal programs do not write to the disk directly, there are some programs that need to do so, including:

Format utilities. Direct disk access is the only way to format a disk.

Disk diagnosis utilities (such as the Norton Disk Doctor, and DOS's CHKDSK command and ScanDisk utility).

Disk optimizers and defragmenters (such as Norton SpeedDisk and DOS's Defrag utility).

Since many viruses can perform direct disk access, it is essential to control this. TbDisk can distinguish between legitimate programs and a virus with the help of the ANTI-VIR.DAT records, which you can generate using TbSetup.

Whenever TbDisk pops up a message that says a program accesses to the disk directly, consider its purpose carefully. While it is perfectly acceptable for a format utility or a disk optimizer to format or edit disk sectors, this is not acceptable for a word processor or database. When TbDisk warns you that a spreadsheet or some other normal program is about to format a sector, you can be sure that something is wrong. Terminate the program pronto! Then check things out with a virus scanner before the worst happens.

3.10 Using TbUtil

This section describes TbUtil, which is designed primarily to make a precautionary backup of clean partition tables and boot sectors.

3.10.1 Understanding and using TbUtil

TbUtil provides a defense against partition table and boot sector viruses. TbUtil can be used to:

Copy the partition table, boot sector and CMOS data area into a file. You can use TbUtil on a regular basis to compare both the current and the original versions of the partition table, boot sector and CMOS data area. After an accident virus, (virus or otherwise), you can restore the copy using the TbUtil program.

Remove a partition table virus without having to low-level format the hard disk, even if there is no backup of the partition table.

Remove boot sector viruses and creates a partition table that has some first-line virus defenses built-in.

Replace the infected or clean boot sector with a safe TBAV boot sector.

NOTE:

What is a partition table? A physical hard disk might consist of more than one "partition" (or division). Each partition is a logical disk drive and has it own ID, such as C:, D:, and E:. The partition table, then, contains the disk lay-out and the starting and ending cylinder of every partition. The partition table also contains information about the operating system of a partition and which partition should be used to boot. The partition table (also called the Master Boot Record, or MBR) always resides at the very first sector of the hard disk.

Unlike most file viruses, partition table viruses are hard to remove. The only solution is to low-level format the hard disk and to make a new partition table, or to make use of scantily documented DOS commands.

TbUtil, however, makes a backup of the partition table and boot sector and uses this backup to compare and restore both the original partition table and boot sector once they become infected. You no longer have to

format your disk to get rid of a partition table or boot sector virus. The program can also restore the CMOS configuration.

Optionally, TbUtil replaces the partition table code with an immunized partition table containing facilities against viruses. The TbUtil partition code executes before the boot sector gains control, so it is able to check the boot sector in a clean environment. Once the boot sector executes, it is difficult to check it because the virus is already resident in memory and can deceive a protection scheme. Instead of booting from a clean DOS diskette just to inspect the boot sector, the TbUtil partition code performs a CRC calculation on the boot sector just before passing control to it.

If TbUtil detects a change in the boot sector, the TbUtil partition code warns you about it. The TbUtil partition code also checks the RAM layout and informs you when it changes. TbUtil does all of this every time you boot from your hard disk.

TbUtil can replace infected and clean diskette boot sectors with a new and specialized boot sector, which has several advantages over the standard boot sector:

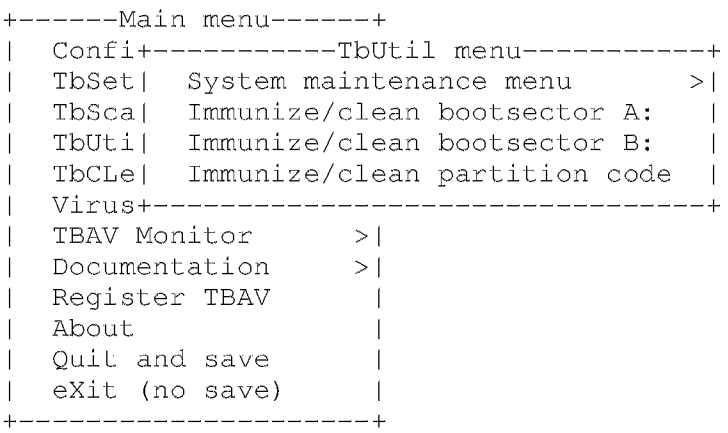
It has boot sector virus detection capabilities.

It performs a sanity check.

It offers you the possibility to redirect the boot process to the hard disk without opening the diskette drive door.

3.10.2 Working with the TbUtil Menu

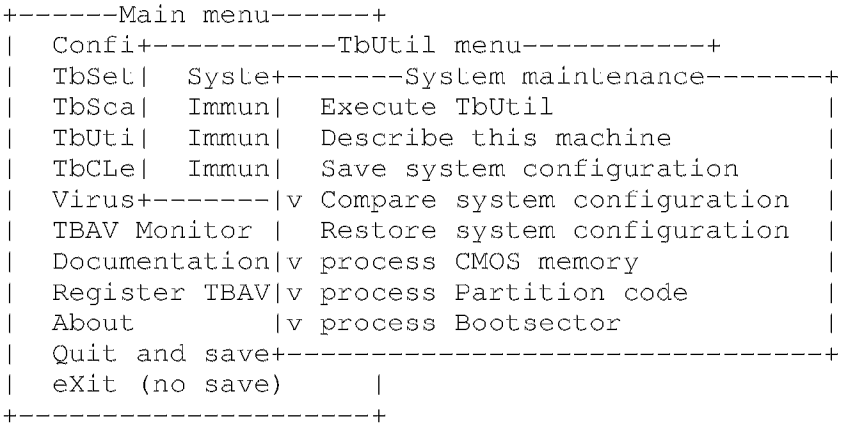
The TbUtil module contains several programs, which you can execute from either the TbUtil Menu or, in case of an emergency, from a TbUtil recovery diskette using the DOS command line. The menu, however, offers some additional menu options. Selecting the "TbUtil" option from the TBAV Main Menu displays the following menu:



We'll now explore these menu options.

The "System Maintenance Menu" Option

Selecting the "System maintenance menu" option displays the System Maintenance menu:



This menu contains the actual TbUtil program. The program takes care of saving, restoring or comparing the system configuration of your PC. It stores the backup system configuration on a diskette in a file with either a default name or a name you can specify yourself.

WARNING:
You can only restore a system configuration data file on the machine that created the data file. Restoring a configuration file from one PC to another makes the PC inaccessible!

The "System Maintenance Menu" contains the following items:

Execute TbUtil.

Before activating this option, you must select one of the optional functions: Save, Compare, or Restore the system configuration. Move to the desired option you want to activate and press ENTER. A check mark indicates that an option is active.

Describe this machine.

Enter a meaningful description of the machine. Enter something like, "486DX4 @ 100MHz, 32Mb, 2 Gb SCSI disk, room 12, Mr. Smith." You do NOT have to remember this description; TbUtil displays it on the screen when comparing or restoring, which helps you to verify that the data file belongs to the machine.

Save system configuration.

This option stores the partition table, boot sector and CMOS data area into the TbUtil data file.

WARNING:

Since the PC is completely inaccessible to DOS if the partition table becomes damaged, we RECOMMEND that you store both the TbUtil data file AND the program TBUTIL.EXE itself on a "rescue" diskette! If the partition table is damaged or destroyed, then the only solution to the problem may reside on the "rescue" diskette, since your hard drive may be inaccessible!

When loading TbUtil from the command line you must specify a filename after the STORE option. In contrast, using the TBAV menu, you can use the default filename TBUTIL.DAT. If you own more than one PC, we recommend that you create one TbUtil diskette with all TbUtil data files of all your PC's on it. Use the extension of the file for PC identification, as in the following:

A:TBUTIL.<NUMBER>

Compare system configuration.

This option enables you to check on a regular basis that everything is still okay. If you specify this option, TbUtil compares the

information in the TbUtil data file against the partition table, boot sector, and CMOS data areas. It also displays the comment stored in the data file. Using this option also guarantees that the TbUtil data file is still readable.

Restore system configuration.

This option enables you to restore the partition table, boot sector, and CMOS data area. It asks you to confirm that the data file belongs to the current machine. Finally, it can restore the partition table, boot sector of the partition to be used to boot, and the CMOS data area.

Process CMOS memory,
Process Partition code, and
Process Boot sector.

By default, TbUtil restores the partition code, boot sector, and CMOS if you specify the "Restore system configuration" option. If you use one of the above options in combination with the "Restore option," TbUtil restores only the items you specify.

The "Immunize/Clean Boot sector A: [or] B:" Options

You can use these options to clean diskettes infected by a boot sector virus or to replace the standard boot sector with a boot sector that has advantages over the original one:

The TBAV boot sector has virus detection capabilities. The TBAV boot sector checks that it resides on the correct place on the diskette, and that Int 13h and/or Int 40h still exist in system ROM. This makes it possible to detect even stealth and boot sector viruses.

The TBAV boot sector can load the system files if they are available on the disk, but if the DOS system files are not on the disk, the TBAV boot sector displays a small menu offering you two possibilities: retry the boot operation with another diskette, or boot from the hard disk. If you select the latter, you don't have to open the diskette drive door.

The "Immunize/Clean Partition Code" Option

This is an extremely powerful option, which you can use to clean an infected partition table if there is no TbUtil data file. It saves the original partition code in a file and replaces the existing partition table code with a new partition routine that contains some virus detection capabilities. You must execute TbUtil from a floppy drive or you have to specify the name of the file (the specified drive should be a diskette drive) to store the original partition code.

If the original partition table becomes irreparably damaged and can't be used to build a new one, TbUtil scans the entire disk for information about the original disk layout. TbUtil also searches for TbUtil data files on the hard disk.

CAUTION:

> While it is a good idea to keep a copy of the data file on the hard disk, we recommend that you store the data file on a diskette. Just in case!

If your system configuration changes, that is, you update your DOS version or change the amount of memory, you need to update the information stored in the immune partition as well. You can do this by using this option.

In the unlikely event that the system does not boot properly, you can restore the original partition table using the TbUtil RESTORE option (refer to The "System Maintenance Menu Option" section above) or by using the DOS version 5 or above FDISK /MBR command (which creates a new partition table).

TIP:

> If you have installed two hard drives in your computer, you can immunize the partition code of the second hard drive by specifying the physical drive number rather than the drive ID (i.e., execute the command TbUtil 2:)

If the new partition code works properly, you should make a backup copy of it on a diskette using the TbUtil STORE option (refer to The "System Maintenance Menu Option" section above).

3.10.3 Maximizing TbUtil

This section describes how to fully maximize TbUtil in three ways: use command line option, use the anti-virus partition, use the TbUtil diskette.

Now that you know how to use TbUtil's menus, you can more easily understand how to maximize its performance by using its command line options.

option parameter	short	explanation
immunize <drive>	im	Immunize/Clean boot sector or MBR of <drive>
getboot <drive>	gb	Save boot sector/MBR into file
store [<filename>]	st	Store system information
restore [<filename>]	re	Restore system information
compare [<filename>]	co	Compare system information

Sub-options of immunize option:

norepeat	nr	Do not ask for next diskette
nomem	nm	Do not check for amount of RAM
batch	ba	Do not prompt to insert a disk

Sub-options of store option:

description=<descr.>	de	Add description to data file

Sub-options of restore option:

part	pt	Restore partition table
boot	bo	Restore boot sector of hard disk
cmos	cm	Restore CMOS data memory

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

Immunize <floppy drive> (im).

You can use this option to clean diskettes infected by a boot sector virus or to replace the standard boot sector by a boot sector that has advantages over the original one:

The TBAV boot sector has virus detection capabilities. The boot sector checks that it still resides on the correct place on the diskette, and that Int 13h and/or Int 40h still exist in system ROM. This makes it possible to detect even stealth and boot sector viruses.

The TBAV boot sector is able to load the system files if they are available on the disk, but if the DOS system files are not on the disk, the TBAV boot sector displays a small menu offering you two possibilities: retry the boot operation with another diskette, or boot from the hard disk. If you select the latter, you don't have to open the diskette drive door.

Immunize c: (im c:).

This is an extremely powerful option, which you can use to clean an infected partition table if there is no TbUtil data file. It saves the original partition code in a file and replaces the existing partition table code with a new partition routine that contains some virus detection capabilities. You have to execute TbUtil from a floppy drive or you have to specify the name of the file (the specified drive should be a diskette drive) to store the original partition code.

TIP:

If you have installed two hard drives in your computer, you can immunize the partition code of the second hard drive by specifying the physical drive number rather than the drive ID (i.e., execute the command TbUtil 2:)

If the original partition table becomes irreparably damaged and consequently can't be used to build a new one, TbUtil scans the entire disk for information about the original disk layout. TbUtil also searches for TbUtil data files on the hard disk.

CAUTION:

While it is a good idea to keep a copy of the data file on the hard disk, we recommend that you store the data file on a diskette. Just in case!

If your system configuration changes, that is, you update your DOS version or change the amount of memory, you need to update the information stored in the immune partition as well. You can do this by using this option.

In the unlikely event that the system does not boot properly, you can restore the original partition table using the TbUtil RESTORE option (refer to The "System Maintenance Menu Option" section above) or by using the DOS version 5 or above FDISK /MBR command (which creates a new partition table).

getboot <drive> (gb).

With this option you can copy the boot sector of the specified drive into a file.

store [<filename>] (st).

This option stores the partition table, boot sector and CMOS data area into the TbUtil data file.

WARNING:
Since the PC is completely inaccessible to DOS if the partition table becomes damaged, we RECOMMEND that you store both the TbUtil data file AND the program TBUTIL.EXE itself on a rescue diskette! If the partition table is damaged or destroyed, then the only solution to the problem may reside on the "rescue" diskette, since your hard drive may be inaccessible!

When loading TbUtil from the command line you must specify a filename after the STORE option. In contrast, using the TBAV menu, you can use the default filename TBUTIL.DAT. If you own more than one PC, we recommend that you create one TbUtil diskette with all TbUtil data files of all your PC's on it. Use the extension of the file for PC identification, as in the following:

A:TBUTIL.<NUMBER>

restore [<filename>] (re).

This option enables you to restore the partition table, boot sector, and CMOS data area. It asks you to confirm that the data file belongs to the current machine. Finally, it restores the partition table, boot sector of the partition to be used to boot, and the CMOS data area.

compare [<filename>] (co).

This option enables you to check on a regular basis that everything is still okay. If you specify this option, TbUtil compares the information in the TbUtil data file against the partition table, boot sector, and CMOS data area. It also displays the comments stored in the data file. Using this option guarantees that the TbUtil data file is still readable.

norepeat (nr).

By default, TbUtil prompts you for the next diskette after you have immunized a diskette. This option disables this function.

nomem (nm).

If you specify this option when you are immunizing your partition code, the partition code skips the RAM check while booting. This is necessary for some systems that change the memory setup during the boot process.

batch (ba).

If you specify this option, TbUtil will assume a disk has already been inserted in your disk drive. This option is particularly useful with batch files.

description =<descr.> (de).

For <desc.> enter a meaningful description of the machine. Enter something like, "486DX4 @ 100MHz, 32 Mb, 2 Gb SCSI disk, room 12, Mr. Smith." You do NOT have to remember this description; TbUtil displays it on the screen when comparing or restoring, which helps you to verify that the data file belongs to the machine.

part (pt) ,
boot (bo), and
cmos (cm).

By default, TbUtil restores the partition code, boot sector, and CMOS if you specify the RESTORE option. If you use one of these options in combination with the RESTORE option, however, TbUtil restores only the items you specify.

In the following two examples TbUtil simply store system information gathered from the partition table and boot sectors of your fixed disk(s) and the CMOS data area into a file in the current directory called TBUTIL.DAT.

```
TBUTIL STORE
TBUTIL ST
```

The following example does the same as the previous, except that TbUtil stores the information on a diskette instead of in the current directory.

```
TBUTIL STORE A:TBUTIL.DAT
```

It's a good idea to describe the machine from which you are saving information about the partition table, boot sectors and CMOS data. You can use the DESCRIPTION option to add a small, single-line description of the machine:

```
TBUTIL STORE A:TBUTIL.DAT DESCRIPTION = "TEST MACHINE"
```

You can always fall back on the information TbUtil stores if you suspect an infection by a boot sector virus. Suppose the information gathered earlier by TbUtil is stored in the file A:\TBUTIL.DAT. To compare the current system information with the information stored in the TbUtil data file, you could use this command:

```
TBUTIL COMPARE A:TBUTIL.DAT
```

Now suppose that TbUtil informs you that the current system information (that is, the partition table and the CMOS data area) does not match the information stored earlier. If you did not change the configuration of your computer, it is most likely that a virus is guilty of the change. You could restore the old system information using this command:

```
TBUTIL RESTORE A:TBUTIL.DAT PART CMOS
```

In case of a boot sector virus infection, we recommend that you disinfect (clean) all diskettes. Using the following command, TbUtil cleans and immunizes the boot sector of the diskette in drive A: and then repeats the action after asking you to insert other (possibly) infected diskettes into the disk drive:

```
TBUTIL IMMUNIZE A:
```

In case of a virus infection you should always make certain that the Master Boot Record of your fixed disk is not infected. The following command specifies an extra option, which you must use in case your computer changes its memory setup during the boot process:

```
TBUTIL IMMUNIZE C: NOMEM
```

You can easily view the contents of a TBUTIL.DAT by using the DOS TYPE command:

```
TYPE A:TBUTIL.DAT
```

3.10.4 Using the Anti-Virus Partition

If you install the ThunderBYTE partition code (by using TbUtil's IMMUNIZE option), you will see the following when booting a clean system:

```
Thunderbyte anti-virus partition (C)1993-95 Thunderbyte BV.

Checking boot sector CRC -> OK!
Checking available RAM -> OK!
Checking INT 13h -> OK!
```

In contrast, if there is a virus in the boot sector or partition table, you will see this message:

```
Thunderbyte anti-virus partition (C)1993-95 Thunderbyte BV.

Checking boot sector CRC -> OK!
Checking available RAM -> Failed!

System might be infected. Continue? (N/Y)
```

Other messages that might appear are:

"No system." This message means that there is no active partition on the disk.

"Disk error." The meaning of this message is obvious.

3.10.5 Using the TbUtil diskette

To use the TbUtil diskette, follow these steps:

1. Take a new diskette and format it as a bootable diskette (by using the DOS FORMAT /S command).

2. Copy the TbUtil files onto the diskette using this command:

```
COPY TBUTIL.* A:
```

The TbUtil files you need are TBUTIL.EXE and TBUTIL.LNG.

3. In case of an emergency (such as a damaged or infected partition table, for example), boot from the TbUtil diskette.

4. Run the TbUtil program, using the IMMUNIZE option:

```
A:\TBUTIL IMMUNIZE C:
```

This cleans the partition table.

5. You should now be able to boot from your hard disk normally.

3.11 Using TbLog

This section describes TbLog, which is designed primarily to create log files in response to various TBAV alert messages.

3.11.1 Understanding and using TbLog

TbLog is a memory resident TBAV utility that writes a record into a log file whenever one of the resident TBAV utilities pops up with an alert message. It also records when a virus is detected.

This utility is primarily for network users. If all workstations have TbLog installed and configured to maintain the same log file, the supervisor can easily keep track of what's going on. When a virus enters the network he is able to determine which machine introduced the virus, and he can take action in time.

A TbLog record provides three pieces of information:

The time stamp of when the event took place.

The name of the machine on which the event occurred.

An informative message about what happened and which files were involved.

This information is very comprehensive and takes only one line.

3.11.2 Working with TbLog

Since TbLog is a memory resident program, you can execute and configure it from the DOS command line or from within a batch file. You should, however, load TbLog automatically and when the computer boots, preferably during the execution of AUTOEXEC.BAT, or better yet, CONFIG.SYS.

You should install TbLog on every workstation. If you want to use all workstations to maintain the same log file, we recommend that you load TbLog after starting the network.

By default, TbLog maintains a log file with the name TBLOG.LOG in the TBAV directory. If you want to use another filename or another disk and/or directory, you can specify a filename (and path) on the TbLog

command line. In a network environment, we recommend that you put the log file on a server disk.

CAUTION:
Be sure to load TbDriver before trying to load TbLog. TbLog will refuse to load without it.

There are three possible ways to load TbLog:

1.From the DOS prompt or within the AUTOEXEC.BAT file:

<PATH>TBLOG

2.From CONFIG.SYS as a TSR (DOS 4 or above):

INSTALL=<PATH>TBLOG.EXE

The INSTALL= CONFIG.SYS command is NOT available in DOS 3.xx.

3.From CONFIG.SYS as a device driver:

DEVICE=<PATH>TBLOG.EXE

NOTE:
Executing TbLog as a device driver does not work in all OEM versions of DOS. If you encounter problems, use the INSTALL= command or make sure to load TbLog from the AUTOEXEC.BAT. Also, unlike other anti-virus products, you can load the ThunderBYTE Anti-Virus utilities before starting a network without losing the protection after the network is started.

In addition to the three loading possibilities, if you are using DOS version 5 or above, you can load TbLog into an available UMB (upper memory block) from AUTOEXEC.BAT using this command:

LOADHIGH <PATH>TBLOG

You can also load TbLog into high memory from within the CONFIG.SYS using this command:

DEVICEHIGH=<PATH>TBLOG.EXE

If you are using Microsoft Windows, you should load TbLog BEFORE starting Windows. When you do this, there is only one copy of TbLog in memory regardless of how many DOS windows you might open. Every DOS window (that

is, every virtual machine) has a fully functional copy of TbLog running in it.

TbLog automatically detects if Windows is running, and switches itself into multi-tasking mode if necessary. You can even disable TbLog in one window without affecting its functionality in another window.

3.11.3 Maximizing TbLog

You can maximize TbLog's performance by using its command line options. The first five options in the following table are always available. The other options are available only if TbLog is not yet memory resident.

option parameter	short	explanation
help	?	Display some on-line help
remove	r	Remove TbLog from memory
on	e	Enable TbLog
off	d	Disable TbLog
test	t	Log test message
machine=<descr.>	m	Description/name of your machine
secure	s	Do not allow removal of TbLog

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

help (?).

Specifying this option displays the brief help as shown above.

remove (r).

This option disables TbLog and attempts to remove the resident part of its code from memory and return this memory space back to the system. Unfortunately, this works only if you loaded TbLog last. An attempt to remove a TSR after you load another TSR leaves a useless gap in memory and could disrupt the interrupt chain. TbLog checks whether it is safe to remove its resident code; if not, it simply disables itself.

on (e).

This option reactivates TbLog after you disabled it using the OFF option.

off (d).

Specifying this option disables TbLog but leaves it in memory.

test (t).

Use this option to record a test message. If you use this option at the initial loading of TbLog, it records the time and machine name into the log file. If you use this option after the initial loading, it simply places a test message into the log file.

machine (m).

Using this option, you can specify the name of the machine on which TbLog is running. This machine name appears in the log file. By default, TbLog uses the network machine name on NetBios compatible machines. On other networks, such as Novell, you must enter the network name on the TbLog command line.

secure (s).

If you specify this option, it is not possible to use the OFF and REMOVE options.

The following command loads TbLog, disables, the OFF and REMOVE options, specifies that the logfile reside in directory F:\SECURITY, and identifies the machine as DESK3:

```
C:\TBAV\TBLOG F:\SECURITY\TBLOG.LOG SECURE MACHINE=DESK3
```

The following CONFIG.SYS command loads TbLog, creates the logfile in directory X:\LOGS, and specifies that the first line of the log file contain a date/time stamp and the name of the computer:

```
DEVICE=C:\TBAV\TBLOG X:\LOGS\TBLOG.LOG MACHINE=JOHN TEST
```

3.12 Using TbNet

TBAV for DOS can cooperate with TBAV for Networks, another ThunderBYTE product, via the program called TbNet. If you do not want to use the combination of TBAV for DOS and TBAV for Networks, you can skip this section.

NOTE:
For more information about TBAV for Networks, please refer its documentation. If you did not purchase TBAV for Networks yet, your local dealer can inform you about this product.

3.12.1 Understanding TbNet

TbNet is a memory resident TBAV utility that implements the communication between TBAV for DOS and TBAV for Networks. TBAV for Networks has several options for controlling remote workstations. For Windows workstations, TBAV for Windows contains all logic needed to implement the communication between the workstation and TBAV for Networks. For DOS workstations you need TbNet for this communication.

3.12.2 Working with TbNet

Since TbNet is a memory resident program, you can execute and configure it from the DOS command line or from within a batch file. You should, however, load TbNet automatically when the computer boots, preferably during the execution of AUTOEXEC.BAT, or better yet, CONFIG.SYS.

You should install TbNet on every workstation.

CAUTION:
Since TbNet uses a public network directory for its communication with TBAV for Networks, you must load TbNet after starting the network.

There are three possible ways to load TbNet:

1. From the DOS prompt or within the AUTOEXEC.BAT file:

   <PATH>TBNET

2. From CONFIG.SYS as a TSR (DOS 4 or above):

```
INSTALL=<PATH>TBNET.EXE
```

The INSTALL= CONFIG.SYS command is NOT available in DOS 3.xx.

3. From CONFIG.SYS as a device driver:

```
DEVICE=<PATH>TBNET.EXE
```

NOTE:
Executing TbNet as a device driver does not work in all OEM versions of DOS. If it doesn't work, use the INSTALL= command or load TbNet from AUTOEXEC.BAT. TbNet should always work correctly if you load it from AUTOEXEC.BAT. Also, unlike other anti-virus products, you can load the ThunderBYTE Anti-Virus utilities before starting a network without losing the protection after the network is started.

In addition to the three loading possibilities, if you are using DOS version 5 or above, you can load TbNet into an available UMB (upper memory block) from AUTOEXEC.BAT using this command:

```
LOADHIGH <PATH>TBNET
```

You can also load TbNet into high memory from within the CONFIG.SYS using this command:

```
DEVICEHIGH=<PATH>TBNET.EXE
```

We recommend that you do not use TbNet if you use MS-Windows, but use TBAV for Windows instead. TBAV for Windows has built-in functionality for communication with TBAV for Networks.

If you do want to use TbNet with MS-Windows for some reason, you should load TbNet BEFORE starting Windows. When you do this, there is only one copy of TbNet in memory regardless of how many DOS windows you might open. Every DOS window (that is, every "virtual machine") has a fully functional copy of TbNet running in it.

TbNet automatically detects if Windows is running, and switches itself into multi-tasking mode if necessary. You can even disable TbNet in one window without affecting the functionality in another window.

3.12.3 Maximizing TbNet

You can maximize TbNet's performance by using its command line options. The help and remove options in the following table are always available. The other options are available only if TbNet is not yet memory resident.

option parameter	short	explanation
help	?	Display some on-line help
remove	r	Remove TbNet from memory
netname=<netname>	n	Netname of the workstation
commdir=<path>	c	Communication directory used by workstation
frequency=<seconds>	f	Poll frequency (default is 30 seconds)
buffers=<number>	b	Number of disk buffers (default is 2)

The explanations in the above table serve as a quick reference, but the following descriptions provide more information about each option.

help (?).

Specifying this option displays the brief help as shown above.

remove (r).

This option disables TbNet and attempts to remove the resident part of its code from memory and return this memory space back to the system. Unfortunately, this works only if you loaded TbNet last. An attempt to remove a TSR after you load another TSR leaves a useless gap in memory and could disrupt the interrupt chain. TbNet checks whether it is safe to remove its resident code; if not, it simply disables itself

netname (n).

TBAV for Networks distinguishes workstations by their unique netnames. These netnames are assigned by TBAV for Networks; the agents software running at the workstations (i.e., TbNet or TBAV for Windows) receive this netname upon registering the workstation with TBAV for Networks. You need to specify this netname for correct behavior of TbNet.

commdir (c).

The communication between TBAV for Networks and the agent software running at the workstations (i.e., TbNet or TBAV for Windows) takes

place via a special "communication directory," a directory that is public to all users. You must specify the path of this directory when loading TbNet.

frequency (f).

TbNet checks the communication directory every once in a while, to see if messages originating from TBAV for Networks need to be processed. You can change the default period of 30 seconds by specifying the FREQUENCY option.

buffers (b).

TbNet internally needs some buffers to speed up the communication with TBAV for Networks. The number of these disk buffers used by TbNet can be changed by using the BUFFERS option.

The following command loads TbNet, for workstation 001AE3, making use of the communication directory J:\TBAVNW.NET.

```
C:\TBAV\TBNET NETNAME=001AE3COMMDIR=J:\TBAVNW.NET
```

4 Understanding Advanced User Information

This chapter presents some advanced information on using memory, TbSetup, TbScan, and TbClean. It also introduces you to another TBAV utility, TbGenSig, signature file compiler. While some of this material is simply for a better understanding of the utilities and might not be of interest to you, we recommend that you at least look at the first section on memory considerations.

4.1 Understanding Memory Considerations

This section presents the memory requirements for each of the TBAV utilities and how you can reduce the requirements of each utility.

4.1.1 Understanding Memory Requirements

The following table lists the memory requirements for each of the TBAV utilities:

TBAV Utility	Memory needed to load	Memory consumed after exiting
TbScan *	200 Kb	-
TbScanX **	10 Kb	800 bytes
TbCheck	4 Kb	600 bytes
TbUtil	64 Kb	-
TbClean ***	96 Kb	-
TbMem	4 Kb	600 bytes
TbFile	5 Kb	1 Kb
TbDisk	4 Kb	800 bytes
TbDriver	5 Kb	3 Kb
TbLog	5 Kb	1 Kb

* If you decide to use a log file, TbScan requires an additional 16 kilobytes of memory for the log file buffer. If TbScan uses its own built-in file system, it uses additional memory to keep the FAT in memory. Note that the memory requirements are independent of the number of signatures. The current memory requirements are adequate to manage at least 2500 signatures.

** The amount of memory TbScanX requires depends on the number of signatures. If you enable all features, TbScanX uses 30 kilobytes of memory when scanning for 1400 family signatures. If you enable swapping, TbScanX normally uses only one kilobyte of memory. You can swap to EMS and XMS memory. Naturally you can load the remaining kilobyte of TbScanX into upper memory.

*** In the heuristic cleaning mode TbClean requires much more memory, depending on the size of the infected file. TbClean can also use expanded memory (EMS).

4.1.2 Reducing Memory Requirements

Most PC users try to maintain as much free DOS memory as possible. The memory resident TBAV utilities (TbScanX, TbCheck, TbMem, TbFile, TbDisk, TbLog and TbDriver) use only a small amount of DOS memory. To decrease the memory requirements of these utilities even further, do the following:

Load the programs from within the CONFIG.SYS file. When loaded as a device driver, a TBAV utility has no Program Segment Prefix (PSP, a DOS-internal memory area), which saves 256 bytes for each TBAV utility.

If you load the TBAV utilities from within the AUTOEXEC.BAT file, load them before establishing environment variables. DOS maintains a list of environment variables for every resident program, so keep this list small while installing TSRs. Once you install all TSRs, you can then define all environment variables without affecting the memory requirements of the TSRs.

Make use of memory swapping. If you use the EMS or XMS option, TbScanX swaps itself to non-DOS memory, leaving only one kilobyte of code in DOS memory. It is better to swap to expanded memory (EMS option) because it is faster.

Use high memory if possible. If you have DOS 5 or higher, try to load the program into an upper memory block using the LOADHIGH or DEVICEHIGH commands. We recommend that you also enable swapping to limit the use of upper memory.

Use one of the processor specific versions of the relevant TBAV utility. They all consume less memory than the generic versions. Processor optimized versions are available on any ThunderBYTE support BBS.

Use memory-saving program options. Consider using TbDriver's NOSTACK option, TbMem's NOCANCEL option, and TbScanX's NOBOOT, EMS and XMS options.

4.2 Understanding TbSetup

This section presents advanced user information about TbSetup. It explains the design of ANTI-VIR data files, editing the TBSETUP.DAT file, and how to easily install TBAV on several machines.

4.2.1 Understanding ANTI-VIR.DAT File Design

Most ThunderBYTE Anti-Virus utilities expect every directory on your system with executable files to contain its own ANTI-VIR.DAT file. Some other anti-virus products maintain a somewhat similar fingerprint list of all executable files, but in one large file rather than a separate file in each directory. TBAV's approach is superior for several reasons:

One file in each directory is easy to maintain. If you want to remove the complete product, you can remove the accompanying ANTI-VIR.DAT file as well.

It consumes less disk space because it is not necessary to store full path information in the information file.

The TBAV utilities perform faster because they do not have to search through a huge file to locate the information for one specific file.

Installation is easier and more reliable in network environments. On a network, it is not unusual that the same files have different drive ID's on different workstations. If there is only one information file, the drive-IDs should be stored as well, so every workstation should maintain its own list. The supervisor can quickly lose control in this type of situation.

4.2.2 Editing the TBSETUP.DAT File

Editing the TBSETUP.DAT file is useful to TBAV site installation (see the next section). Therefore, some information on the format of this file is necessary.

Understanding the Format of TBSETUP.DAT

The format of the TbSetup.Dat file is quite simple. You can either ignore empty lines, lines starting with a semi-colon (;), and lines starting

with a percentage symbol (%), or you can treat them as comment lines. The lines with a preceding percentage symbol also appear in TbSetup's upper window.

Each entry in the TBSETUP.DAT file has four items:

1. The filename. The filename MUST appear in capital letters and without spaces.

2. The length of the file in hexadecimal notation. This field might contain a single asterisk [*] if an exact file length match is not required.

3. The file's 32-bit CRC in hexadecimal notation. You can use a single asterisk if an exact checksum match is not required.

4. The hexadecimal number representing flags you want set when the listed file is found on the system.

You can use the rest of the line for a brief comment.

You can use the following flags. If several flags require setting for a file, you can combine them using the bitwise OR operation:

```
bit 0:    (0001)    Do not perform heuristic analysis
bit 1:    (0002)    Ignore CRC changes (self-modifying file)
bit 2:    (0004)    Scan for all signatures (LAN remote boot file)
bit 3:    (0008)    Do not change read-only attribute of this file
bit 4:    (0010)    The program stays resident in memory
bit 5:    (0020)    The program performs direct disk access
bit 6:    (0040)    Program is allowed to remove read-only
                    attributes
bit 15:   (8000)    Interrupt rehook required for TBDRIVER.EXE
```

The following are a few example entries from a TBSETUP.DAT file:

```
; filename    Length 32-bit CRC Flags  Comment

; Files that trigger the heuristic alarm of TbScan:
4DOS.COM      19FEA         *   0001    ;4Dos 4.0a
AFD.COM       0FEFE   4B351A86  0001    ;AFD debugger
ARGV0FIX.COM  001D8   431E70C0  0001    ;Argv[0]fix
EXE2COM.EXE   00BEA   49276F89  0001    ;Exe to Com conv. util
KILL.EXE      00632   74D41811  0001    ;PcTools 6.0 utility
WATCH.COM     003E1   2353625D  0001    ;TSR monitoring util
```

```
; Files that need to be scanned completely, for ALL viruses:
NET$DOS.SYS      *         *   0004    ;Disk-image Novell boot

; Files without fixed checksum due to internal config area's:
Q.EXE            *         *   000A    ;Qedit (all versions)
TBCONFIG.COM     *         *   000A    ;all versions
```

Defining New Entries in TBSETUP.DAT

If you have any files that we should include in TBSETUP.DAT, please let us know! We would like to receive a copy to enhance our products and keep TBSETUP.DAT up to date. Candidates for inclusion are any programs that trigger the heuristic analysis of TbScan.

Whenever you choose "V)alidate program" in the TbScan message window, you will discover that on subsequent occasions TbSetup displays the value "0001" in the flags field. If your company has several files like this installed on multiple machines, you might want to include these files in the TBSETUP.DAT file yourself. To do this, execute TbSetup for the file in question and make a note of its file length and 32-bit CRC, as displayed on the screen. Then edit the TBSETUP.DAT file, entering the exact filename, the file length, and the CRC number, plus the number of any flags you wish to set for that file. If you now use TbSetup on another machine (using the updated TBSETUP.DAT file), it sets the appropriate flags automatically.

TIP:

You can manually set or clear a flag field value when executing TbSetup at the DOS prompt using the SET or RESET option as follows:

```
TBSETUP TEST.EXE SET=0001.
```

4.2.3 Simplifying Installation on Several Machines

If you need to install the TBAV utilities on several machines in one company, it would be tedious, for example, to run every TSR and disk utility on each machine to "teach" TBAV which programs are valid and which are not. Fortunately, this is not necessary. We present here some examples of how to simplify installation on several machines.

If a resident utility named, for example, TSRUTIL.EXE, is in use throughout the company, you can predefine permission by using TbSetup. First, use TbSetup to determine the length and CRC of the program.

Second, put the name of the program, along with its other information, in the TBSETUP.DAT file, and then assign the flag 0010 to it:

```
TSRUTIL.EXE  01286  E387AB21  0010  ;OUR TSR UTILITY
```

If a disk utility named, for example, DISKUTIL.EXE, is in use throughout the company, you can predefine permission by using TbSetup. First, use TbSetup to determine the length and CRC of the program. Second, put the name of the program, along with its other information, in the TBSETUP.DAT file, and then assign the flag 0020 to it:

```
DISKUTIL.EXE 01286  E387AB21  0020  ;OUR DISK UTILITY
```

If a utility named, for example, UTIL.EXE, causes TbScan to give false positives and is in use throughout the company, you can use TbSetup to "teach" TbScan to avoid heuristic scanning of the program. First, use TbSetup to determine the length and CRC of the program. Second, put the name of the program, along with its other information, in the TBSETUP.DAT file, and then assign the value 0001 to it:

```
UTIL.EXE     01286  E387AB21  0001  ;OUR UTILITY
```

If you now run TbSetup on every machine (you have to do this anyway), it recognizes the utilities you added in the TBSETUP.DAT file. Additionally, all the TBAV utilities automatically adapt their behavior for those files.

TIP:
Consult the TBSETUP.DAT file itself. It contains useful comments on this subject.

4.3 Understanding TbScan

This section offers advanced information about TbScan, including: heuristic scanning, integrity checking, program validation, algorithms, and the TBSCAN.LNG file.

4.3.1 Understanding Heuristic Scanning

What makes TbScan so unique is that it is not just a signature scanner, but it is also a disassembler. It disassembles files for the following purposes:

By disassembling a file, the scanner restricts itself to the area of the file where the virus might reside, reducing false alarms and speeding up the process. Disassembling a file makes it possible to use the algorithmic detection method on encrypted viruses whose signatures would otherwise remain invisible to the scanner.

Disassembling the file makes it possible to detect suspicious instruction sequences.

This detection of suspicious instruction sequences is "heuristic scanning." This extremely powerful feature enables you to detect new or modified viruses and to verify the results of the signature scan. You no longer have to rely on the scanner's publisher having the same virus as you might have. In normal cases a scanner can find a virus only if the scanner's publisher had a sample of that virus and includes that virus's signature in a signature file. In contrast, heuristic scanning does not require signatures, enabling the scanner to detect yet unknown viruses by looking for the characteristics of a virus instead of a signature.

Never underestimate the importance of heuristic scanning, since every month at least 50 new viruses are reported, and it is extremely unlikely that a publisher is the first one to get a new virus.

TbScan distinguishes two heuristic levels. The following table describes the properties of these levels:

Heuristic Level 1	Heuristic Level 2
always enabled	only enabled with command-line option "heuristic", or TBAV menu option "High heuristic sensitivity," or after a virus has been found
detects 50 % of (yet) unknown viruses	detects 90 % of (yet) unknown viruses
almost never causes false alarms	might cause few false alarms
displays "Probably infected"	displays "Might be infected"

The following lines show the effect of scanning four files, each having its own characteristics. Please note the heuristic flags that appear next to the word "scanning."

```
FILE1.EXE   scanning...OK      (no flags)
FILE2.EXE   scanning...ROK     (nothing serious)
FILE3.EXE   scanning...FRM     might be infected by unknown virus
FILE4.EXE   scanning...FRALM# probably infected by unknown virus
```

It is obvious from these four examples that heuristic scanning (resulting in the heuristic flags) is very powerful for finding yet unknown viruses.

4.3.2 Understanding How Heuristic Scanning Works

Every program contains instructions for the computer's microprocessor. By looking into the file's contents and interpreting the instructions, TbScan is able to detect the purpose of these instructions. If the purpose appears to be formatting a disk, or infecting a file, TbScan issues a warning. There are many instruction sequences that are very common for viruses but are very uncommon for normal programs. TbScan, therefore, assigns every suspicious instruction sequence to a character called a heuristic flag. Every heuristic flag denotes a score. If the total score (that is, the sum of scores for each flag that triggered) exceeds a predefined limit, TbScan assumes the file contains a virus.

There are actually two predefined limits. The first limit is quite sensitive and can be reached by some normal innocent programs. If the suspicious program reaches this limit, TbScan highlights the heuristic flags that appear on the screen and increases the suspicious item's counter. TbScan does not indicate the existence of a virus unless you specify the heuristic or high heuristic sensitivity option. If you do specify this option, TbScan informs you that the file Might be infected by an unknown virus.

In contrast to the first option, many viruses trigger the second heuristic limit, while normal programs do not. If a suspicious program reaches this limit, TbScan informs you that the file is Probably infected by an unknown virus.

NOTE:

TbScan performs heuristic analysis only near the entry-point of a file. Therefore, TbScan does not detect direct writes to disk by some disk utilities nor does it detect some programs as TSR programs. This is simply the result of a specific approach that minimizes false alarms. In case of a virus, the offending instructions are always near the entry-point (except when the virus is over 10Kb in size), so TbScan detects suspicious phenomena in these situations anyway.

4.3.3 Understanding Integrity Checking

TbScan performs integrity checking while scanning. For this purpose, you must use TbSetup to generate the ANTI-VIR.DAT files. Once these files exist on your system, TbScan verifies that every file being scanned matches the information maintained in the ANTI-VIR.DAT files. If a virus infects a file, the maintained information no longer matches the now changed file, and TbScan informs you of this.

NOTE:

> There are no command line options to enable this feature. TbScan performs integrity checking automatically if it detects the ANTI-VIR.DAT files.

Note that TbScan reports only those file changes that could indicate a virus. While internal configuration areas of program files might also change, TbScan normally does not report these. If a file becomes infected with a known or unknown virus, however, the vital information does change and TbScan does indeed report it to you!

In contrast, there might be files that change themselves frequently or change frequently due to another cause. In such a case you might want to exclude the program from integrity checking to avoid future false alarms. If TbScan detects such a change, it informs you of it. Additionally, TbScan offers the possibility to Validate the program, which is the subject of the next section.

Understanding Program Validation This section applies only if you use TbSetup to generate the ANTI-VIR.DAT records. Without these records, program validation is not an option.

TbScan performs as intended on most programs. There are some programs, however, that require special attention in order to avoid false alarms. TbSetup recognizes most of these programs automatically. Nevertheless it is certainly possible your PC contains some program files that trigger the heuristic alarm of TbScan and/or programs files that change frequently.

If TbScan finds an infection using heuristic analysis or integrity checking, and if there is an ANTI-VIR.DAT record available, it offers an additional option in its virus-alert window, namely, V)alidate program.

If you are sure that the indicated program does not contain a virus, you can press V to set a flag in the program s ANTI-VIR.DAT record. This avoids future false alarms.

There are two validation modes. If TbScan alarms you to a file change, the validation applies to future file changes only. If the alarm is due to heuristic analysis, the validation applies only to heuristic results. If you exclude the file from heuristic analysis, TbScan still performs an integrity check. Conversely, if you exclude the file from integrity checking, TbScan still performs heuristic analysis.

CAUTION:

If you replaced a file (for example, because of a software upgrade) and you did not apply TbSetup to the changed files, TbScan pops up its virus alert window to inform you of the file change. Do not select the validation option in this case, because this would exclude the file from future integrity checking. You should abort TbScan and execute TbSetup on the changed file(s) instead.

4.3.4 Understanding the Scan Algorithms

When TbScan processes a file it displays one of the following messages:

Looking.

"Looking" indicates that TbScan has successfully located the entry point of the program in one step; that is, it has identified the program code so it knows where to search without the need of additional analysis. TbScan uses "Looking" on most known software.

Checking.

"Checking" indicates TbScan has successfully located the entry point of the program, and is scanning a frame of about two kilobytes around the entry point. If the file is infected, the virus signature appears in this area. "Checking" is a very fast and reliable scan algorithm, so TbScan applies it to most unknown software.

Tracing.

"Tracing" means that TbScan has successfully traced a chain of jumps or calls while locating the entry point of the program and is scanning a frame of about two kilobytes around this location. If the

file has been infected, the signature of the virus appears in this area. "Tracing" is a fast and reliable scan algorithm. TbScan uses it primarily for memory resident COM programs. Most viruses force TbScan to use "Tracing."

Scanning.

"Scanning" indicates that TbScan is scanning the entire file (except for the EXE-header that cannot contain any viral code). It uses this only if it can't safely use "Looking," "Checking," or "Tracing." Such is the case when the entry point of the program contains other jumps and calls to code located outside the scanning frame, or when the heuristic analyzer finds something that you should investigate more thoroughly. Because Scanning is a slow algorithm, it processes almost the entire file, including data areas, and it is more likely to trigger false alarms. TbScan uses this algorithm when scanning boot sectors, SYS files, and BIN files.

Skipping.

"Skipping" occurs only with SYS and OVL files. It simply means that the file will not be scanned. As there are many SYS files (such as CONFIG.SYS) that contain no code at all, it makes absolutely no sense to scan these files for viruses. The same applies to .OV? files. Many overlay files do not deserve the name overlay because they lack an EXE-header. Such files cannot execute through DOS, which in-turn makes them just as invulnerable to direct virus attacks as .TXT files. If TbScan reports that a virus has infected an .OV? file, that file is one of the relatively few overlay files that does contain an EXE-header. In such a case, the infection was the result of the virus monitoring the DOS exec-call (function 4Bh) and thereby infecting any program that executes that way, including real overlay files.

Decrypting.

TbScan detected that the file is encrypted, and decrypts it to be able to "look inside." TbScan performs signature scanning and heuristic analysis on the decrypted code since that is very reliable and also reveals polymorphic viruses.

4.3.5 Understanding the TBSCAN.LNG File

The TBSCAN.LNG file contains all the text that TbScan displays. You can translate or customize the messages with any ASCII editor. A dollar sign [$] separates the messages.

The first message displays our address and registration information. You can edit this message as you please, adding, for example, your company name and logo.

CAUTION:

> Take care in customizing messages so that you don't change the essence of the message.

You can also add color codes to the TBSCAN.LNG file. You must precede a color code with the "pipe" [|] character. Each color code consists of a foreground (or highlight) color and a background color. The following table lists the available color codes (all numbers are in hexadecimal notation):

Color	Foreground	Highlight	Background
Black	00	08	00
Blue	01	09	10
Green	02	0A	20
Cyan	03	0B	30
Read	04	0C	40
Magenta	05	0D	50
Brown	06	0E (yellow)	60
Gray	07	0F (white)	70

To make characters blink, add 80 to the background color codes.

Here are few examples of defining colors:

> To make a highlighted green character on a red background, use the color code 0A+40=4A. To make the character blink, add 80h to the result (4A+80=CA). To display white characters on a blue background, use the color code 0F in combination with color code 10: 0F+10=1F.
>
> If you prefer a cyan background with a gray foreground, you should add 30 to 07 (30+07=37). If you want the characters to blink, the color code becomes 37+80=B7.

4.3.6 Understanding the TBAV.MSG File

The TBAV menu displays the contents of a file named TBAV.MSG, if it exists in the ThunderBYTE directory. You can use this feature to display your company logo on the TBAV screen. As in the TbScan language file, you can embed color codes in this file. Consult the previous section for more information about color codes.

4.4 Understanding TbClean

This section takes a look at how TbClean works by explaining how a virus goes about infecting a file and the difference between conventional cleaners and generic cleaners.

4.4.1 Understanding how a Virus infects a file

To understand how a cleaning program works, try to imagine how a virus usually goes about infecting a program. The basic principle is really quite simple. A virus, which is simply another computer program, adds itself to the end of the program it infects. The additional viral code obviously increases the size of the program.

Simply appending a viral program to another program, however, is not enough to do any real harm. To do damage, the viral code must first be executed. To accomplish this, the virus grabs the first few bytes at the start of the program and replaces them with a jump instruction to its own viral code. That way the virus is able to take control when the program starts. Chances are you will never even notice the momentary delay while the extra code executes and does whatever the virus has been programmed to do. The virus then restores the original instructions and restarts the program (jumps to the original start of the program). Your program, more often than not, works as usual, and of course, any virus worth its salt makes sure it doesn't draw undue attention to itself, at least not too soon.

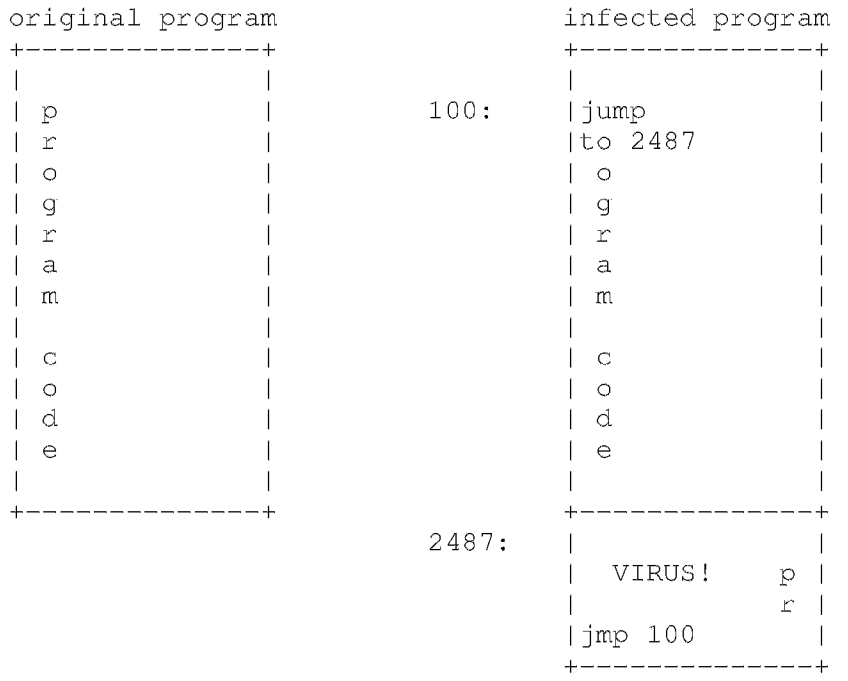
So, in order to purge a program, we must first restore the starting instruction bytes, which the virus replaced with the jump to its own code. The virus is going to need these bytes again later on, so it stores them somewhere in the viral code. The cleaner starts out to find those bytes, puts them back in their proper place, and trims the file to the original size.

Cleaner programs basically come in two types: the conventional type, for specific types of viruses, and the far more advanced generic cleaner, which offers a much wider scope. Let's take a closer look at both cleaner types and find out where they differ.

4.4.2 Understanding Conventional Cleaners

A conventional cleaner has to know which virus to remove. Suppose one of your programs is infected with a Jerusalem/PLO virus. This means that the

infected program has grown in size in comparison with the original program, and that the first few bytes have been replaced by a "jump" instruction to the viral code. The following drawing ilustrates this process:



When you start a conventional cleaner, a procedure much like the following takes place:

> "Hey, the signature file tells me this file is infected with the Jerusalem/PLO virus. Okay, let's see, this virus tacks on 1873 bytes at the end and overwrites the first three bytes of the original program with a jump to itself. The original bytes are located at offset 483 in the viral code. So, I have to take those bytes, copy them to the beginning of the file, and then remove 1873 bytes of the file. That's it!"

But there are several pitfalls to worry about in a scenario like this. For one thing, the cleaner obviously must have some means to recognize the virus it should remove. A conventional cleaner cannot cope with a virus unless it knows exactly what to look for.

To make matters worse, it's even more important to establish whether or not the virus is exactly the same one that the cleaner knows about. Imagine what would happen if the virus in our example had been modified

and is now 1869 bytes in size instead of 1873. The cleaner would remove too much! This is not an exceptional case at all. On the contrary, there is a virtual epidemic of countless so-called mutant strains. The Jerusalem/PLO family, to name but one example, now has more than 100 mutant members!

4.4.3 Understanding Generic Cleaners

A generic cleaner works on the principle that any kind of virus, whether or not it has made the signature "charts," is just plain bad news. That's why TbClean works with a completely different disinfection scheme that is effective with almost all viruses; it doesn't even need to recognize them. Actually, TbClean represents two cleaners in one: a "repair" cleaner and a "heuristic" cleaner.

Repair cleaning

Repair cleaning needs an ANTI-VIR.DAT file generated by TbSetup before the infection occurred. The ANTI-VIR.DAT file stores vital information about programs, including their original size, the first few instruction codes, and a cryptographic checksum. This information is usually all it takes to disinfect a file, no matter what virus, known or unknown, caused the infection. The cleaner simply restores the bytes at the beginning of the program, trims the file to its original size, and verifies the result using the original checksum. It's just that simple (and effective).

Heuristic cleaning

TbClean is the first cleaner in the world that has a heuristic cleaning mode. Like the repair cleaner, this mode does not need any information about viruses either, but it also has the added advantage that it doesn't even care about the original, uninfected state of a program. This cleaning mode is very effective if your system becomes infected with an unknown virus and you neglected to let TbSetup generate the ANTI-VIR.DAT files before infection.

In heuristic mode, TbClean loads the infected file and starts emulating the program code. It uses a combination of disassembly, emulation and, sometimes, execution to trace the flow of the viral code, pretending to do more or less exactly what the virus would normally be doing. When the virus gets to the original program's instructions and jumps back to the original program code, TbClean stops the emulation process, with a

tongue-in-check thank you to the virus for its cooperation in restoring the original bytes.

The actual cleaning process involves almost the same three steps as with repair cleaning. First, TbClean repairs the program startup code and copies it back to the file. Second, it removes the now ineffective code for the sake of security. Third, it does a final analysis of the purged program file.

4.5 Using TbGenSig

This final section of Chapter 4 introduces you to TbGenSig, an advanced user utility that enables you to define your own virus signatures.

4.5.1 Understanding and using TbGenSig

TbGenSig is a signature file compiler. Since we distribute TBAV with an up to date, ready-to-use signature file, you do not really need the signature file compiler.

If, however, you want to define your own virus signatures, you will need this utility. You can use either published signatures or define your own, if you are familiar with the structure of software.

One way or another, you need to do this only in case of an emergency, such as in the unfortunate event that a yet unknown, and thus unrecognized, virus attacks your machine, or even your company. We recommend that you send a few samples of the virus to some of our researchers, to insure that they can be examined and the results included in one of the subsequent updates to our software.

NOTE:
Since it's not possible to explain the whole subject of virus hunting in one manual, this section assumes you have enough experience and knowledge to create your own virus signatures.

TbGenSig searches for the USERSIG.DAT file in the current directory. This file should contain the signatures you want to add to the TBAV signature file TBSCAN.SIG. TbGenSig checks the contents of the USERSIG.DAT file and applies it to the TBSCAN.SIG file.

If you want to delete or modify your signatures, just edit or delete the USERSIG.DAT file and run TbGenSig again.

TbGenSig lists all signatures in the TBSCAN.SIG file on screen as it runs.

4.5.2 Working with TbGenSig

This section describes how to use TbGenSig. It outlines how to format the text in the USERSIG.DAT file, add published signatures, define your own signatures, and other procedures.

Formatting Text in USERSIG.DAT

You can create and edit the USERSIG.DAT file using any DOS text editor (such as DOS 5+ EDIT program) that uses un-formatted (ASCII) text. All lines starting with a semicolon (;) are comment lines. TbGenSig ignores these lines. Lines starting with a percentage character (%) appear in the upper TbGenSig window.

The first line should contain the name of a virus, the second line contains one or more keywords, and the third line contains the signature itself. We call this combination of three lines a signature record. A signature record should look like this:

```
TEST VIRUS
EXE COM INF
ABCD21436587ABCD
```

You can use spaces in the signature for your own convenience; TbGenSig will just ignore them.

Adding a Published Signature

As outlined above, adding an already published signature is simply a matter of editing or creating the USERSIG.DAT file to convert the signature to an acceptable format for TbGenSig. Format the three lines to include the virus name, keywords, and the signature, as in the following:

```
NEW VIRUS
EXE COM BOOT INF
1234ABCD5678EFAB
```

After editing the file, execute TbGenSig.

4.5.3 Defining a Signature with TbScan

This section is for advanced users who have registered their copy of ThunderBYTE Anti-Virus.

Although the TBSCAN.SIG file updates frequently, new viruses appear every day, outpacing the regular upgrading service of the TbScan signature file. It is possible for your system to become infected by a recently created virus not yet listed in the signature file. TbScan will not

always detect the virus in such cases, not even with its heuristic analysis. If you are sure that your system has become infected without TbScan confirming this, this section will supply you with a valuable tool to detect unknown viruses. This section offers step-by-step assistance in creating an emergency signature that you can (temporarily) add to your copy of TbScan.Sig

1. Collect some infected files and copy them into a temporary directory.

2. Boot from a clean write-protected diskette.

WARNING:
Do NOT execute ANY program from the infected system, even though you expect this program to be clean.

3. Execute TbScan from your write-protected TbScan diskette using the EXTRACT option. Make sure that the temporary directory where you stored the infected files is TbScan's target directory. Using the EXTRACT option, TbScan will NOT scan the files but, instead, displays the first instructions that it finds at the entry-point of the infected programs.

NOTE:
We recommend that you also set TbScan's LOG option to generate a log file.

4. Compare the "signatures" extracted by TbScan. You should see something like this:

```
NOVIRUS1.COM    2E67BCDEAB1290909 09090 ABCD123490CD
NOVIRUS2.COM    N/A
VIRUS1.COM      1234ABCD5678EFAB9 09090 ABCD123478FF
VIRUS2.COM      1234ABCD5678EFAB9 01234 ABCD123478FF
VIRUS3.COM      1234ABCD5678EFAB9 A5678 ABCD123478FF
```

If the "signatures" of the files are completely different, the files are either probably not infected, or they have become infected by a polymorphic virus that requires an algorithmic detection module to detect it.

5. If there are some differences in the "signatures," you can use the question mark wildcard (?). A signature to detect the virus in the example above could be:

```
1234ABCD5678EFAB ?3 ABCD123478FF
```

The "?3" means that there are three bytes at that position that should be skipped. Note that two digits in the signature represent a byte in your program.

6. Add the signature to USERSIG.DAT. Give the virus a name in the first line of its entry, specify the COM, EXE, INF, and ATE in the second line, and enter the signature in the third, as in the following:

```
NEW VIRUS
EXE COM ATE INF
1234ABCD5678EFAB?3ABCD123478FF
```

7. Run TbGenSig. Make sure the resulting TbScan.Sig file is in the TBSCAN directory.

8. Run TbScan again in the directory containing the infected files. TbScan should now detect the virus.

9. Send a couple of infected files to a recommended virus expert, preferably to the ThunderBYTE Corporation.

Congratulations! You have defined a signature all by yourself! Now you can scan all your machines in search of the new virus.

CAUTION:

Keep in mind that this method of extracting a signature is a "quick-and-dirty" solution to viral problems. The extracted signature might not detect the presence of the virus in all cases. You can make a signature guaranteed to detect all instances of the virus only after complete disassembly of the new virus. For these reasons you should NEVER distribute your home-made "signature" to others. In most cases, the signature eventually assembled by experienced anti-virus researchers may be different from your homemade version.

4.5.4 Understanding Keywords

You can use keywords for several purposes. You can separate them by spaces, commas, or tabs and use a maximum line length of 80 characters. You also should specify at least one of the following flags: BOOT, COM, EXE, HIGH, LOW, SYS, or WIN.

These seven flags fall into three categories: "Item Keywords," "Message Keywords," and "Position Keywords."

Using Item Keywords

Item keywords tell the scanner where to search for viruses with those keywords. For example, the BOOT keyword tells the scanner that the accompanying virus signature can reside only in a boot sector or partition table. The Item keywords include the following:

BOOT. Specifies that the signature can be found in boot sectors and/or partition tables.

COM. Specifies that the signature can be found in COM programs. This flag instructs the scanner to search for this signature in executable files that do not have an EXE header or device header.

NOTE: Always keep in mind that the file content determines the file type, not the filename extension!

EXE. Specifies that the signature can be found in EXE programs. This flag instructs the scanner to search for this signature in the load module of EXE type files. EXE files are files that have an EXE header. (See the Note under the COM keyword.)

HIGH. Specifies that the signature can be found in HIGH memory (above program).This flag instructs the scanner to search for this signature in memory above the memory allocated by the scanner. This keyword is for resident viruses that allocate memory at "system boot" or viruses that decrease the size of the last MCB (Memory Control Block). Please note that the flag HIGH does not mean that the signature should be searched in UPPER memory.

LOW. Specifies that the signature can be found in LOW memory. This flag instructs the scanner to search for this signature in memory below the PSP (Program Segment Prefix) of the scanner and in the UMBs (Upper Memory Blocks). This keyword is for viruses that remain resident in memory, using the normal DOS TSR (Terminate and Stay Resident) function calls.

SYS. Specifies that the signature can be found in SYS programs, such as device drivers.

WIN. Specifies that the signature can be found in Windows programs.

Message keywords

Message keywords describe the type and behavior of the virus. For each keyword, this results in the scanner displaying a different message when it finds such a virus. These keywords include the following:

DAM.	Message prefix:	damaged by.
DROP.	Message prefix:	dropper of.
FND.	Message prefix:	found the.
INF.	Message prefix:	infected by.
	Message suffix:	virus.
JOKE.	Message prefix:	joke named.
OVW.	Message prefix:	garbage: (not a virus).
PROB.	Message pre-prefix:	probably.
TROJ.	Message prefix:	trojanized by.

Position keywords

Position keywords indicate special file areas where the virus can be found. If you use a position keyword, the virus must reside at the specific position. TbGenSig can handle three position keywords:

UATE. Specifies that the signature starts directly at the unresolved entry-point of the viral code. With some polymorphic viruses, it might be possible to create a signature from the degarbling routine, although it might be either too short or give false positives with a global search. An initial branch instruction can be part of the signature. The unresolved entry-point is defined for COM-, EXE-, and Windows-type files:

COM type files:	top of file (IP 0100h).
EXE type files:	CS:IP as defined in the EXE-header.
WIN type files:	Non-DOS CS:IP of the new EXE-header.

NOTE:

The UATE keyword is not allowed for BOOT, SYS, LOW, HMA, or HIGH type signatures.

ATE. Specifies that the signature starts directly at the entry-point of the viral code. With some polymorphic viruses, it might be possible to create a signature from the degarbling routine, although it might either be too short or give false positives with a global search. Therefore, use the ATE keyword to ensure that the scanners do not scan the entire file for the signature, but only look at the entry-point for the signature.

The first instruction that is not equal to either a "JUMP SHORT," a "JUMP," or a "CALL NEAR" instruction defines the entry point of a virus.

Let's examine the following code fragment:

```
Unresolved entry point:  1    JUMP SHORT 3
                         2    ...
                         3    JUMP 5
                         4    ...
                         5    CALL NEAR 7
                         6    ...
                         7    CALL NEAR 9
                         8    ...
Resolved entry point:    9    POP <reg>
```

The entry-point of the above fragment is Line 9, as this is the first instruction to execute that is not a "JUMP SHORT," a "JUMP," or a "CALL NEAR."

NOTE: You can determine the entry-point by a code analyzer to cope with tricks such as coding an NOP or DEC just before the branch instruction. Therefore test the results of the scanner carefully. In case of trouble, use the TbScan EXTRACT option to find out what TbScan considers to be the entry point of the program. Also, the ATE flag is not allowed for BOOT, SYS, LOW, HMA or HIGH type signatures.

XHD. Specifies that the signature can be found at offset 2 of the EXE header, but is rarely used. You should use it only to detect the also very rare high-level language viruses, viruses written in a programming language such as C or Basic. These viruses normally contain standard setup

routines and library routines that are not suitable to defining a signature. Use this keyword as a last resort to detect such viruses.

NOTE:
You can use this flag only for EXE or WIN type signatures.

Using Wildcards

You can use wildcards characters in a virus signature to recognize so called polymorphic (self-modifying or self-mutating) virus code. TbGenSig distinguishes two wildcard categories: position wildcards and opcode wildcards (note that all numbers are in hexadecimal):

Using Position Wildcards

Position wildcard affect the position where the parts of the signature match.

Skip fixed amount of bytes

?n Skip n bytes and continue.(0h <= n <= Fh)
?@nn Skip nn bytes and continue.(00h <= nn <= 7Fh)

Skip variable amount of bytes

*n Skip up to n bytes and continue. (0h <= n <= Fh)
*@nn Skip up to nn bytes and continue. (00h <= nn <= 1Fh)

Using Opcode wildcards

The opcode wildcards detect instruction ranges.

Low opcode
nL One of the interactions in the range of n0h to n7h.

High opcode

nH One of the interactions in the range of n8h up to nFh.

Since the opcode wildcards are rather difficult to understand, let's explore an example.

Suppose a polymorphic virus puts a value in a word register (using a MOV WREG,VALUE instruction), increments a register (using an INC WREG instruction), and pops a word register from the stack (using a POP instruction). Both the registers and the value are variable. This means that the signature you are writing to detect this virus should be able to detect all code sequences for every value of the registers and the value, but this is far too much work. Now, consider that B8-BF are the opcodes for MOV WREG,VALUE, that 40-47 are the opcodes for INC WREG, and that 58-5F are the opcodes for POP REG.

By using the opcode wildcards, you can detect a sequence of these three instructions using the following signature fragment:

```
bH4L5H
```

4.5.5 Understanding a Sample Signature: Haifa.Mozkin

To show the power of using the appropriate keywords and wildcards, here is the signature of the Haifa.Mozkin virus. This virus is highly polymorphic and encrypted. It contains a small variable decryptor to decrypt the virus.

There are two problems here: most bytes are encrypted or variable, thus not suitable to be part of a signature, and the remainder is short and would cause dozens of false alarms.

Using the appropriate keywords and wildcards, however, it s possible to define a reliable signature. TbScan actually uses the signature below to detect the Haifa.Mozkin virus.

```
Haifa.Mozkin
com exe ate inf
bh?2bh?109?2*22e80?24l4h75fl
```

Now let's analyze this signature. The first line describes the name of the virus. The second line tells the scanner to search for this signature in COM and EXE type files. It also tells the scanner that it should report the file as infected if the signature matches. The keyword ATE instructs the scanner to match this signature only at the resolved entry-point of the file. The virus starts, of course, by decrypting itself, so it is certain that the scanner will scan this location. The ATE instruction limits the scope of this signature to just one position in a file, so this significantly reduces the chances of false alarms.

The third line is the signature definition. Let's reverse engineer it:

bh?2 Means a byte in the B8-BF range is followed by two variable bytes. B8-BF is a MOV WREG,VALUE instruction. From the register we only know it is a word register; the value is unknown as well.

bh?109 Means another MOV WREG,VALUE instruction. The register is a word register, and from the value we know that it is in the range 0900 to 09FF.

?2*2 Means skip two to four bytes. The virus inserts this instruction to make it harder to define a signature.

2e80?2 Means that the virus performs an arithmetic byte sized operation with an immediate value (decrypts one byte) with a CS: segment override. The exact operation, the memory location, and the value are unknown.

4l Means a byte in the 40-47 range. This is an INC WREG instruction. The virus increments the counter to the next byte to be decrypted.

4h Means a byte in the 48-4F range. This is a DEC WREG instruction. The virus decrements the iteration count.

75f1 Opcode 75 is a JNZ instruction. If the decremented register did not reach zero, the virus jumps back and repeats the operation. How much does it jump? That tells the f1 part: somewhere between -16 (F0h) to -8 (F7h) bytes.

NOTE:

Although the signature language of TbGenSig is extremely powerful, there are viruses that are simply so highly polymorphic that they require even more sophisticated wildcards, keywords, or even special detection algorithms. The explanation of these wildcards, keywords, and algorithmic detection definitions, however, is beyond the scope of this user manual.

Appendices

Appendix A: TBAV messages

The TBAV utilities might display various messages when run. Most messages are self-explanatory, but here is some additional information about specific messages.

A.1 TbClean

ANTI-VIR.DAT record found: information matches the currentstate of the file.

The ANTI-VIR.DAT record has been found, but the information matches the current state of the file.

The ANTI-VIR.DAT file was created after the infection. Trying emulation...

The ANTI-VIR.DAT record was created after the file became infected, or the file is not changed at all. TbClean is going to emulate the file to clean it heuristically.

ANTI-VIR.DAT record found: reconstructing original state...

The ANTI-VIR.DAT record that belongs to the infected file has been found. The information will be used to reconstruct the file.

ANTI-VIR.DAT record not found: original state unknown. Trying emulation...

The ANTI-VIR.DAT file did not exist or did not contain information about the infected program, so the original state of the infected program is unknown to TbClean. TbClean switches to its heuristic mode to determine the state of the original file.

NOTE:

To prevent this situation, use the TbSetup program to generate the ANTI-VIR.DAT records. These records are of great help to TbClean. After infection, it's too late to generate the ANTI-VIR.DAT records.

Emulation terminated: <Reason>

The emulation process terminated for the reason specified. TbClean now consults the collected information to see if it can disinfect the file. The reason for termination can be one of the following:

Jump to BIOS code. The virus tried to perform a call or jump directly into BIOS code. TBAV cannot emulate this process, so aborts. The infected program probably cannot be disinfected.

Approached stack crash. The emulated program is approaching a crash. Something went wrong while emulating the program so it aborts. The infected program probably cannot be disinfected.

Attempt to violate license agreements. TbClean will not disassemble this program for obvious reasons.

Encountered keyboard input request. The emulated program tries to read the keyboard. This is very unusual for viruses, so the file is probably not infected at all.
Encountered an invalid instruction. The emulator encountered an unknown instruction. For some reason the emulation failed. The infected program probably cannot be disinfected.

DOS program-terminate request. The emulated program requests DOS to stop execution. The program is either not infected at all, or infected by an overwriting virus that does not pass control to its host program. The infected program cannot be disinfected.

Jumped to original program entry point. The program jumped back to the start position. It is very likely infected, but can probably be disinfected.

Undocumented DOS call with pointers to relocated code. This is very common for viruses that add themselves in front of the COM type program. The program can probably be disinfected.

Encountered an endless loop. TbClean encountered a situation in which the program is executing the same instruction sequences repeatedly for hundreds of thousands of times. It is unlikely that the program will ever escape from this loop, so the emulation aborts.

Ctrl-break pressed. The user pressed <Ctrl>-<Break> so the clean attempt aborts.

Emulation aborted for unknown reason. If this message appears, please send a copy of the file being emulated to the ThunderBYTE organization or one of the support BBS .

Sorry, the collected information is not sufficient to clean file... The heuristic cleaning mode of TbClean aborts with success. The only option left is to restore the file from a backup or to re-install the program.

Collected enough information to attempt a reliable clean operation... The emulation of the virus provided TbClean with all information needed to disinfect the file.

Some DOS error occurred. TbClean aborted! Some DOS error occurred while trying to clean the file. Check that no files are read-only or located on a write protected disk, and make sure there is a reasonable amount of free disk space.

The clean attempt seems to be successful. Test the file carefully! TbClean thoroughly and reliably removed the virus from the file. However, take care and test the file carefully to see if it works as correctly.

Reconstruction failed. Program might be overwritten. Trying emulation... TbClean tried to reconstruct the original file with the help of the ANTI-VIR.DAT record, but the attempt failed. TbClean is going to emulate the file to try to clean it heuristically.

Reconstruction successfully completed. TbClean has reconstructed the file to its original state with the help of the information in the ANTI-VIR.DAT record. The CRC (checksum) of the original file and the cleaned file are completely equal, so it is almost certain that the cleaned file is equal to the original file.

Starting clean attempt. Analyzing infected file... TbClean is analyzing the infected file and trying to locate the ANTI-VIR.DAT record.

A.2 TbDriver

Another version of TbDriver is already resident!
You started a TBDRIVER.EXE with another version number or processor type than the TbDriver already in memory.

Cannot remove TbDriver. Unload other TSRs first!

You tried to remove TbDriver from memory, but other resident software was loaded after TbDriver. You can only remove resident programs from memory by unloading them in reverse order.

LAN support was already installed.
You tried to use the NET option a second time, or TbDriver already enabled network support automatically.

TbDriver not active. Load TbDriver first!
The resident TBAV utilities need TbDriver, so you need to load TbDriver first.

TbDriver is not <version>.
The version of TbDriver found in memory does not match the version number of this resident TBAV utility. Be sure you do not mix version numbers!

This version of TbDriver requires a <typeID> processor.
You are using a processor optimized version of TbDriver that the current processor cannot execute.

A.3 TbScan

Cannot create logfile.
The specified log file path is illegal, the disk is full or write protected, or the file already exists and cannot be overwritten.

[Cannot read datafile]
TbScan needs access to its data file to be able to tell you the name of the virus. If it cannot access the data file, it displays this message instead of the virus.

Command line error.
You specified an invalid or illegal command line option.

No matching executable files found.
The specified path does not exist, is empty, or is not an executable file.

Sanity check failed!
TbScan detected that its internal checksum no longer matches. It is possible that TbScan is contaminated by a virus. Obtain a clean copy of TbScan, copy the program on a write protected system diskette, boot from that diskette, and try again.

A.4 TbScanX

Data file not found.
TbScanX cannot locate the data file.

Not enough memory.
There is not enough free memory to process the data file. Try to enable swapping, or if you are already doing so, try another swapping mode. See also the Understanding Memory Considerations section in Chapter 4.

Appendix B: TbScan Heuristic Flag Descriptions

This appendix describes TBAV's heuristic flags.

- Decryptor code found

The file possibly contains a self-decryption routine. Some copy-protected software is encrypted, so this warning might appear for some of your files. If, however, this warning appears in combination with, for example, the "T" warning, there could be a virus involved and TbScan assumes contamination. Many viruses encrypt themselves and trigger this warning.

! - Invalid program.

Invalid opcode (non-8088 instructions) or out-of-range branch. The program has either an entry point that located outside the body of the file, or reveals a chain of jumps that can be traced to a location outside the program file. Another possibility is that the program contains invalid processor instructions. The program being checked is probably damaged and cannot execute in most cases. At any rate, TbScan avoids risk and uses the scan method to scan the file.

1 - 80186+ instructions.

The file contains instructions which cannot be executed by 8088 processors, and require an 80186 or better processor.

@ - Strange instructions

The file contains instructions which are not likely to be generated by an assembler, but by some code generator like a polymorphic virus instead.

? - Inconsistent header.

The program being processed has an EXE-header that does not reflect the actual program lay-out. Many viruses do not update the EXE-header of an EXE file correctly after they infect the file, so if this warning pops up frequently, it appears you have a problem.

c - No integrity check

This warning indicates that TBAV found no checksum/recovery information for the indicated file. We recommend you use TbSetup in this case to store the file's information. TBAV uses this information for integrity checking and to recover from virus infections.

h - Hidden or System file.

The file has the Hidden or the System file attribute set. This means that the file is not visible in a DOS directory display but TbScan scans it anyway. If you don t know the origin and/or purpose of this file, you

might be dealing with a Trojan Horse or a joke virus program. Copy such a file onto a diskette, remove it from its program environment, and then check if the program concerned is missing the file. If a program does not miss it, you not only have freed some disk space, but you might also have prevented a future disaster.

i - Internal overlay.
The program being processed has additional data or code behind the load-module as specified in the EXE-header of the file. The program might have internal overlay(s) or configuration or debug information appended behind the load-module of the EXE file.

p - Packed or compressed file.
This means that the program is packed or compressed. There are some utilities that can compress program files, such as EXEPACK and PKLITE. If the file became infected after compression, TbScan is able to detect the virus. However, if the file became infected before compression, the virus was also compressed in the process, and a virus scanner might no longer be able to recognize the virus. Fortunately, this does not happen very often, but you should still beware! A new program might look clean, but can turn out to be the carrier of a compressed virus. Other files in your system will become infected too, and it is these infections that will be clearly visible to virus scanners.

w - Windows or OS/2 header.
The program can be or is intended to run in a Windows (or OS/2) environment. TbScan offers a specialized scanning method for these files.

A - Suspicious Memory Allocation
The program uses a non-standard way to search for, and/or to allocate memory. Many viruses try to hide themselves in memory, so they use a non-standard way to allocate this memory. Some programs (such as high-loaders or diagnostic software) also use non-standard ways to search or allocate memory.

B - Back to entry.
The program seems to execute some code, and after that jumps back to the entry-point of the program. Normally this results in an endless loop, except when the program also modifies some of its instructions. This is quite common behavior for computer viruses. In combination with any other flag, TbScan reports a virus.

C - File has been changed
This warning appears only if you use TbSetup to generate the ANTI-VIR.DAT files and means the file has been changed. Upgrading the software would

trigger this message. Otherwise, it is very likely that a virus infected the file!
NOTE:
TbScan does not display this warning if only some internal configuration area of the file changes. This warning means that code at the program entry point, the entry-point itself, and/or the file size has been changed.

D - Direct disk access
This flag appears if the program being processed has instructions near the entry-point to write to a disk directly. It is quite normal that some disk related utilities trigger this flag. If several files that should not be writing directly to the disk trigger this flag, your system might be infected by an unknown virus.
NOTE:
A program that accesses the disk directly does not always have the "D" flag. Only when the direct disk instructions are near the program entry point does TbScan report it. If a virus is at fault, the harmful instructions are always near the entry point, so it is only there that TbScan looks for them.

E - Flexible Entry-point
This flag indicates that the program starts with a routine that determines its location within the program file. This is rather suspicious because sound programs have a fixed entry-point so they do not have to determine this location. For viruses, however, this is quite common. Approximately 50% of the known viruses trigger this flag.

F - Suspicious file access
TbScan has found instruction sequences common to infection schemes that viruses use. This flag appears with those programs that are able to create or modify existing files.

G - Garbage instructions.
The program contains code that seems to have no purpose other than encryption or avoiding recognition by virus scanners. In most cases there won't be any other flag since the file is encrypted and the instructions are hidden.
NOTE:
This flag appears occasionally on "normal" files. This simply indicates, however, that these are poorly designed, not infected..

J - Suspicious jump construct.
The program did not start at the program entry point. The code has either jumped at least twice before reaching the final startup code, or the program jumped using an indirect operand. Sound programs should not

display this kind of strange behavior. If several files trigger this flag, you should investigate your system thoroughly.

K - Unusual stack.
The EXE file being processed has an odd (instead of even) stack offset or a suspicious stack segment. Many viruses are quite buggy by setting up an illegal stack value.

L - Program load trap
The program might trap the execution of other software. If the file also triggers the "M" flag (memory resident code), it is very likely that the file is a resident program that determines when another program executes. Many viruses trap the program load and use it to infect the program. Some anti-virus utilities also trap the program load.

M - Memory resident code.
TbScan has found instruction sequences that could cause the program to hook into important interrupts. Many TSR (Terminate and Stay Resident) programs trigger this flag because hooking into interrupts is part of their usual behavior. If several non-TSR programs trigger this warning flag, however, you should be suspicious. It is likely that a virus that remains resident in memory infected your files.
NOTE:
This warning does not appear with all true TSR programs, nor can you always rely upon TSR detection in non-TSR programs.

N - Wrong name extension.
Indicates a name conflict; that is, the program carries the extension .EXE but appears to be an ordinary .COM file, or it has the extension .COM but the internal layout of an .EXE file. A wrong name extension might in some cases indicate a virus, but in most cases it does not.

O - code Overwrite.
This flag appears if TbScan detects that the program overwrites some of its instructions. However, it does not seem to have a complete (de)cryptor routine.

R - Suspicious relocator
Indicates a suspicious relocator. A relocator is a sequence of instructions that changes the proportion of CS:IP. Viruses often use this. Those viruses have to relocate the CS:IP proportion because they were compiled for a specific location in the executable file; a virus that infects another program can hardly ever use its original location in the file as it is appended to this file. Sound programs know their location in the executable file, so they don t have to relocate

themselves. On systems that operate normally, only a small percentage of the programs should trigger this flag.

S - Search for executables
The program searches for *.COM or *.EXE files. This by itself does not indicate a virus, but it is an ingredient of most viruses, since they have to search for suitable files to spread themselves. If accompanied by other flags, TbScan assumes the file is infected by a virus.

T - Invalid timestamp.
The timestamp of the program is invalid; that is, the number of seconds in the time stamp is illegal, or the date is illegal or later than the year 2000. This is suspicious because many viruses set the time stamp to an illegal value (such as 62 seconds) to mark that they already infected the file so they won't infect a file a second time. It is possible that the program being checked is contaminated with a virus that is still unknown, especially if several files on your system have an invalid time stamp. If only very few programs have an invalid time stamp, you d better correct it and scan frequently to check that the time stamp of the files remains valid.

U - Undocumented system call.
The program uses unknown DOS calls or interrupts. These unknown calls can be issued to invoke undocumented DOS features, or to communicate with an unknown driver in memory. Since many viruses use undocumented DOS features, or communicate with memory resident parts of a previously loaded instance of the virus, a program is suspicious if it performs unknown or undocumented communications. This does not necessarily indicate a virus, however, since some tricky programs also use undocumented features.

V - Validated program
The program has been validated to avoid false alarms. The design of this program would normally cause a false alarm by the heuristic scan mode of TbScan, or this program might change frequently, and TbScan excludes the file from integrity checking. Either TbSetup (automatically) or by TbScan (manually) stores these exclusions in the ANTI-VIR.DAT.

Y - Invalid boot sector.
The boot sector is not completely according to the IBM defined boot sector format. It is possible that the boot sector contains a virus or has been corrupted.

Z - EXE/COM determinator.
The program seems to check whether a file is a COM or EXE type program. Infecting a COM file is a process that is not similar to infecting an EXE

file, which implies that viruses able to infect both program types should also be able to distinguish between them. There are, of course, innocent programs that need to find out whether a file is a COM or EXE file. Executable file compressors, EXE2COM, converters, debuggers, and high-loaders are examples of programs that might contain a routine to distinguish between EXE and COM files.

Appendix C: Solving Incompatibility Problems

Although TBAV utilities cooperate very well with other resident software, other software might not behave so well. This can cause system errors or even more serious problems. This section describes some common problems and their solutions.

PROBLEM:

If a TBAV utility tries to display a message, the text message "file <filename> could not be opened" appears.

Specify the FULL path and filename of the file to use as a message file after the TbDriver loading command. The default file name is TBDRIVER.LNG.

PROBLEM:

One of your utilities is loading a TSR into memory without an executable filename extension, such as .EXE or .COM. Since TbSetup creates ANTI-VIR.DAT records only for files with an executable extension, there is no ANTI-VIR.DAT, so TbMem is not able to record the TSR permission information.

Run TbSetup and specify the exact filename of the TSR. TbSetup creates an ANTI-VIR.DAT record, regardless of the filename extension, so TbMem can now record its information.

Although the ANTI-VIR.DAT record exists, TbScan does not use it to check the CRC to avoid false alarms.

PROBLEM:

You are running a network, and one of the following problems arises:

1. TbScanX is installed, but does not display the *scanning* message while accessing files. It also does not detect viruses.

2. TbCheck is installed, but does not display the *checking* message while accessing files. It also does not detect viruses.
3. TbFile is installed, but does not detect anything.
4. TbMem is installed, but does not detect TSRs.

Use the "TbDriver net" command after the network loads.

PROBLEM:

The system sometimes hangs when the message *scanning* is on the screen.

Try TbScanX without the EMS or XMS option. If TbScanX now works without any problems, add the EMS or XMS option again along with the COMPAT option. On some systems, you cannot use the TbScanX XMS option at all because these systems do not allow resident software to use extended memory.

If the problem relates to the XMS option and still occurs when you use the COMPAT option, you can use the XMSSEG = <VALUE> option to change the XMS swap segment address. The value should be between 2000 and 8000. The default value is 4000.

PROBLEM:

After you have given permission for a program to remain resident in memory, TbMem asks the same question the next time.

First, the SECURE option of TbDriver is in use. Remove this option, reboot and try again.

Second, the program mentioned does not appear in the ANTI-VIR.DAT file and, therefore, TbMem cannot permanently store the permission flag. Use TbSetup first to generate this program's ANTI-VIR.DAT record.

Third, for some reason it is not possible to write to the Anti-Vir.Dat file. The file might reside on a write protected diskette, on a network in a read-only directory, or the Anti-Vir.Dat file has the read-only attribute set.

PROBLEM:

The system sometimes hangs when you answer "YES" (abort program) to a TbMem message.

A solution here is difficult. Some resident programs seriously interfere with the system, and once rejected from memory, the system becomes unstable.

PROBLEM:

When you load TbDisk from the DOS command prompt, everything works fine. When you install TbDisk from within the CONFIG.SYS or AUTOEXEC.BAT file, however, it continually warns that programs write to disk directly.

Load TbDisk at the end of your AUTOEXEC.BAT file.

PROBLEM:

You formatted the hard disk using DOS FORMAT, but TbDisk did not display a message until the process was almost complete.

This is not a problem. A high level format program such as DOS's FORMAT.COM does not actually format the disk (that is, divide the disk into tracks and sectors), rather it reads all tracks to locate possible bad spots and clears the FAT and directory structure. Only this last step implies a disk write, so it is the only one TbDisk detects.

PROBLEM:

After you give permission for a program to perform direct disk access, TbDisk asks the same question the next time.

First, the SECURE option of TbDriver is in use. Remove this option, reboot and try again.

Second, the program mentioned does not appear in the ANTI-VIR.DAT file and therefore TbDisk can not permanently store the permission flag. Use TbSetup first to generate this program's ANTI-VIR.DAT record.

PROBLEM:

If you try to use Windows fast 32-bit disk access, Windows displays an error message.

Use the WIN32 option on the TbDisk command line.

Appendix D: TBAV Exit Codes and Batch Files

All TBAV utilities return to DOS with an error code that you can use with DOS's ERRORLEVEL command. The chief use of these error codes is in batch files. This appendix lists these error codes. Consult your DOS manual for information how to use error codes in batch files.

D.1 TbScan Exit Codes

TbScan terminates with one of the following exit codes:

Errorlevel	Description
0	No viruses found/ No error occurred
1	No files found
2	An error occurred
3	Files have changed
4	Virus found using heuristic analysis
5	Virus found using signature scanning
255	Sanity check failed

D.2 TbUtil Exit Codes

TbUtil terminates with one of the following exit codes:

Errorlevel	Description
0	No error occurred
1	Option "compare" failed/An error occurred

D.3 General Exit Codes

All the TBAV utilities except TbScan and TbUtil (see above) exit with one of the following exit codes:

Errorlevel	Description
0	No error occurred
1	A error occurred

D.4 Program Installation Check

To detect within a batch file whether a resident TBAV utility loaded, you can check for the device names. All TBAV utilities install a device name, whether they load from CONFIG.SYS or AUTOEXEC.BAT.

You can use the DOS IF EXIST batch file command to check for the device names. The following example, illustrating a part of a batch file, uses this construction to test whether TbScanX is loaded:

```
@ECHO OFF
IF NOT EXIST SCANX ECHO TBSCANX HAS NOT BEEN LOADED!
```

You could also branch to a label by using the GOTO command:

```
@ECHO OFF
IF NOT EXIST SCANX GOTO NOSCANX
ECHO TBSCANX EXISTS !
GOTO END
:NOSCANX
ECHO TBSCANX DOES NOT EXIST !
:END
```

Finally, the following table lists the device names used by the TBAV utilities:

TBAV program	Device name
TbScanX	SCANX
TbCheck	TBCHKXXX
TbMem	TBMEMXXX
TbFile	TBFILXXX
TbDisk	TBDSKXXX
TbLog	TBLOGXXX

Appendix E: Virus Detection and Naming

E.1 How Many Viruses Does TbScan Detect?

Most of the TbScan signatures are family signatures; that is, one signature detects an entire set of viruses. All these viruses relate to one another. The Jerusalem signature, for example, covers more than 100 viruses. For this reason, there is no way of knowing how many viruses TbScan detects.

Some competitive products treat each virus mutant as a separate virus, thus claiming to detect over 4000 viruses. TbScan, however, detects viruses using only 2000 signatures. If you want to compare virus scanners, you have to rely on the tests frequently published in magazines.

E.2 The Virus Naming Convention

TbScan follows the CARO virus naming recommendations. CARO is an organization in which leading anti-virus researchers participate. The CARO approach groups viruses in a hierarchical tree, which indicates to which family viruses belong. TbScan shows the complete CARO name where possible.

In contrast, however, many other anti-virus products simply indicate the family name or the member name. For example, many products might refer to the Leprosy.Seneca.493 using the family name Leprosy or member name Seneca, or even by the variant name 493. Worse yet, anti-virus products developed by non CARO members might even use a completely different name.

TbScan, however, tries to display as much of the name as possible. Building on the previous example, if TbScan can t distinguish between the Leprosy.Seneca.493 and Leprosy.Seneca.517 viruses, it indicates both by the name Leprosy.Seneca

Some viruses mutate themselves frequently. To detect all instances of such a virus, it is sometimes necessary to use multiple signatures. Although these signatures cover exactly the same virus, they do have a slightly different indication. Behind the name of the virus you will see a number in angle brackets. This number has nothing to do with the name of the virus, but is there just for maintenance reasons.

Index