AntiVirus Advanced SeT
version 7.7

Graphic design and typesetting by Eduard Kučera, ALWIL Software
The document was processed on PageMaker 6.5 CZ

Table of Contents

AVAST!

Introduction

Congratulations on your purchase of the AVAST! set of anti–virus programs. You have before you the latest version of this popular program, the 7.50 version. This Introduction explains the basic features of the software and the terms used to guide you throughout this handbook. Later chapters will acquaint you with the installation procedure, functions and control of the individual programmes and with the characteristics of some viruses with which the user is most likely to be afflicted. You will also learn what to do when you discover a virus in your system.

Although AVAST! offers full protection against viruses, we sincerely hope that you never meet with any of them. But in case you should, and our software proves ineffective against any particular one, please let us know immediately, either in writing or by telephone. In this way we shall be able to help you and to continually upgrade our programs as new viruses appear.

Since computer viruses represent the most dynamic area in today's world of computer technology, it is important to use the latest upgrades of the antivirus products. This means that the user must always be on the lookout for what is new in the field. To facilitate this task we have set up an antivirus service called **AVS**. Under this service you will **automatically** receive the latest versions of the AVAST! software for a period of one year, along with technical support and consulting services and last but not least you will receive a magazine called Screen, which is published once per three months and contains a great deal of important information. Keeping current is also facilitated by our money–saving BBS service and our Internet WWW home page. We will be happy to provide you with further information about these services.

New Features of AVAST! 7.7

General changes

Program VGUARD is no longer supported in the new version.

Program CHKAVAST has a lot of new features – see below.

Programs LGUARD and AGUARD have multiple–page help.

Changes in LGUARD program

Many new viruses can be detected now. The internal structure of VPS file has been changed, the new version of it is not compatible with the previous one.

Our scanning programs are now able to detect **macro viruses** more reliably. They contain the tool which is able to analyze OLE2 files, Word documents and macros. There is no need to test the whole file content anymore. Macro viruses are detected by two different methods: by exact identification, based on CRC of virus macros and by search strings which can detect the presence of new viruses from known virus families.

DOS program LGUARD can even remove the virus macros from the infected document, which can be used without any problems afterwards. To delete macros you can use the same parameter as to delete files infected by clasic file virus. If the macrovirus is detected, you will have the possibility to delete just the macros of particular virus (in the case of exact identification), all macros or the whole document.

You can specify this action also using the additional parameter with **/Z switch**.

Parameter **/ZM** means delete just the virus macros, **/ZA** means delete all macros and **/ZF** means delete the whole file. In this case the program will ask you to give Yes or No reply. These parameters can be used together with /P (no pauses) too. The default action with /Z only deletes virus macros for exactly identified viruses and all macros for virus families. If no virus is found, program LGUARD displays the information when the file contains some macros, when it is the Word template, Word document or OLE2 file.

Program LGUARD now has faster algorithm for virus detection.

Program LGUARD now displays the amount of data verified and read in percents while loading the VPS file (this is specially important when the program is loaded from floppy).

Program LGUARD can handle several characters from virus name while displaying the set of known viruses (parameter /V). It moves through the list accordingly.

Program LGUARD now recognizes the special virus category – viruses which are in the wild (ITW). Such viruses have character W in the list of known viruses and have special warning when they are detected and virus description is displayed (via the F1 key). Macroviruses have the character M in the list of known viruses.

Program LGUARD now displays the total number of known macroviruses and viruses which are in the wild in the summary after list of known viruses.

Program LGUARD is able to handle macroviruses (M) and ITW viruses (W) in the user–defined strings.

Program LGUARD displays in the list of known viruses duplicate virus names (in red colour) and user–defined viruses (in yellow colour).

Changes in AGUARD program

Program AGUARD has the newly defined Hotkey while displaying the list of changed programs. Pressing Ctrl–E moves the cursor to the next directory with some changes. The description of all Hotkeys could be displayed by pressing the F1 key.

Program AGUARD supports the new parameter /X to not test the system areas of the disk. This could be very useful when you test the integrity of electronically transferred files on different computers (typically desktop and notebook).

Changes in RGUARD and BGUARD programs

Program RGUARD displays the character W for virues, which were found ITW.

Program RGUARD tests the presence of floppy disk in drive A, when you press Ctrl–Alt–Del and does not allow the reboot until the diskette is removed. This function could be switched off by parameter /R (on computers without disk drives or on computers having floppy boot disabled).

Program RGUARD requires the XMS memory for boot virus tests.

Program BGUARD has new parameters (entered in non–capital letters). Use them very carefully however!!:

To create the new boot sector of floppy disk:

```
bguard clean <disk> <size>
```

disk=A:, B:, size=360, 720, 1.2, 1.44, 2.88

To save the system are of the hard disk to floppy:

```
bguard save <disk> "description"
```

description could contain the info about the computer. It must be in quotations!

To restore the system area of the hard disk:

```
bguard restore <disk>
```

To compare the system area of the hard disk with previously saved info:

```
bguard compare <disk>
```

When those parameters are used, there is no output on the screen and everything is reported via the return codes!

Defined return code values:

1	data resored
2	data saved
3	nothing done
4	error while working with floppy disk
5	error while working with hard disk
6	compare is OK
7	compare is not OK
99	help

Changes in CHKAVAST program

Program CHKAVAST has many new parameters:

```
CHKAVAST ISTIME hh:mm
```

Serves to determine the system time; The return code is set to 1, if the system time is ahead the given time and to 0 if the time is behind or equal.

```
CHKAVAST ISDAY [NEW] [OLD] [<day>]
     [<date>]
```

Serves to determine the system day and date settings and to branch commands in batch files if the condition match. The return code is set to 0 if the condition matches and to 1 if it does not match. The NEW parameter determines whether the computer has been booted for the first time this day, and OLD has the opposite meaning.

In addition to these parameters, the days of week (MON, TUE, etc.) and dates (a pair of two–figure numbers denoting day and month) may be selected. If either of these numbers equals zero, they denote any value, so that for example 00–08 means any day in August and 15–00 means the 15th day of any month. The form of entering the date must conform to the current country code setting (European or US) Parameters could be combined; the final result will be correct only if all the conditions for each match.

Example:

```
chkavast isday mon new
if errorlevel 1 goto label1
echo At beginning of week do all the tests!!
lguard c:\*.* /s /e*
:label1
... isday 13-00 fri
if errorlevel 1 goto label2
pause Today is Friday the 13th. Good luck!
:label2
```

```
CHKAVAST ISMONTH [NEW] [OLD]
```

Serves to determine whether the computer has been booted for the first time in the month. The return code is set to 0 if the condition matches and to 1 if it does not match. The NEW parameter determines whether the computer has been booted for the first time in the month, and OLD has the opposite meaning.

Example:

```
chkavast ismonth new
if errorlevel 1 goto label1
echo Let's make month backup!!
:label1 ...
```

```
CHKAVAST ISWEEK [NEW] [OLD]
```

Serves to determine whether the computer has been booted for the first time in the week. The return code is set to 0 if the condition matches and to 1 if it does not match. The NEW parameter determines whether the computer has been booted for the first time in the week, and OLD has the opposite meaning.

Example (the same as for ISDAY but better, because it works fine even if the computer was not turned on on Monday).

```
chkavast isweek new
if errorlevel 1 goto label1
echo At the beginning of week do all the tests!!
  lguard c:\*.* /s /e*
:label1 ...
```

```
CHKAVAST ISYEAR [NEW] [OLD]
```

Serves to determine whether the computer has been booted for the first time in the year. The return code is set to 0 if the condition matches and to 1 if it does not match. The NEW parameter determines whether the computer has been booted for the first time in the year, and OLD has the opposite meaning.

Example:

```
isyear new
if errorlevel 1 goto label1
```

```
echo Happy New Year!!
:label1 ...
```

CHKAVAST uses the file AVAST.DAT which is located in the same directory as the program itself. It uses this file to determine the correct answer for NEW and OLD commands. Parameters ISDAY, ISWEEK, ISMONTH and ISYEAR are quite independent; for example on Monday morning both ISDAY NEW and ISWEEK NEW are true.

Typefaces Used in the Handbook

This user's handbook uses apart from the normal text typeset in Century Schoolbook font another typeface, which serves to distinguish the presented information:

```
FGUARD /H
```

Examples are printed in Courier. They may be entered in the computer either in upper or lowercase. If a printed command line is longer than one line of text, another line is indented and it is to be entered into the computer without the carriage return, and the Enter key is used only after completion of the command.

```
SGUARD [/K] [/L] [/N] [/D] file1 sum1 [file2 sum2
   [...]]
```

General specifications for running the program are printed in Courier too. Information appearing between square brackets is optional, the rest is necessary. [...] indicates repetition of the preceding parameter. For example:

```
SGUARD
```

is incorrect (file sum parameters which are necessary are missing).

```
SGUARD IO.SYS 12345
```

is correct (the parameters follow the command).

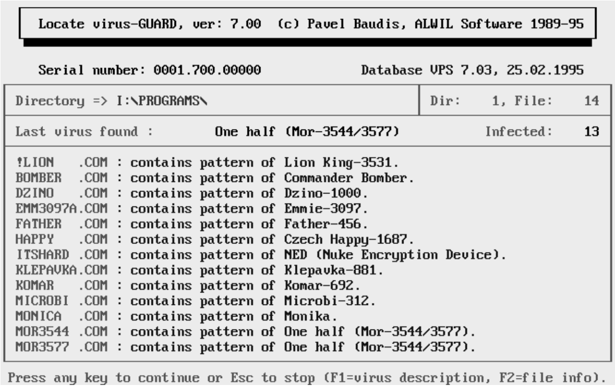
```
SGUARD /L IO.SYS.12345 COMMAND.COM 23456
```

is also correct.

```
SGUARD /X IO.SYS 12345
```

is incorrect (/X switch is unknown).

The contents of displays are shown as pictures no matter whether the programs dealt with run under MS–DOS or Windows. For example:



```
 Locate virus-GUARD, ver: 7.00  (c) Pavel Baudis, ALWIL Software 1989-95

 Serial number: 0001.700.00000            Database VPS 7.03, 25.02.1995

Directory => I:\PROGRAMS\                      Dir:     1, File:     14

Last virus found :        One half (Mor-3544/3577)        Infected:    13

!LION   .COM : contains pattern of Lion King-3531.
BOMBER  .COM : contains pattern of Commander Bomber.
DZINO   .COM : contains pattern of Dzino-1000.
EMM3097A.COM : contains pattern of Emmie-3097.
FATHER  .COM : contains pattern of Father-456.
HAPPY   .COM : contains pattern of Czech Happy-1687.
ITSHARD .COM : contains pattern of NED (Nuke Encryption Device).
KLEPAVKA.COM : contains pattern of Klepavka-881.
KOMAR   .COM : contains pattern of Komar-692.
MICROBI .COM : contains pattern of Microbi-312.
MONICA  .COM : contains pattern of Monika.
MOR3544 .COM : contains pattern of One half (Mor-3544/3577).
MOR3577 .COM : contains pattern of One half (Mor-3544/3577).

Press any key to continue or Esc to stop (F1=virus description, F2=file info).
```

The content of the MS–DOS screens is typeset in special typeface, which is the exact copy of the characters displayed on the screen of the computer equipped with VGA adapter. The MS–DOS screens are printed in inversion, as this environment normally uses a light typeface on a dark background. That is why the windows shades are light.

Installation of AVAST! Software

AVAST! software is intended for personal computers compatible with IBM PC, IBM PC/XT and IBM PC/AT, with at least 256 KB RAM. It requires the MS–DOS or PC–DOS operating system, versions 3.30 and higher, and works with all standard display adapters (MDA, CGA, EGA, VGA, EGC, Hercules Graphics Card, etc.).

AVAST! software is normally delivered on Distribution Diskettes, which are write–protected. You should **first boot the computer from the original system diskette** which you received with the computer. Then you should make copies of the distribution diskette(s), using the following command (which must be repeated for each diskette):

```
DISKCOPY A: A:
```

First insert the distribution diskette (Source), then your blank diskette (Target). Then follow the instructions you receive from the DISKCOPY program. After copying the original Distribution Diskettes, store them in a safe place, and use your copies for installation. **These copies should be write–protected!**

Installation for MS–DOS

The installation itself is very simple. It is done by the INSTALL program on the distribution diskette. This program first tests your floppy disk drive to determine whether it can write to write–protected diskettes. Then it checks your computer for any known viruses. Finally it copies all necessary files to a directory chosen by you. the program is started with the following commands:

```
A:
INSTALL
```

After starting the program, just follow the simple instructions. As mentioned above, the program first tests whether your floppy disk drive can write to a write–protected diskette. If so, the drive is faulty and must be repaired before proceeding further. This is quite rare, however we have only encountered three such cases in our experience. On the other hand, this fault can be an important factor in spreading computer viruses. Because of this test, you must be sure that, like the originals, your copies of the distribution diskettes **are write–protected**.

AVAST! also includes programs operating under Windows. In order to install these, you need to run the program called AVINST in Windows environment to execute the proper installation.

The INSTALL program also creates modified initializing command information (AUTOEXEC.BAT or START.BCH). This new information is placed in the main directory of the C: drive and is called AVAST!.NEW. If you wish to use this file, you will first have to rename it.

Besides installing on the hard disk, you should also create a **special rescue diskette** containing tools for restoring the system in case of a virus infection. The following steps are recommended:

1. Boot the system with the original system diskette that came with your computer.
2. Prepare in advance a system diskette with at least a 720 KB capacity, using the FORMAT /S command.
3. Start the INSTALL program. On the first line type the drive in which the diskette is located and on the second type the directory in which the AVAST! software is located. Choose the option CREATING A RESCUE DISKETTE and follow the instructions of the program.
4. Write–protect the diskette, label it, and test it to **be sure that it works**. Then keep it in a safe place. It may prove very useful in future!

Installation for Windows

The AVINST.EXE installation program is designed to complete the AVAST! system installation for Windows. It is a very simple program, the task of which is to set the Windows perimeters correctly so that the user may immediately use all antivirus programs that are a part of this product.

Another very important purpose of the program is the option of deinstallation of the AVAST! product in such a way, so that no trace of it is left in the configuration files of the Windows system. If you decide to deinstall the AVAST! system, use the AVINST program as well, thus you will enable it to distinguish which operation you require.

Installation of the AVAST! for Windows system unfortunately does not occur flawlessly on some computers. The problems mainly apply to installation of SGW and FGW programs into the start–up group of Windows. If upon running the AVINST program you do not find these two programs in the Start–up group, copy them manually from the created AVAST! group. After manual copying of the SGW program, it is necessary to adjust its features in such a way so that it is run from the Windows Start–up group in a minimized format.

How and When to Use the Programs

The general antivirus program AGUARD is designed to be started up periodically upon the system preventive maintenance. It is important to use it regularly since it is an integral part of the protection system. It is able to remove viruses from infected files, but only if it is used routinely that means regularly!!!. The same application method is used for AGUARD program for Windows – AGW.

The search program LGUARD should be run whenever new programs are added. It can also be used periodically to verify that the computer is free from infection. LGUARD with the /M parameter (memory and boot sector test only) should be run whenever the system is booted (AUTOEXEC.BAT).

In addition we recommend that the LGUARD program be used periodically to test all diskettes which may conceivably contain viruses (unfamiliar diskettes, those that have been used on other computers, those coming from other companies, and so forth).

The LGUARD for Windows program may be placed in the Start–up group, so that it will be run during each Windows start–up. It can work in the background, periodically testing the computer's disks. The user will be unaware of its activity unless a virus is discovered.

The memory resident program RGUARD is designed for testing boot sectors of floppy disks and also for testing all running programs. It does not discover any more information than LGUARD, but it continually tests for the presence of viruses, so that the user need not remind himself to run the LGUARD program.

The general antivirus programs SGUARD and FGUARD should never be lacking in the AUTOEXEC.BAT file, so that they may detect the presence of viruses in time.

The SGUARD for Windows program (SGW) should be placed into the Start–up group and designate which files it should test upon each start–up of Windows (for instance the DOS system files and important applications).

As mentioned above, the BGUARD program should be part of the „rescue“ diskette that should be created for each computer. Aside from that, it is useful for eliminating boot viruses from infected diskettes. It can also be used as protection for all data diskettes.

The VGUARD program should be used to eliminate the most widespread kinds of viruses.

With the proper use of AVAST! software you can protect your system against attack by computer viruses and thereby avoid the great damage that can be caused by these pests.

Notice to Users of Earlier Versions

The new version of the AGUARD program is not compatible with earlier versions of the database AGUARD.DAT. Therefore we recommend that you check all file contents one last time with the old AGUARD program before running the new AGUARD program. If this new AGUARD program detects the non–compatible AGUARD.DAT database, it will offer to erase it and create a new database. The SGUARD program may, under certain conditions, report a change in the system area after the change from the older system. This is due to the fact that it no longer tests the 15 bytes in which the volume label and series number are saved: thus the change in the volume label will have caused a ,,false alarm“.

The LGUARD program, versions 4.3 and above, is of the EXE type. The LGUARD.COM file of the earlier version will be automatically deleted when you use the INSTALL program to install AVAST! in the same directory in which the older version is found. If you merely copy files from the distribution diskette (which is definitely not recommended), or if you install them in a different directory, you will have to delete the LGUARD.COM file by yourself. The earlier version can be recognized by its number and also by the fact that the version number of the VPS file will be absent from the upper right–hand corner of the screen. Deletion is absolutely necessary because when running the programs, the older COM extension will take precedence over the newer EXE extension.

This page is intentionally left blank.

The Problem of Computer Viruses

In the middle of the 1980's a great danger for computer users appeared: the computer virus. Since that time they have multiplied rapidly, and the number of known types is constantly growing along with the damage that they can do.

Many false ideas about computer viruses persist – for example that they represent a form of life, that they possess some form of artificial intelligence, that they may be spread among computers without physical contact, that they may survive in electrical circuits that have been turned off, or in the CMOS memory, and so forth.

What, then, are computer viruses in reality? They are programs that are able to attack other programs by copying themselves onto those programs without the direct action or even the knowledge of the user. In this way the infected programs themselves become carriers of the viruses, helping them to spread still further.

How do computer viruses spread? Viruses can attack any code executed in the computer. In practice, it appears that they attack the so–called Disk System Area (Disk Partition Table Sector or Boot Sector), or files containing programs.

What do computer viruses do? In addition to modifying programs in the course of their spread (which in itself represents a serious danger for programs being run), viruses can cause other kinds of serious damage. They take up space in the memory and on the disk; they slow the computer down; they destroy, modify or damage files or disk system areas; they can reformat or overwrite disks or diskettes; they can display various messages on the screen or redefine the keyboard. They are created by people for fun, out of malice or revenge, or with the express purpose of destroying programs and data or damaging computers. Generally speaking, there can be no harmless virus. All of them interfere with the host's program code, and certainly none of them is to be invited or welcomed into the system.

Protection against computer viruses can be highly uncer–tain because viruses come in many forms and operate in many different ways. Their number has multiplied in recent years. Nevertheless there are methods of detecting viruses in time to avoid the dangers that they pose. Detection can be carried out in a number of ways.

If a particular virus is known in advance, it is possible to write a special program against it. But such programs, of course, will be ineffective against other types of viruses.

Another type of antivirus program is the ,,locating" or ,,scan" program. These operate with knowledge of certain characteristics of the virus but without any detailed information—they may be able to identify certain aspects of its code, for example. During their running they search all specified files to test for the presence of a given sequence of bytes. If they find such a sequence, they suspect a virus. Whether this is in fact the case depends on the number of "suspicious" sequences that they find. The limitations of this type of program are due on the one hand to the increasing numbers of new virus strains, which necessitates frequent and constant updating of the program, and on the other to a recent increase of so–called polymorphic viruses, which vary in each of their copies, making it impossible to construct a model of them by any simple means.

A great contribution may be made by general antivirus pro–grams which are also effective against unknown virus types. An example of such a program is one which stores compressed information about the other files and is capable of ascertaining any changes to them and also of restoring them to their origi–nal state if necessary. Another example is a program able to detect modifications in programs over time and requires user approval before any hazardous operations can be carried out.

The best protection against computer viruses of course con–tinues to be **prevention**. It may not always be a hundred per–cent effective, but here as elsewhere it is true that those least prepared are most at risk. Effective prevention consists in the rigorous observance of the following rules:

– Never run programs of unknown origin.

– Write–protect diskettes. Cover the write–protect notch on original distribution diskettes prior to installing the programs. Copy them and store them in a safe place. Do not use the originals again unless necessary.

– Leave diskettes in disk drives only for the time that you are using them. Remove them immediately after finishing work. Whenever there is nothing more you wish to save on them, write–protect them.

– Reconsider your attitude toward illegally obtained software. Who knows whose hands it has been through before it comes to you? Remember also that the use of such software is illegal.

– **Back up your data and programs regularly and frequently**. This is a good idea even if no viruses threaten your system.

– Employ all available means of antivirus protection, both general and specific programs.

– Do not let other people work on your computer. Employ programs limiting computer access to authorized personnel.

Remember that each obstacle set in the way of computer viruses reduces their chance of infecting protected systems

Computer viruses first appeared in 1987. Their spread has been hastened by the rapid expansion in the use of personal computers since then, with programs being exchanged among users who are often unaware of the risks and dangers presented by viruses. To some extent their spread has been aided by distribution of programs by unauthorized suppliers and the exchange of illegally obtained programs.

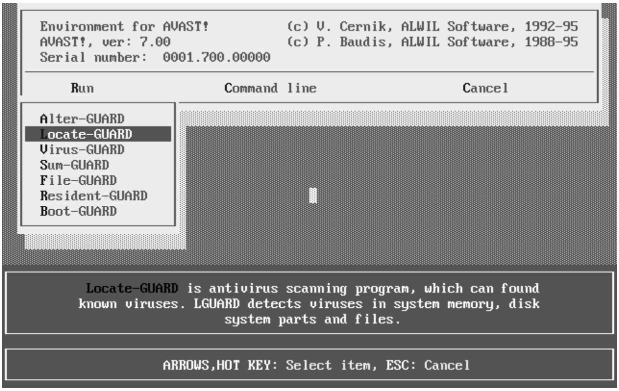
The AVAST! package of antivirus programs is the first to be available in Czechoslovakia. From the beginning these programs have been constantly updated in line with the latest advances in this highly dynamic field. In this effort we have cooperated with many international experts. The most recent version contains a number of independent antivirus programs, so that the whole is quite complex.

One of these programs is a specific antivirus program aimed at the viruses most widespread. Then there are detection programs for DOS and Windows which are capable of finding many more virus strains, and memory resident programs with the same purpose. There follow four programs which are general, so that they may be used against so far unknown computer viruses. General programs cannot be a hundred percent effective in all circumstances, but if used correctly they can detect the presence of most viruses and thereby prevent enormous damage. Moreover they are important from a methodological point of view because they demonstrate the tools that can be used to protect against viruses. Neither the general nor the detection programs can ascertain the structure of a virus, and therefore they are unable to eliminate it. But they do give ample warning of the presence of the virus. Further action is up to the user. In any case, these subsequent steps must be carried out by someone acquainted with the system and aware of the potential effects!

The AVAST! Integrated Environment

The individual components of the AVAST! system are designed so that they can be used independently for specific tasks. The user can run them from the command line and specify parameters as described in the following chapters. In addition he can use the AVAST.EXE program to create an integrated environment in the DOS system. In this environment the parameters for all the programs can be set and the programs started. The user may, if he wishes, store the parameters in a special configuration file AVAST.CFG, from which they are read at the next program start.

The basic screen for the AVAST! program looks like this:



```
Environment for AVAST!          (c) V. Cernik, ALWIL Software, 1992-95
AVAST!, ver: 7.00               (c) P. Baudis, ALWIL Software, 1988-95
Serial number:  0001.700.00000
```

```
Locate-GUARD is antivirus scanning program, which can found
known viruses. LGUARD detects viruses in system memory, disk
system parts and files.

ARROWS,HOT KEY: Select item, ESC: Cancel
```

The AVAST! program uses a pull–down menu system controlled by the cursor keys, by pressing the highlighted letters, or using the mouse. The first menu is for running the programs, the second allows user input and modification of the command line. To facilitate modification of the command line, a detailed help for the given program appears on the screen explaining the use of all possible parameters. The FGUARD and RGUARD programs cannot be loaded and installed in the memory from this program, but it can set their parameters. These programs should be installed with the help of the AUTOEXEC.BAT when the computer is turned on.

Locating Antivirus Programs

These programs are intended for use against known virus strains. Since they are intended only to locate a virus, they need not know its particular means of spreading, its special features or its effects. All they need to know is one of its characteristics. This may be for example a certain instruction sequence or characteristic data appearing in each program infected by the virus. Thus the identification of a new virus does not require a complicated analysis, nor must the virus even be physically present. It is enough to know one of its characteristics. Thus it is clear that this type of program can react much more quickly to a new virus strain. On the other hand, this has led to a fierce and unproductive competition among firms developing antivirus programs and the users using them. Whether a given program is able to locate 2300 or 3500 virus strains is really beside the point, because the user will never meet with most of them anyway (for example, some are exotic viruses found only in Taiwan or Israel). Moreover the true number of viruses is difficult to determine, since some merely represent variants of already listed viruses. Therefore it is far more important that the program be easy to use routinely and that in case of infection it be capable of reacting effectively.

Advantages of this type of program include the relatively large number of strains that it is able to search for (at present between 3500 and 3800, numbers which are constantly increasing), and its ease of modification (adding characteristics of newly discovered virus strains). This is reinforced by the fact that the user himself may add further characteristics.

However, locating programs also have some disadvantages. The great increase in the number of characteristics leads to a disproportionate increase in the time needed for the search and in the effort spent on updating. This may prove a great drawback within the very near future, when the number of char–

acteristics will approach many thousands. A further problem is collecting verified characteristics of individual viruses. Once such information is exchanged and publicized, the danger arises that somebody might modify known viruses so that they cannot be detected by existing locating programs. Finally, locating programs are prone to giving false alarms by identifying a nonexistent virus in a program which in fact has not been infected.

Another weakness of the locating programs is that, since they use simple characteristics, they are unable to locate certain of the virus strains (polymorphic) that have appeared in recent years. This is because there is no characteristics to be chosen. These viruses are coded differently in each of their copies, and the part of them that serves to decode them (and is a necessary component) may be “randomized” so that determining their characteristics becomes impossible. In such cases the locating must be accomplished with an algorithm based on an analysis of the virus, which in itself defeats the purpose of the locating program.

In spite of all this, the locating programs can obviously be highly effective as prevention tools. AVAST! contains three locating programs, each of which has its particular function and its own place in the range of antivirus tools. The first is a classic locating program, the second is intended for use with Windows, and the third is memory resident, that is, capable of searching for viruses in real time.

All of our locating programs are able (as of 25th February 1996) to find around 5601 virus strains. The number increases each day, so that the version which you have obtained will probably be effective against other strains as well.

All three programs contain a special algorithm for seeking viruses which is able to search for all characteristics in parallel fashion. It is also able to handle tens of thousands of separate characteristics. Thus our programs are at present among the fastest locating programs in the world.

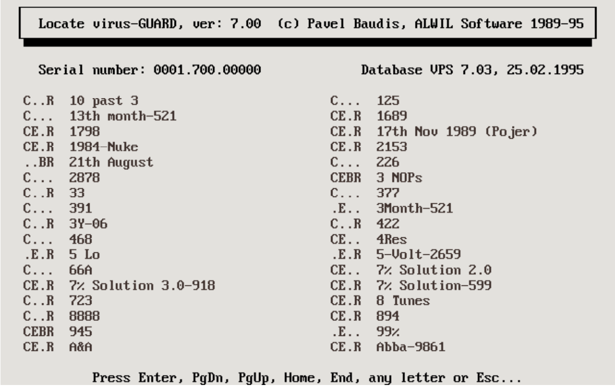
Locate–GUARD

Notice: Structure of LGUARD.VPS data file has been changed, so VPS version 7.7 is not compatible with previous version.

At this point in earlier versions of the handbook we presented a list of viruses which LGUARD is able to find. Today, however, this list has grown so long that it is impracticable to reproduce it here. In any case, such a list can never be really complete because the LGUARD program is also able to detect a whole series of variants and mutations of the basic virus types. Their exact number at any point in time is difficult to determine and anyway could serve little purpose apart from bombastic and misleading advertising. The up–to–date list of viruses can be saved in the VIRLIST.TXT file, if you run LGUARD program with /V and /R parameters.

The database of viruses' characteristics is located in the external file LGUARD.VPS. This enables us to keep the program updated merely by distributing the latest updates of this file. The number of the LGUARD.VPS version and the date of its creation are displayed in the upper right corner of the screen each time it is run.

The LGUARD program can never be "completed" as long as new virus strains continue to be discovered. When six months have elapsed since the creation of the LGUARD.VPS in use, the LGUARD program alerts the user so that he may consider updating. Our antivirus service includes about 12 updates per year. The present state of the program and a list of known viruses can be displayed with the /V option: the following figure shows the first page of the list:

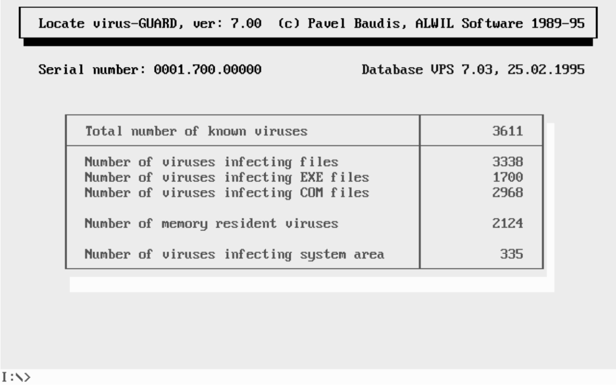


```
 Locate virus-GUARD, ver: 7.00  (c) Pavel Baudis, ALWIL Software 1989-95

  Serial number: 0001.700.00000             Database VPS 7.03, 25.02.1995

C..R  10 past 3                        C...  125
C...  13th month-521                   CE.R  1689
CE.R  1798                             CE.R  17th Nov 1989 (Pojer)
CE.R  1984-Nuke                        CE.R  2153
..BR  21th August                      C...  226
C...  2878                             CEBR  3 NOPs
C..R  33                               C...  377
C...  391                              .E..  3Month-521
C..R  3Y-06                            C..R  422
C...  468                              CE..  4Res
.E.R  5 Lo                             .E.R  5-Volt-2659
C...  66A                              CE..  7% Solution 2.0
CE.R  7% Solution 3.0-918              CE.R  7% Solution-599
C..R  723                              CE.R  8 Tunes
C..R  8888                             CE.R  894
CEBR  945                              .E..  99%
CE.R  A&A                              CE.R  Abba-9861

            Press Enter, PgDn, PgUp, Home, End, any letter or Esc...
```

The list also indicates the targets of the viruses. The letter “**C**” means that the virus infects files of the COM type, “**E**” of the EXE type, “**B**” that the virus infects the disk system area (disk partition table sector and/or boot sector), and “**R**” that the virus is resident in the operating memory.

The user may move through the individual listing screens using the keys PgUp, PgDn, Home and End. Upon pressing any letter, a screen listing the viruses commencing with this letter will appear.

At the end, the program displays the test results: the number of viruses checked for, the number found to be infect–ing the files, COM files and EXE files, and the number of vi–ruses in the system area and the number that are memory resident:



```
Locate virus-GUARD, ver: 7.00  (c) Pavel Baudis, ALWIL Software 1989-95

Serial number: 0001.700.00000            Database VPS 7.03, 25.02.1995

    Total number of known viruses                    3611

    Number of viruses infecting files                3338
    Number of viruses infecting EXE files            1700
    Number of viruses infecting COM files            2968

    Number of memory resident viruses                2124

    Number of viruses infecting system area           335

I:\>
```

On being started, the LGUARD program first searches the computer memory and checks for the viruses listed. If it locates a characteristic of any of the viruses in the memory, it reports the fact to the user. Should this occur, we strongly recommend that the computer be turned off and the system rebooted from the original write–protected system diskette (or a copy prepared in advance for this purpose). Then the original LGUARD program should be started from a copy of the AVAST! distribution diskette. Omitting this step could lead to all files tested by the LGUARD program being infected by the virus!

If LGUARD detects a virus in the operating memory, it suggests rebooting the system. Once the user agrees, he is prompted to insert the system diskette into the A: drive, and the system is rebooted. Nevertheless, we recommend that the system first be turned off, then turned on again, before rebooting.

After testing the memory, LGUARD searches the disk system area (boot sector and disk partition table) to check for the presence of so–called “boot viruses”. Such viruses do not infect files but spread through the system area. If LGUARD finds such a virus it alerts the user.

If both the memory and the disk system area are found to be virus free, LGUARD displays a report which remains on the screen for three seconds. This time can be shortened by pressing any key. Then LGUARD scans disk files for viruses. The user may specify the disk and directory to be scanned (the root directory of the current drive and all subdirectories by default). Additional disks may be specified at the same time. When testing a diskette, upon completion LGUARD asks the user if he wishes to test a further diskette and requests its insertion on receiving an affirmative response. In addition the user may request testing of file extensions of tested files (by default the extensions COM, EXE, SYS, OV? and BIN are tested). The LGUARD program always checks itself first, since there is always the danger that it may have become infected on start–up if not stored on a write–protected diskette. This is always done whether or not it is stored on the tested disk. File testing can be terminated at any time by pressing the Esc key. LGUARD then asks whether it should really end and does so upon receiving an affirmative response. If a password was given when running the program, it must be used correctly in order to end.

If LGUARD finds a file containing a virus characteristic, it reports the fact. The user may learn the basic characteristic by pressing the F1 key. The description contains information about the target attacked and indicates whether the virus is memory resident. By pressing the F2 key the basic fileinfo is displayed.

The virus description is solely informative, and the suspected infection must be confirmed! This is especially true of viruses created in high level programming languages such as Pascal and C, in which case it is difficult to determine a virus characteristic with precision. The Kamikaze virus is a good example of one likely to cause a false alarm. The program is continued by pressing any key. On completion the LGUARD program displays a table with information about the number of files tested and the viruses found.

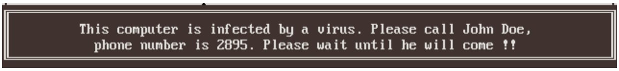
We have placed great emphasis on developing a quick and effective algorithm for seeking virus characteristics. As a result, our locating program is faster than most of our competitors' products. Nevertheless, the user may find the time required for a complete disk test too long, especially if the disk contains a large number of files. Therefore, taking into account the features of the great majority of known viruses, the program tests only the beginning and end of each program by default, in the sufficient length of 8192 bytes. However, by se–

lecting option /C the user may specify a complete search of the files. The whole content of files will be also automatically tested if any virus has already been detected.

LGUARD is able to recognize files compressed by programs such as PKLITE and LZEXE or Diet and by some linking programs such as linker. These files may be infected in two ways: "internally" (before compression) or "externally" (after compression). The LGUARD program is also able to test the content of files, which were compressed by some of the standard compression programmes such as Diet, Lzexe, Pklite, Ice and Shrink. This test occurs if the program was assigned to test the whole files (or if a virus was discovered before that). The presence of polymorphic viruses is not tested in the compressed files. The report of the internal test execution is displayed on the screen and written in the report file.

LGUARD can also recognize a number of polymorphic viruses. These are coded in such a way that they differ in each of their copies, so that no model can be constructed to aid in their detection. Thus they must be located by algorithms or by statistical methods. Examples of such viruses are Flip, Maltese Amoeba, Tremor, Slovakia and viruses created with MtE (Mutation Engine) or Slovak One half.

If LGUARD is started using the option **/W[password]**, a two–line centred text, defined by the user and stored in the LGUARD.MSG file, appears on the screen each time a virus is detected. If the /W option is selected, this file must be located in the same directory as the program LGUARD.EXE. The message may contain up to two lines of 76 characters each, and will ensure that even a computer beginner will take the correct action if a virus is detected:



```
This computer is infected by a virus. Please call John Doe,
 phone number is 2895. Please wait until he will come !!
```

If a password is included in the command line after the /W option, it must be given before the program can continue. This insures that an inexperienced user will proceed no further. The password must also be given to interrupt the program. This insures that the testing will continue to the end. If the character "+" is chosen as the password, the message is displayed on the screen immediately (as a message display test). An example of a LGUARD.MSG file is found on the distribution diskette. If you wish to use this option, do not forget to change the content of the message, so that the user does not in fact call "John Doe".

If the **/P** option is selected, the program runs continuously without pausing after testing the memory or detecting a virus. In this case, test results can be determined only by checking the exit code.

If the **/I** option is selected, the program upon testing the exchange media (floppies) does not ask, whether another medium is to be tested, but ends.

If the **/R** option is selected, a report file with a list of viruses detected and a statistical table is created at the end of the program. If a character "*" is placed after the /R parameter, the report file will also contain the files, in which no virus was detected.

If the **/Z** option is selected, the program offers to delete each file in which a virus is detected.

If the **/X** option is selected, the program offers to rename each file in which a virus is detected. The first extension character is after renaming the letter "V" (for instance, COM becomes VOM, EXE becomes VXE, etc.).

LGUARD contains a further interesting feature: the user may add more definitions of virus characteristics and so respond easily to the appearance of new strains. Using the **/F** option he may specify the file which contains these characteristics. Lines in this file beginning with the "*" character are considered comments and are ignored. A definition of a virus characteristic consists of three lines. The first contains the name of the virus (up to 29 characters); the second specifies the virus's targets (identified by the letters "C", "E", "B", and "R" as explained above); the third line contains the actual virus characteristic, given in the form of a hexadecimal string of characters without separators to a maximum of 32 bytes or 64 characters (each byte containing two characters). The description may also contain wildchars, meaning that in a given position a random character may be found. This is defined by the characters “??”. Wildchars may not be placed in the first two positions. Their too frequent use increases the risk of false alarms, so that they should be used sparingly. The following figure shows a description of a fictitious virus:

```
* ==============================
* this is a comment: sample definition of fictitious viruses
* the following virus attacks the system area of the disk and is resident:
Bluebeard
BR
8CC8EAD8EDA0BCBBF0FBA2367CA2097
*
* the following virus attacks COM files only and contains wildchars:
Alice
C
F687772A01081740F3??DB??FF4D01BC
*
* the following virus attacks COM and EXE files; it is memory resident.
HUHU
CER
FA8B8005BEB310172A0B74D0131C75F8
* ===============================
```

If the virus description is composed correctly, 16 bytes or 32 characters should be sufficient. From the point of view of the speed of the LGUARD algorithm, this is advantageous as long as the description does not begin with binary zero (“00”).

LGUARD also supports the Novell computer network. Network disks can be tested, and reports can be sent to designated users whenever viruses are discovered at the work station. This choice is specified by option **/U**, which may be followed by the name of the designated user or recipient. If no name is specified, reports are sent to the “Supervisor” user. Reports are sent only to authorized network users who have been specified as recipients. In addition, reports of virus detection are sent (with date and time) to the network system console as follows:

```
10.10. 16:12 SMITH[05]virus Anti-Telefonica (CSFR)
```

Reports are sent only during the first discovery of a virus during one running of the program. If LGUARD finds additional viruses, they are not reported. This function is designed to maximize the security of the network, whose operation may be threatened far more by viruses. The **/U** parameter may be replaced by defining the **AVASTMES** environment variable. This variable designates the Novell network user, to which the report of the discovered virus will be sent and serves to ensure more flexible reporting of the virus occurrence in the network. If defined, the /U parameter does not have to be set.

The report of virus detected on a given work station may also be sent to a user, who finds himself in a different context (applies to the Novell network, version 4.00 and higher). Such a user must be specified exactly in the same pattern, as it is listed by the NLIST program with parameters USER /A /B. The only exception is that the character "=" must be for the AVASTMES variable replaced by ":", to meet the syntax requirements of the DOS operating system.

The setting may be as follows:

```
SET AVASTMES=Brown
```

where Brown is the name of the user, to whom the report of the discovered virus is to be sent.

The main disadvantage of locating programs is the frequent need for updating the virus database. A six–month–old program may not be able to cope with the rapid expansion of new viruses. Therefore the LGUARD program determines the date on which the LGUARD.VPS was created and reports if more than six months have elapsed. It then continues operating in the normal manner.

The actions of LGUARD program may be partially influenced by setting the AVAST environment variable, which suppress the beeping sound in AGUARD, LGUARD and VGUARD. This variable may be set as follows:

```
SET AVAST=[S]
```

where S stands for suppressing the sound warnings.

Running the Program

```
LGUARD /H
LGUARD /V
LGUARD [[D:\]path][/E[ext1[;ext2...]]][/D][/C]
   [/M][/Ffile][/W[password|+]][/U[user]][/P]
   [/R[file]][/Z[M|A|F]][/X][/I][/S]
```

/H or /?

LGUARD displays **instructions** for use and ends.

/V

LGUARD displays current **list of viruses** it can detect and ends.

d:\path parameter

User may specify **disk and directory** of files to be tested. If this parameter is omitted, files in the root directory of the current drive are tested. More disks may be specified. If diskettes are tested, LGUARD asks whether another diskette is to be tested, and if yes, it prompts its insertion into the drive.

Parameter *:

When selected, LGUARD will test all local disks

Parameter #:

When selected, LGUARD will test all network disks. If „#:“ parameter is used and no network disk is found, the program displays an error message.

. parameter

This parameter enables to the user to specify that **currently valid directory on the currently valid disk will be tested**.

/E

User may specify **file extensions** of files to be tested. Up to 10 extensions may be specified, separated by a semicolon. Extensions may contain the characters "?" and "*" for requesting more than one file at a time. If this option is omitted, the COM, EXE, SYS, OV? and BIN extensions are checked.

/D

With this option, files in **subdirectories of a given directory will not be checked**. If it is omitted, such files are checked.

/C

This option specifies that the **entire content of the files is to be checked**. By default only the first and last 8192 bytes of each file are checked, which greatly increases speed. After detecting any virus, the program automatically switches to the mode of testing the whole files.

/M

With this option, only the **memory and the disk boot sector will be checked** for viruses. No file testing takes place. This enables a useful quick system check, for example in the AUTOEXEC.BAT file.

/B

The operating **memory will not be checked** for viruses. If, for example, you are sure that the computer contains no viruses and only wish to test diskettes, this option offers a useful short cut.

/A

With this option, the presence of **boot viruses in files** will be tested. In this way files containing an "image" of the system area of the disk may be tested without having to be copied back to the disk or diskette. After discovering any virus, the program automatically switches back to the mode of testing all types of viruses.

/Q

LGUARD skips the VPS file verification which makes execution (from a floppy) much faster.

/Ffile

Virus descriptions specified by the user will be used to test a given file. This enables users to add newly appeared virus types to the program.

/W[password|+]

When a virus is detected, a **user–defined message appears on the screen** giving directions for further action. The message is stored in the LGUARD.MSG file in the same directory as the LGUARD.COM program. If a password is specified, it must be given in order to continue the program after discovery of a virus and to interrupt the program. The parameter "+" serves to test the functioning of the message.

/U[user]

On discovery of a virus in the Novell network, a **message is sent to a specified network user**. If no such user is specified, the message is sent to the "Supervisor" user. The recipient must be authorized to the network and specified as the recipient. A message is also sent to the **system console of the network**.

/P

The program **runs continuously** without pause after the disk system area test, without waiting for the user's response when a virus is located. At the end, the results can be obtained from the exit codes.

/R[file]

A **report file will be created** listing the viruses found and containing a statistical table. If no filename is specified, the report file LGUARD.RPT will be saved in the current directory. If /R parameter is followed by a "*", the report file also records the files, in which no virus was found.

/Z

Parameter **/ZM** means delete just the virus macros, **/ZA** means delete all macros and **/ZF** means delete the whole file. In this case the program will ask you to give Yes or No reply. These parameters can be used together with /P (no pauses) too. The default action with /Z only deletes virus macros for exactly identified viruses and all macros for virus families. If no virus is found, program LGUARD displays the information when the file contains some macros, when it is the Word template, Word document or OLE2 file.

/X

This switch offers renaming of the suspicious files containing virus characteristics by the user. The first character of extension is after renaming the letter "V" (for instance COM becomes VOM, EXE becomes VXE, etc.).

/I

This switch is designed to **ignore the exchange media**. If a medium is tested in a drive that supports the exchange media, the program normally asks whether another medium will be tested. If an / I parameter is set, the program ends its activity after testing the medium.

/S

Detection of a virus will not be indicated by a **beep**. By default, detection of virus characteristics is accompanied by a beep.

Exit Codes

When the LGUARD program completes its operation, it returns an exit code to the operating system. This code may then be tested either by another ("parent") program or by using the batch file command IF ERRORLEVEL. The LGUARD exit code may only have the following values:

0	program ended normally, no virus found.
1	virus found in memory.
2	virus found on disk but not in memory.
3	program interrupted by user; no virus found so far.

4 error occurred during operation, no virus found so far.
98 program displayed list of viruses it can detect.
99 program displayed instructions for use.

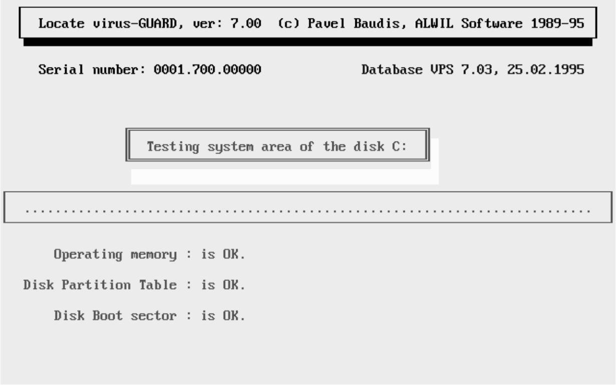
The exit codes thus defined are used especially in command batches for branching upon detection of a virus. An example is the command batch SCAN.BAT found on the distribution diskette which demonstrates the use of the LGUARD exit code and multiple disk test. Use of the exit code and the IF ERROR–LEVEL is apparent also from the following example, in which the meaning of the function is indicated by the jump label:

```
..
lguard c:
if errorlevel 4 goto error
if errorlevel 3 goto interrupt
if errorlevel 2 goto diskvir
if errorlevel 1 goto memvir
echo ** no virus found on disk c: **
..
```

Examples of Use

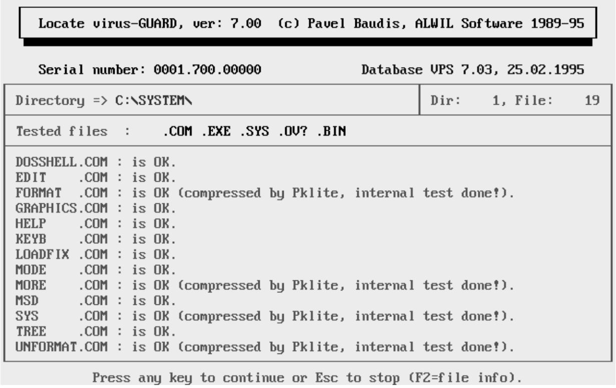
```
LGUARD C:\SYSTEM
```

The program scans the SYSTEM directory on the C: drive. All files of the COM, EXE, SYS, OV?, and BIN type in this directory are checked. Memory and system area on the hard disk are tested first. If nothing is found, the following information appears:



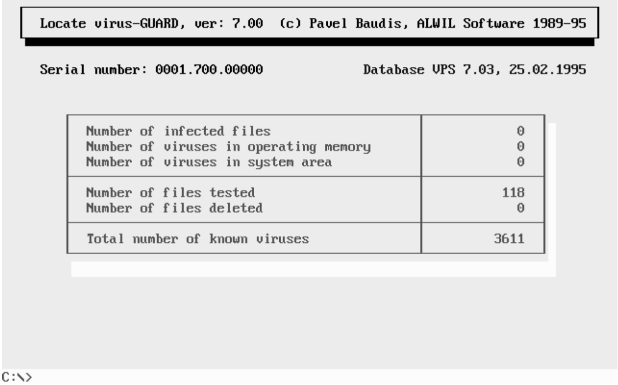
```
 Locate virus-GUARD, ver: 7.00  (c) Pavel Baudis, ALWIL Software 1989-95

 Serial number: 0001.700.00000           Database VPS 7.03, 25.02.1995


              Testing system area of the disk C:

 ..........................................................................

     Operating memory : is OK.

 Disk Partition Table : is OK.

     Disk Boot sector : is OK.
```

Then individual files are checked, and the user is informed as follows:



```
 Locate virus-GUARD, ver: 7.00  (c) Pavel Baudis, ALWIL Software 1989-95

 Serial number: 0001.700.00000           Database VPS 7.03, 25.02.1995
 Directory => C:\SYSTEM\                         Dir:    1, File:    19
 Tested files  :    .COM .EXE .SYS .OV? .BIN
 DOSSHELL.COM : is OK.
 EDIT    .COM : is OK.
 FORMAT  .COM : is OK (compressed by Pklite, internal test done!).
 GRAPHICS.COM : is OK.
 HELP    .COM : is OK.
 KEYB    .COM : is OK.
 LOADFIX .COM : is OK.
 MODE    .COM : is OK.
 MORE    .COM : is OK (compressed by Pklite, internal test done!).
 MSD     .COM : is OK.
 SYS     .COM : is OK (compressed by Pklite, internal test done!).
 TREE    .COM : is OK.
 UNFORMAT.COM : is OK (compressed by Pklite, internal test done!).
           Press any key to continue or Esc to stop (F2=file info).
```

At the end, the following table appears:



```
 Locate virus-GUARD, ver: 7.00  (c) Pavel Baudis, ALWIL Software 1989-95

 Serial number: 0001.700.00000            Database VPS 7.03, 25.02.1995

     Number of infected files                                 0
     Number of viruses in operating memory                    0
     Number of viruses in system area                         0

     Number of files tested                                 118
     Number of files deleted                                  0

     Total number of known viruses                         3611

C:\>
```

This table shows the number of viruses found in the operating memory, the system area of the disk, the total number of infected files and also the number of files deleted by the user. Also shown is the number of virus characteristics that LGUARD is currently able to detect.

Resident–GUARD

Along with traditional locating programs which must be run by the user, the AVAST! system also includes a memory resident program which can perform the same function automatically. This makes it possible to check any program for viruses before it is run.

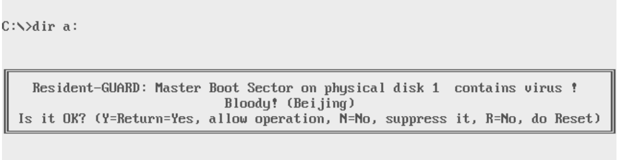
This type of program generally poses two serious problems: a large memory requirement, and a slow–down during testing. The RGUARD program solves the first problem by allowing the user to choose what is to be tested, and the part of the program for testing files takes about **1 KB of RAM**. In tackling the second problem we chose a compromise. File contents are tested only during program start–up and not during their reading, when there would be a long delay. Infected programs can therefore be copied, but they cannot be run. A slow–down occurs only during the start–up of an application, and there is no delay when working with files.

Further testing problems can emerge in computer networks. If the user is authorized only to execute programs but not to read (Execute Only attribute), then clearly the content of the file cannot be tested for viruses. Nevertheless, RGUARD allows the program to be executed.

RGUARD uses the same virus characteristic database for testing as LGUARD. The database is stored in the LGUARD.VPS file. This means that it is not able to detect any more viruses than LGUARD. On the other hand, it tests for the presence of viruses continually and automatically, which is a great advantage for many users.

If more than six months have elapsed since the LGUARD.VPS file was created, RGUARD reports the fact and recommends that you update the file. Our AVS antivirus service updates the file about twelve times per year.

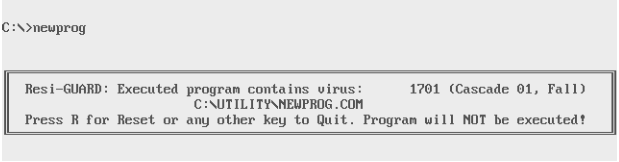
RGUARD offers two independent types of search. The first tests the system sector of the disks and diskettes used. If you use diskettes (even if they only contain files), the boot sector is scanned for boot viruses. The only way a boot virus can attack a computer is by spreading during an attempt to boot the system from an infected diskette. Users often forget to remove the diskette in the A: drive, and thereby viruses can infect the hard disk. This kind of test does not slow down work with the computer and takes up about 8 KB of operating memory. If RGUARD finds a virus, it reports the fact in a special window:



```
C:\>dir a:

  Resident-GUARD: Master Boot Sector on physical disk 1  contains virus !
                            Bloody! (Beijing)
 Is it OK? (Y=Return=Yes, allow operation, N=No, suppress it, R=No, do Reset)
```

The answer N suppresses reading. If the user wishes to continue working with the diskette in spite of the virus (for example to use data files stored on it), he may select Y. Pressing Enter has the same effect. The user has a further option: R, by which the system is rebooted, as when Ctrl+Alt+Del are pressed. The user is then prompted to insert the original system diskette in drive A: and switch the computer off and on again.

RGUARD also offers the possibility, as mentioned earlier, of testing the contents of executed files. The code for the test is very small, only about 1 KB of operating memory. The test uses the file RGUARD.OVL, which must be placed in the same directory as the RGUARD program. The virus search in the files is carried out before the start of the application and if a virus is found, again a window appears with a message:



```
C:\>newprog

 Resi-GUARD: Executed program contains virus:      1701 (Cascade 01, Fall)
                         C:\UTILITY\NEWPROG.COM
 Press R for Reset or any other key to Quit. Program will NOT be executed!
```

In this case the user has only two options. If the R key is pressed, the computer will be rebooted. If any other key is pressed, the program is cancelled.

The RGUARD program is also able to employ user–defined virus characteristics. By selecting the /F option during RGUARD installation, the name of user–defined characteristic file can be specified. Techniques for defining virus characteristics are described in the chapter about LGUARD.

RGUARD automatically recognizes a Novell network during its installation and is able to test programs run from network disks. To test disks of other networks, the network control must be established before the program is installed.

RGUARD also supports the Novell network in other ways. It offers the possibility of sending reports of virus detection at a work station to selected recipients. For this, the /U option is selected, followed by the name of the recipient. If no name is given, the report is sent to the “Supervisor” user. The user will receive the report only if he is authorized to the network and specified as a recipient. In addition the report is sent, together with the date and time, to the system console of the network, as follows:

```
10.10 16:20 DOE[07] virus Arusiek
```

Reports are sent during an attempt to start an infected program and on detection of a boot virus. Reports of a boot virus are sent only if both parts of the program are installed in the memory (see below). It is also sent with a certain delay. This is designed to assure the security of the computer network, whose functioning may be under far greater threat from viruses. The **/U** parameter may be replaced by defining the **AVASTMES** environment variable. This variable designates the Novell network user, to which the report of the discovered virus will be sent and serves to ensure more flexible reporting of the virus occurrence in the network. If defined during the RGUARD installation, the /U parameter does not have to be set.

The report of virus detected on a given work station may also be sent to a user, who finds himself in a different context (applies to the Novell network, version 4.00 and higher). Such a user must be specified exactly in the same pattern, as it is listed by the NLIST program with parameters USER /A /B. The only exception is that the character "=" must be for the AVASTMES variable replaced by ":", to meet the syntax requirements of the DOS operating system.

The setting may be as follows:

```
SET AVASTMES=Brown
```

where Brown is the name of the user, to whom the report of the discovered virus is to be sent.

RGUARD is also able to utilize the XMS type memory, and therefore may only occupy 6.2 KB of basic memory even in case both of its functions are installed. It may be even be installed to UMB memory outside the basic memory. During its installation to the UMB memory, a report "insufficient UMB memory" may appear under certain conditions. In such a case, its installation needs to be shifted before other resident programs or to the basic memory.

Note for Programmers

RGUARD can work with programmer–designed programs which can perform a useful function in case viruses are found. If RGUARD finds a virus, it sends a message to the system through an interrupt. For this message the following interface was created using multiplexer (interrupt 2Fh):

a) Virus found in disk system area

```
register AX     =       DAD1h
register BX     =       1
register DL     =       number of disk tested (0=diskette A, 80h=first hard disk)
```

```
registers ES:DI =       pointer on detected virus name, ending with zero
```

b) Virus found in executed program

```
register AX     =       DAD1h
register BX     =       2
registers DS:DX =       pointer on detected virus name, ending with zero
registers ES:DI =       pointer on detected virus name, ending with zero
```

Running the Program

```
RGUARD /H
RGUARD /V
RGUARD [/B[+|-]] [/E[+|-]] [/U[user]] [/Ffile]
```

/H or /?

RGUARD displays **instructions** on use and ends.

/V

RGUARD displays **current list of known boot viruses** that it is able to detect and ends.

/B[+|–]

When selected, this option enables the user to test the system areas of disks and diskettes for **boot viruses**. If selected during installation of the RGUARD program, space is reserved in the memory for this test. Testing is activated or deactivated by selecting "+" or "–", which can be done at any time, provided that space was reserved for the test at installation.

/E[+|–]

When selected, this option enables the user to test **executed programs for the presence of viruses**. If selected during installation of the RGUARD program, space is reserved in the memory for this test. Testing is activated or deactivated by selecting "+" or "–", which can be done at any time, provided that space was reserved for the test at installation.

/U (User)

When selected, this option insures that reports of virus detection in a **Novell network will be sent to specified users**. If no name is given, reports are sent to the Supervisor user. The recipient user must be authorized to the network and specified by name. In addition reports are sent to the network system console.

/3[+|–]

This switch enables to the user to specify permission or prohibition of a 32–bit access to the disk in the Windows environment.

/Ffile

When selected, this option insures that **user–defined characteristics** from the appropriate file will be used in the search. This enables the user to add new virus characteristics which may have emerged in his system to the program.

/X

This switch **prohibits** any parameter and operation mode changes upon the following start–ups or when the Hot Key is used.

/W

This switch determines that the Novell network installation will be ignored. The network drivers may then be removed from the memory, however, RGUARD is in this case incapable of monitoring the network activity (operation).

/O

This switch is designed for use of **floptical disks** or otherwise nonstandard diskette drives.

/D

This switch makes sure that **the diskettes infected by a boot virus are not accessible.** The data on this diskette cannot then be read and the user is notified of the virus' presence and asked to remove the diskette from the drive immediately.

Exit Codes

At the moment of RGUARD's termination, it returns an exit code to the operating system. This code may later be tested either by another (parent) program or in the command batch using IF ERRORLEVEL. The RGUARD exit code may have only the following values with the following meanings:

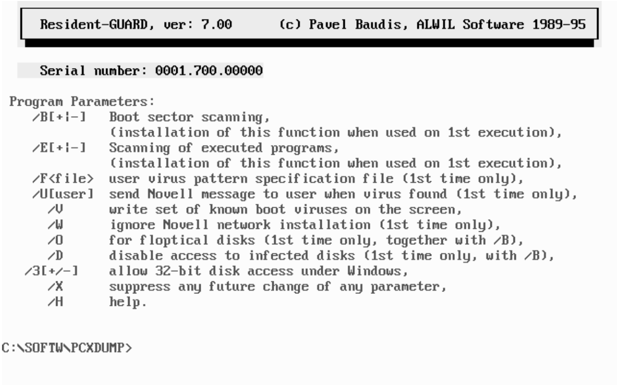
0 Program installed in memory
1 Program installed earlier in memory
2 Error occurred; program cannot be installed in memory
98 Program generated a list of viruses it can locate
99 Program generated instructions on use.

These exit codes can be used especially in command batches for branching during installation of the program in the memory.

Examples of Use

```
RGUARD /?
```

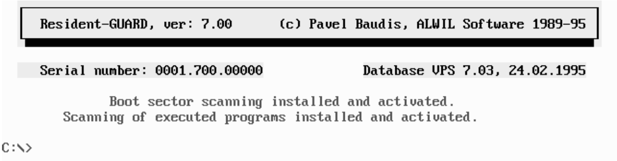
The program displays simple instructions for its use.



```
 Resident-GUARD, ver: 7.00      (c) Pavel Baudis, ALWIL Software 1989-95

 Serial number: 0001.700.00000

Program Parameters:
   /B[+¦-]   Boot sector scanning,
             (installation of this function when used on 1st execution),
   /E[+¦-]   Scanning of executed programs,
             (installation of this function when used on 1st execution),
  /F<file>   user virus pattern specification file (1st time only),
  /U[user]   send Novell message to user when virus found (1st time only),
     /V      write set of known boot viruses on the screen,
     /W      ignore Novell network installation (1st time only),
     /O      for floptical disks (1st time only, together with /B),
     /D      disable access to infected disks (1st time only, with /B),
  /3[+/-]    allow 32-bit disk access under Windows,
     /X      suppress any future change of any parameter,
     /H      help.

C:\SOFTW\PCXDUMP>
```

RGUARD

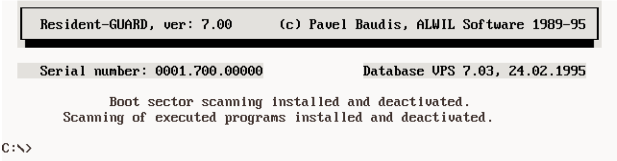
At the first start–up the RGUARD program is installed in the memory and all safeguards are active (see figure below); on further start–ups the current state of the program is displayed.



```
 Resident-GUARD, ver: 7.00      (c) Pavel Baudis, ALWIL Software 1989-95

 Serial number: 0001.700.00000            Database VPS 7.03, 24.02.1995

             Boot sector scanning installed and activated.
       Scanning of executed programs installed and activated.

C:\>
```

RGUARD /B- /E-

At the first start–up the RGUARD program is installed in the memory and space is reserved for testing both virus types. All safeguards are inactive (see figure). At further start–ups all active safeguards are deactivated.



```
 Resident-GUARD, ver: 7.00      (c) Pavel Baudis, ALWIL Software 1989-95

 Serial number: 0001.700.00000            Database VPS 7.03, 24.02.1995

             Boot sector scanning installed and deactivated.
       Scanning of executed programs installed and deactivated.

C:\>
```

General Antivirus Programs

General antivirus programs are very useful protective aids against computer viruses. These programs are not aimed against specific virus strains, but they have a general effectiveness. They are not always a hundred percent reliable in all circumstances, as are the specific programs dealt with above, but they are highly effective as **preventive tools** and as an early warning of the presence of viruses in systems already infected. They are able to discover the activity of so–far unknown viruses as well as alerting the user to the presence of known viruses.

These programs are based on a simple principle. Viruses can only spread by modifying a code executed by the computer. This code can only be located in the executive files and in system areas on hard disks and diskettes. In order to spread, viruses **must modify this executive code**. General antivirus programs monitor and test the integrity (i.e. the intactness) of data. Any changes in intactness are reported to the user, who can take appropriate action once the cause of the change has been established.

However, this is not the only useful means. Viruses possess certain qualities and common features that are connected with their spread. When they infect files, viruses make use of operating system services and BIOS services in order to infect the system areas. The particular functions often employed by viruses can be ascertained. These functions may therefore be highly dangerous. Based on the architecture of the processor and features of the operating system, a memory resident program may be designed to monitor the functioning of the operating system and BIOS in order to detect any suspicious circumstances.

A program able to save and then to restore critical system areas on hard disks represents an invaluable tool against viruses. Such programs are easily able to remove any boot virus.

The AVAST! package of antivirus programs includes four general antivirus programs. If used in time they can prevent infection of personal computers, easily and effectively remove viruses, and thus prevent great losses.

Sum–GUARD

The simplest general antivirus program in the AVAST! package is SGUARD (Sum–GUARD), which very quickly and easily checks whether any changes have occurred in a specified file, the system area of a disk, or the operating memory. It is especially useful for checking often–used programs and important files and may also be used to check system programs (IO.SYS, MSDOS.SYS, and COMMAND.COM of the Micro–soft MS–DOS, or the IBMIO.COM, IBMDOS.COM and COMMAND.COM of the IBM PC–DOS) each time the computer is turned on. Testing for any changes in the operating memory is also very important.

The program functions in two operating modes. In the calculate mode a checksum is generated for a specified file. In this case a filename nay contain "*" and "?" to indicate more files. Checksums are then generated sequentially for all files matching the wildchar. (Refer to the user's manual of the operating system for more information). If the keyword "**.SYSTEM.X**" ("X" designating the disk) is given as a filename, a checksum of the disk system area is generated (the keyword begins with a dot in order to differentiate it from a filename).

If the keyword "**.MEMORY**" is used as a filename, the operating memory is checked rather than any part of the disk. Vectors of important interrupts, key data of the operating system, total memory size and the beginning of free memory are all taken into account. This feature of the SGUARD program enables it to detect effectively even unknown virus strains, whose presence might not show up in a test. This function is especially important at the beginning of the AUTOEXEC.BAT file. A false alarm can only be caused by changing the compu–

ter configuration (for example adding a new device driver in the CONFIG.SYS file). Since each memory resident program is included in this test, the following line should be typed in the AUTOEXEC.BAT file:

```
SGUARD .MEMORY 0
```

At the first reboot after this change, the SGUARD program reports the “0” as an error in the checksum along with the real determined value. Then the true value must be substituted for the value 0 in the AUTOEXEC.BAT file.

In the verify mode the current checksum of a specified file is compared with the sum given by the user on the command line. The user may specify multiple pairs of [file,sum] to check multiple files. If the currently valid sum agrees with the sum specified by the user, the program ends with the exit code 0 passed on to the system. If any mismatch is discovered, this is displayed on the screen accompanied by a beep, and the program ends with exit code 1. If the keyword “**.SYSTEM.X**” has been given as the filename (in which “X” designates the disk), the given sum will be compared with the currently valid system area sum of the specified disk. If the keyword “**.MEMORY**” has been specified as the filename, the given sum will be compared with the current sum of the currently selected valid areas of the operating memory.

The program can be started with the /B option for batch processing. Then in the calculate mode the program output corresponds to the batch command syntax. This can be easily created by redirecting the standard output to a BAT extension file, as shown in the section below. In the verify mode mismatches are not reported on the screen but only by the exit code appropriate for further batch processing. The exit code may be tested in batches by the IF ERRORLEVEL command, and appropriate steps can be taken according to its value. The SGUARD program returns an exit code of zero when all checksums agree, and a code of one if one or more of the sums does not agree.

Running the Program

```
SGUARD [/H]
SGUARD [/K] [/L] [/N] [/D] mask
SGUARD [/K] [/L] [/N] [/D] file1 sum1 [file2 sum2
   [....]]
```

/H or /?

SGUARD displays **instructions** for use and ends.

/B

With this option, the user may specify that SGUARD run in **batch mode** (see above).

"mask" parameter

This is a **general filename specification** for the calculate mode. If the "**.SYSTEM.X**" keyword is specified as the filename ("X" designating the disk), the checksum of the system area of the specified disk will be generated. If the keyword "**.MEMORY**" is specified as the filename, the checksum of the selected areas of the operating memory is generated.

"file1 sum1" parameters

This is a **pair specification [file,sum]** for the verify mode. If the "**.SYSTEM.X**" keyword is specified as filename ("X" designating the disk), the system area of the specified disk will be verified. If the keyword "**.MEMORY**" has been specified, the operating memory is checked.

Exit Codes

As mentioned above, the SGUARD program may be run in the verify mode using the /D option. The program then reports mismatches to the operating system only with the exit codes. This code is then to be tested either by another (parent) program or in a command batch with the IF ERRORLEVEL command. Only the following values with the following respective meanings are possible:

0	No change detected; all sums agree
1	Change in one or more checksums detected
99	Program displayed instructions for use.

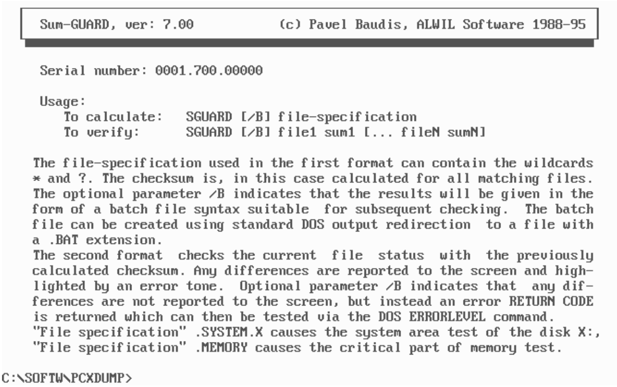
The exit codes thus defined may be used especially in command batches for branching upon virus detection. The use of the exit code and the IF ERRORLEVEL command is also clear in the following example, in which the names of the jump labels suggest the function:

```
..
sguard c:\io.sys 12345 c:\msdos.sys 67890
if errorlevel 1 goto change
rem ** tested file OK **
..

:change
echo Attention: system modified!! Must be checked!
```

Examples of Use

`SGUARD /?`

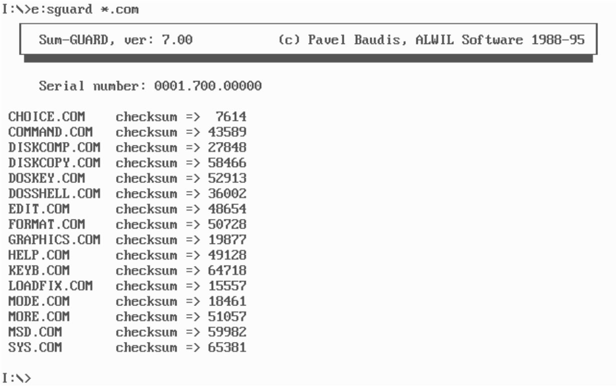
displays simple help instructions of SGUARD:



```
 Sum-GUARD, ver: 7.00          (c) Pavel Baudis, ALWIL Software 1988-95

 Serial number: 0001.700.00000

 Usage:
    To calculate:   SGUARD [/B] file-specification
    To verify:      SGUARD [/B] file1 sum1 [... fileN sumN]

The file-specification used in the first format can contain the wildcards
* and ?. The checksum is, in this case calculated for all matching files.
The optional parameter /B indicates that the results will be given in the
form of a batch file syntax suitable  for subsequent checking.  The batch
file can be created using standard DOS output redirection  to a file with
a .BAT extension.
The second format  checks the current  file  status  with  the previously
calculated checksum. Any differences are reported to the screen and high-
lighted by an error tone.  Optional parameter /B indicates that  any dif-
ferences are not reported to the screen, but instead an error RETURN CODE
is returned which can then be tested via the DOS ERRORLEVEL command.
"File specification" .SYSTEM.X causes the system area test of the disk X:,
"File specification" .MEMORY causes the critical part of memory test.

C:\SOFTW\PCXDUMP>
```

`SGUARD *.COM`

Program is run in calculate mode; checksums of all COM files in current directory are generated and displayed on the screen:



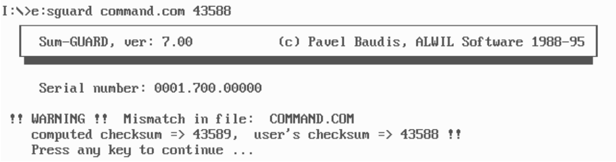
```
I:\>e:sguard *.com

  Sum-GUARD, ver: 7.00          (c) Pavel Baudis, ALWIL Software 1988-95

  Serial number: 0001.700.00000

CHOICE.COM    checksum =>  7614
COMMAND.COM   checksum => 43589
DISKCOMP.COM  checksum => 27848
DISKCOPY.COM  checksum => 58466
DOSKEY.COM    checksum => 52913
DOSSHELL.COM  checksum => 36002
EDIT.COM      checksum => 48654
FORMAT.COM    checksum => 50728
GRAPHICS.COM  checksum => 19877
HELP.COM      checksum => 49128
KEYB.COM      checksum => 64718
LOADFIX.COM   checksum => 15557
MODE.COM      checksum => 18461
MORE.COM      checksum => 51057
MSD.COM       checksum => 59982
SYS.COM       checksum => 65381

I:\>
```

```
SGUARD /D C:\*.* >CHECK.BAT
```

Program is run in calculate mode; the batch file CHECK.BAT is created to check all files in the root directory of drive C:.

```
SGUARD C:\IO.SYS 12345
```

Program is run in verify mode; current checksum of the IO.SYS file is generated and compared with the sum specified on the command line (12345); any mismatch is indicated by a beep and displayed on the screen:



```
I:\>e:sguard command.com 43588

  Sum-GUARD, ver: 7.00          (c) Pavel Baudis, ALWIL Software 1988-95

  Serial number: 0001.700.00000

!! WARNING !!  Mismatch in file:  COMMAND.COM
   computed checksum => 43589,  user's checksum => 43588 !!
   Press any key to continue ...
```

```
SGUARD /D .MEMORY 55932 .SYSTEM.C 26775 C:\IO.SYS
   46514 C:\MSDOS.SYS 51716 C:\COMMAND.COM 58487
```

Program is run in verify mode; currently valid checksums of the system area and the specified file are generated and compared with the sums specified on the command line. Any mismatch is indicated only by the exit code due to the /B option selected.

Alter–GUARD

The second general program in the AVAST! package is program AGUARD (Alter–GUARD). Its functions are similar to those of SGUARD, but it is far more complex. When started for the first time it creates its own database called AGUARD.DAT (stored in the \AVAST! directory on network disks), where all information about the disk system areas and files to be monitored is stored. Besides filenames, this database includes information about file attributes, size, the date and time of the last modification and, with the help of checksums, the contents. Each time the program is run thereafter, it compares all these items in the database with the current state and thereby determines any alterations in the system areas and files. Upon ending, the program displays a list of all files altered since the last database update, plus new files created and files deleted. The user can select files to be updated in the database. Some of the programme changes may of course be legal, for example new versions of programs, deletion of no longer needed files, etc. Normal changes might also include modifications which some programs perform when their configurations are stored (for example LIST, LapLink III, WordPerfect 4.2, Fastlynx, Setver, Sourcer). Thanks to this program, the user can easily determine what files have been modified, whether by viruses or other kinds of data damage.

But AGUARD is capable of much more. It can detect known viruses in modified or new files and can even eliminate viruses by general means (i.e. without knowledge of virus characteristics) and restore original files. With these abilities AGUARD becomes the most powerful and most important part of the AVAST! antivirus package.

AGUARD is a general program which if used correctly can indicate the presences of viruses and eliminate them in time. Moreover it is effective not only against viruses but also against any other form of damage to data.

The AGUARD program is able to recognize a disk compressed by the Stacker program. In this case the boot sector is not tested since it is not really a disk sector but a fictitious sector of the Stacker program which changes over time. The presence of Stacker is displayed for three seconds on the screen, after which the program continues to run. This time may be shortened by pressing any key.

The actions of AGUARD program may be partially influenced by setting the AVAST environment variable, which suppress the beeping sound in AGUARD, LGUARD and VGUARD. This variable may be set as follows:

```
SET AVAST=[S]
```

where S stands for suppressing the sound warnings.

The AGUARD program is also able to process data on extremely large disks. For such procedure, it however requires sufficient XMS memory, otherwise an error message will occur.

Running the Program

```
AGUARD /H
AGUARD [[d:\]path][.][/E[ext1[;ext2;...]]][/D]
   [/S][/A][/R][/F][/P][/O][/T]
```

/H or ?

When selected, AGUARD displays **instructions** for use and ends.

Parameter d:\path

The user may specify **disk and directory** of files to be tested. If deselected, files in the root directory of the current disk are tested.

Parameter *:

When selected, AGUARD will test all local disks

Parameter #:

When selected, AGUARD will test all network disks. If „#:“ parameter is used and no network disk is found, the program displays an error message.

Parameter .

The user may specify that the **current directory** of the current disk be tested.

/E

The user may specify **file extensions** of files to be tested. Up to 10 extensions may be specified, separated by a semicolon. Extensions may contain the wildchars “?” and “*” to specify more files at once. If deselected, files with extensions COM, EXE, BAT, SYS, BIN, OV? and VPS will be tested. To test all files, the parameter /E* should be used.

/D

When selected, files in **subdirectories of the current directory will not be checked**. If deselected these files will be checked.

/S

When selected, **no beep** will occur if mismatch is found. By default all mismatches and requests for user input are accompanied by a beep.

/A

When selected, change in **archive attribute is not tested**. The archive attribute is used by backup programs to detect program changes (it is cleared upon backup and checked for zero at the next backup execution; in this case backup of the file is not performed). This bit has no other meaning; however, all attribute changes in monitored files are checked by default.

/R

When selected, the program will run in **report mode**. Specified files will be checked, but the database cannot be updated.

/G

Parameter for the automatic restoring of infected files.

/F

When selected, **contents of files will not be checked with checksum**. Since all files must be read for checksum calculation, large quantities of data can take up a lot of time. This option permits a check of all directory entries (attributes, date and time, size). The database on the disk cannot be updated.

/C

Parameter for fast integrity checking. In this fast mode it tests the content of the file only when any of other items (attributes, file size, time stamp) has changed. Compared to the /F parameter this mode allows the user to update the database.

/P

When selected, the program will run continuously, smoothly without interferences by the user

/O

When selected, two **text files with the database status report** will be created in the current directory. The first is named AGUARD–X.ALL and contains information about all files stored in the database. The second is named AGUARD–X.DIF and contains information about all differences found between the states as stored in the database and current states detected on the disk. If no differences are detected, the second file will not be created. The letter X in the filename stands for the drive being tested.

/T

When selected, all new and modified files will be tested for presence of known types of viruses (corresponds to pressing Ctrl and T at the result menu of AGUARD program).

/X

This switch allows not testing of system areas of hard drive, which can be useful for data integrity testing of files, transferred electronically e.g. between notebook and desktop PC.

/B[n]

AGUARD is now able to work with extremely large disks. The XMS memory is used to store data, but this approach might cause some troubles with specific memory managers. Theis new parameter / B[n], specifies how many subdirectories should be included (the default number is 6000).

Exit Codes

When AGUARD ends, an exit code returns to the operating system. This code may later be tested either by another (parent) program or by using the IF ERRORLEVEL command. AGUARD exit codes may only have the following values with the following meanings:

0 program ended normally, no data modified
1 data modification detected; database not updated
2 data modification detected; database updated
3 program interrupted by user
4 error occurred during running
5 restore is ok
6 restore failed
7 virus found.
99 program displayed instructions for use

Exit codes thus defined may be used especially in command batches for branching upon virus detection. A good example demonstrating the use of the AGUARD exit code and multiple disk test is the CHECK.BAT file contained on the distribution diskette. Use of the exit code and the IF ERRORLEVEL command is also clear in the following example, in which the names of jump labels indicate the function:

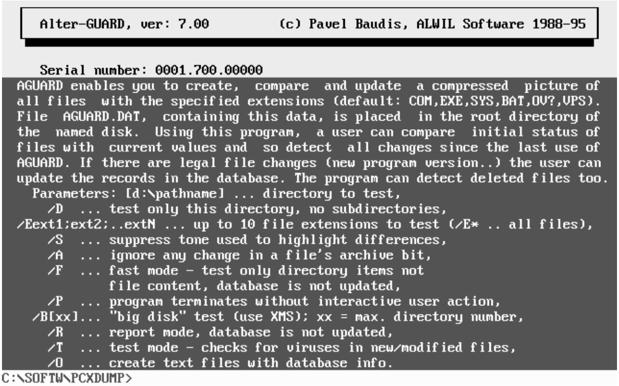
. .

```
aguard c:
if errorlevel 4 goto error
if errorlevel 3 goto interrupt
if errorlevel 2 goto change
if errorlevel 1 goto no_change
echo ** no data change detected on disk c:**
..
```

Examples of Use

AGUARD /?

Program lists a one–page help:



```
 Alter-GUARD, ver: 7.00        (c) Pavel Baudis, ALWIL Software 1988-95

 Serial number: 0001.700.00000
AGUARD enables you to create,  compare  and update  a compressed  picture of
all files  with the specified extensions (default: COM,EXE,SYS,BAT,OV?,VPS).
File  AGUARD.DAT,  containing this data, is placed  in the root directory of
the  named disk.  Using this program,  a user can compare  initial status of
files with  current values and  so detect  all changes since the last use of
AGUARD. If there are legal file changes (new program version..) the user can
update the records in the database. The program can detect deleted files too.
  Parameters: [d:\pathname] ... directory to test,
    /D  ... test only this directory, no subdirectories,
/Eext1;ext2;..extN ... up to 10 file extensions to test (/E* .. all files),
    /S  ... suppress tone used to highlight differences,
    /A  ... ignore any change in a file's archive bit,
    /F  ... fast mode - test only directory items not
            file content, database is not updated,
    /P  ... program terminates without interactive user action,
  /B[xx]... "big disk" test (use XMS); xx = max. directory number,
    /R  ... report mode, database is not updated,
    /T  ... test mode - checks for viruses in new/modified files,
    /O  ... create text files with database info.
C:\SOFTW\PCXDUMP>
```

AGUARD

Program is run for the current disk. All COM, EXE, BAT, SYS, OV?, BIN and VPS files on this disk are checked.

AGUARD C:\SYSTEM /D /S

Program scans the system directory on disk C:. Files in subdirectories are not checked. Mismatches are not accompanied by a beep.

AGUARD C:\ /E* /F /S

Program is run for a quick check of all files on disk C:. File content is not checked, so that the test is very quick.

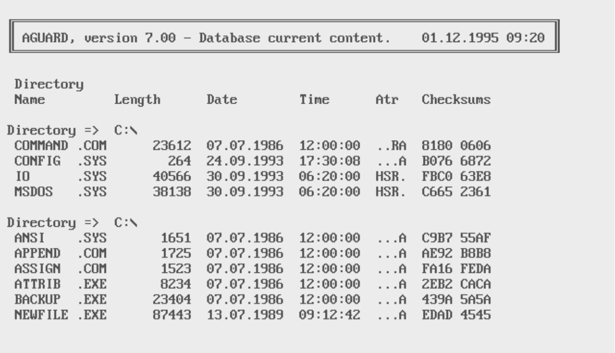
AGUARD D:\SUBDIR /Edwg;com;dat;doc /A /F

Program checks files with extensions DWG,COM, DAT and DOC in the SUBDIR directory on disk D:. No file contents but only directory entries are checked. Archive attribute changes are ignored.

```
AGUARD C:\ /O
```

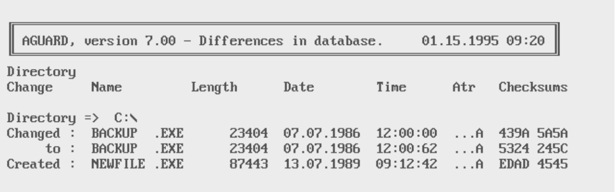
Program checks files with extensions COM, EXE, BAT, SYS and OV?. On the current disk two text files are created: the AGUARD–C.ALL file with the complete content of the database, and the AGUARD–C.DIF file containing differences detected during testing. These files appear as follows:

AGUARD–C.ALL file:



```
AGUARD, version 7.00 - Database current content.    01.12.1995 09:20

 Directory
 Name         Length      Date        Time       Atr    Checksums

Directory =>  C:\
 COMMAND .COM      23612  07.07.1986  12:00:00   ..RA   8180 0606
 CONFIG  .SYS        264  24.09.1993  17:30:08   ...A   B076 6872
 IO      .SYS      40566  30.09.1993  06:20:00   HSR.   FBC0 63E8
 MSDOS   .SYS      38138  30.09.1993  06:20:00   HSR.   C665 2361

Directory =>  C:\
 ANSI    .SYS       1651  07.07.1986  12:00:00   ...A   C9B7 55AF
 APPEND  .COM       1725  07.07.1986  12:00:00   ...A   AE92 B8B8
 ASSIGN  .COM       1523  07.07.1986  12:00:00   ...A   FA16 FEDA
 ATTRIB  .EXE       8234  07.07.1986  12:00:00   ...A   2EB2 CACA
 BACKUP  .EXE      23404  07.07.1986  12:00:00   ...A   439A 5A5A
 NEWFILE .EXE      87443  13.07.1989  09:12:42   ...A   EDAD 4545
```

AGUARD–C.DIF file:



```
AGUARD, version 7.00 - Differences in database.    01.15.1995 09:20

Directory
Change     Name          Length      Date        Time       Atr    Checksums

Directory =>  C:\
Changed :  BACKUP  .EXE      23404  07.07.1986  12:00:00   ...A   439A 5A5A
     to :  BACKUP  .EXE      23404  07.07.1986  12:00:62   ...A   5324 245C
Created :  NEWFILE .EXE      87443  13.07.1989  09:12:42   ...A   EDAD 4545
```

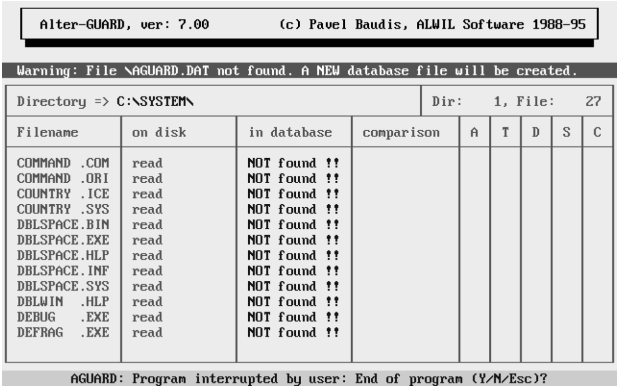
Communication between Program and User

The AGUARD program is the most complex of the AVAST! package and offers the user a variety of options. So here we show an example illustrating the program's use and its communication with the user.

The program is started for the first time with the command:

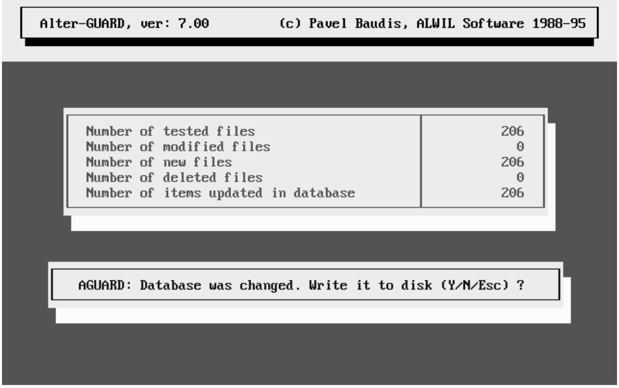
```
AGUARD C:
```

If the program does not find the AGUARD.DAT file in the root directory of the current disk, a new database will be created automatically. In this case the user's input is not needed for adding programs to the database. All files matching conditions found on the command line (directory, subdirectories, extensions) are included automatically. The AGUARD.DAT file is attributed "READ ONLY": it can be neither deleted nor modified by mistake. The following information appears on the screen:



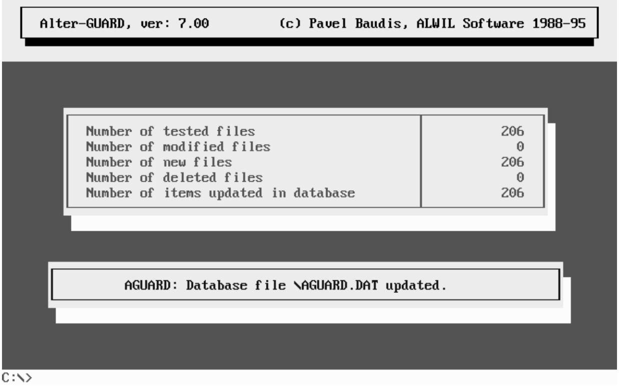
```
 Alter-GUARD, ver: 7.00          (c) Pavel Baudis, ALWIL Software 1988-95

Warning: File \AGUARD.DAT not found. A NEW database file will be created.

Directory => C:\SYSTEM\                                  Dir:    1, File:    27
Filename      on disk       in database    comparison     A   T   D   S   C

COMMAND  .COM  read          NOT found !!
COMMAND  .ORI  read          NOT found !!
COUNTRY  .ICE  read          NOT found !!
COUNTRY  .SYS  read          NOT found !!
DBLSPACE.BIN   read          NOT found !!
DBLSPACE.EXE   read          NOT found !!
DBLSPACE.HLP   read          NOT found !!
DBLSPACE.INF   read          NOT found !!
DBLSPACE.SYS   read          NOT found !!
DBLWIN   .HLP  read          NOT found !!
DEBUG    .EXE  read          NOT found !!
DEFRAG   .EXE  read          NOT found !!

         AGUARD: Program interrupted by user: End of program (Y/N/Esc)?
```

Here it can be seen that the database will be created. All files are reported as not found. Therefore no comparison was carried out. The program asks whether the database should be stored on the disk:



```
Alter-GUARD, ver: 7.00          (c) Pavel Baudis, ALWIL Software 1988-95

   Number of tested files                                206
   Number of modified files                                0
   Number of new files                                   206
   Number of deleted files                                 0
   Number of items updated in database                   206

   AGUARD: Database was changed. Write it to disk (Y/N/Esc) ?
```

The user saves the database by answering “Y”, and updating is reported on the screen:



```
Alter-GUARD, ver: 7.00          (c) Pavel Baudis, ALWIL Software 1988-95

   Number of tested files                                206
   Number of modified files                                0
   Number of new files                                   206
   Number of deleted files                                 0
   Number of items updated in database                   206

          AGUARD: Database file \AGUARD.DAT updated.

C:\>
```

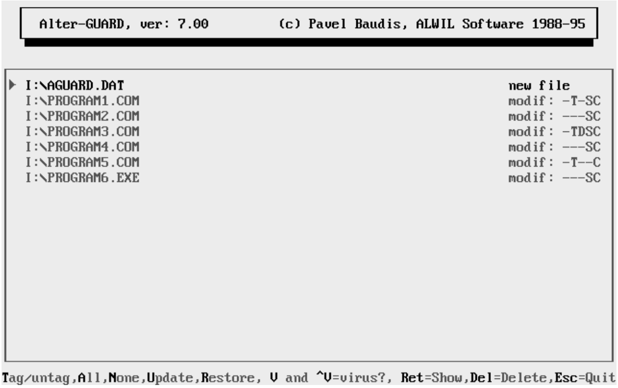
At this point we shall modify some files in the C:\PROGRAMS directory. Afterwards we test AGUARD’s response by starting the program with the following command:

```
AGUARD C:
```

If the program finds the AGUARD.DAT file in the root directory of the current disk, it reads the content and checks the current state of the tested parts of the files against the information in the database. All mismatches are reported on the screen as follows.

The table shows that files were found in the database and that they could be compared. Agreement is indicated by a dot and disagreement by a rectangle in the respective column. If all items are consistent, “OK” appears in the comparison column. Other files are marked “not OK!!” so that the user may process them.

If no changes have been detected, the program ends. If changes were found, a list of all modified files, all new files, and all files not found appears on the screen. The user can move through the list with a triangular cursor using the arrows and the PgUp, PgDn, Home and End keys.



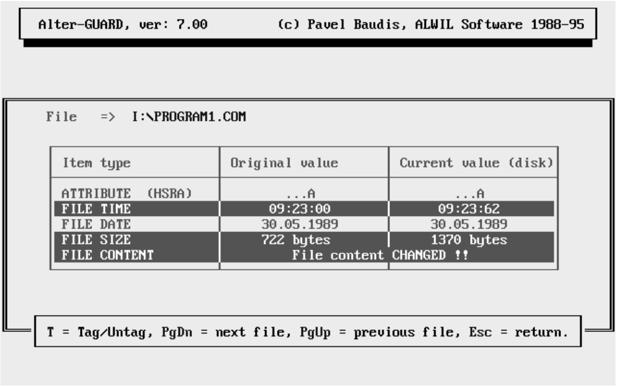
```
Alter-GUARD, ver: 7.00        (c) Pavel Baudis, ALWIL Software 1988-95

▶ I:\AGUARD.DAT                                          new file
  I:\PROGRAM1.COM                                        modif: -T-SC
  I:\PROGRAM2.COM                                        modif: ---SC
  I:\PROGRAM3.COM                                        modif: -TDSC
  I:\PROGRAM4.COM                                        modif: ---SC
  I:\PROGRAM5.COM                                        modif: -T--C
  I:\PROGRAM6.EXE                                        modif: ---SC

Tag/untag,All,None,Update,Restore, V and ^V=virus?, Ret=Show,Del=Delete,Esc=Quit
```

Using the “T” key (Tag/Untag), the user can tag files to be updated in the database (files changed purposely, new files or files deleted). A file is untagged by pressing the key again. The “A” key (Tag All) tags all files for updating in the database, while the “N” key (None) untags all files. If the AGUARD program was not started in Report mode, new and modified files can be deleted. After pressing the Del key the user is prompted to confirm deletion of the highlighted file. Once approved by the user, the file is permanently deleted from the disk.

There are new options for selecting files since version 7.5:it is Ctrl–M for selection of all modified files, Ctrl–N for selection of all new files, Ctrl–D for selection of all deleted files, Ctrl–A for selection of all files with changed archive attribute and Ctrl–F for selection of all files in current directory. It is possible to use Ctrl–E for finding next folder containing some changes. A short help would appear after pressing F1 key in the table with changes.

The AGUARD program can be ended by pressing Esc. After pressing the “U” key (Update and Quit), all tagged files are updated in the database: entries of modified files are updated, entries of new files are added to the database, and entries of deleted files are deleted from the database.

With the Enter key, all monitored items for a chosen file are displayed from the database as well as their actual status. Any changes are displayed in reverse video. If the triangular cursor is placed on the PROGAM1.COM file, for example, and Enter is pressed, the following information is displayed:



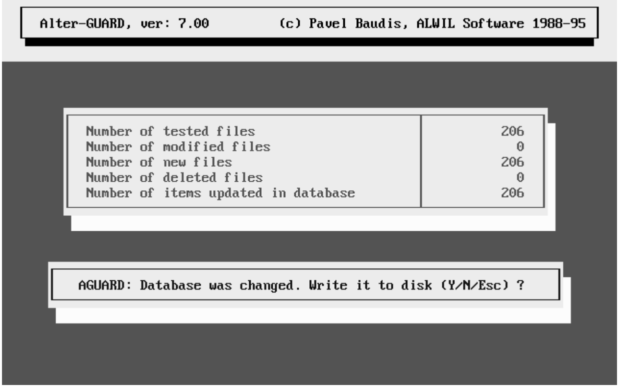
```
Alter-GUARD, ver: 7.00        (c) Pavel Baudis, ALWIL Software 1988-95

   File    =>  I:\PROGRAM1.COM

   Item type          Original value        Current value (disk)

   ATTRIBUTE  (HSRA)       ...A                  ...A
   FILE TIME             09:23:00              09:23:62
   FILE DATE            30.05.1989            30.05.1989
   FILE SIZE             722 bytes             1370 bytes
   FILE CONTENT               File content CHANGED !!

 T = Tag/Untag, PgDn = next file, PgUp = previous file, Esc = return.
```

The kind of change can be seen for each modified file. The difference between the length of an original file and its modified version is of special importance for detecting virus attack. This difference is specific to many viruses. In our case the difference is 648 bytes, which, together with the time of the last modification (62 seconds), points to the presence of virus 648. We can check this in a moment.

Even in this “detailed” menu files can be selected for updating in the database with the “T” (Tag/Untag) key. The file will be untagged when the key is pressed again.

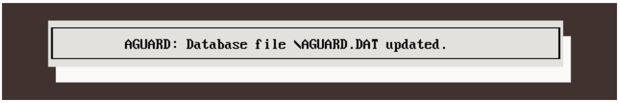
The user can move through the files with the PgUp and PgDn keys, and the Esc key will end the display.

If all the changes detected in the files are deemed correct, the "A" key (Tag All) and "U" (Update and Quit) may be pressed. The database is modified, and the program asks if the changes should be stored on the disk:



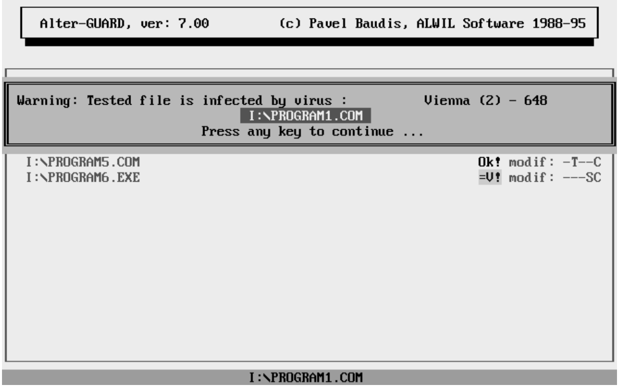
```
Alter-GUARD, ver: 7.00        (c) Pavel Baudis, ALWIL Software 1988-95

Number of tested files                          206
Number of modified files                          0
Number of new files                             206
Number of deleted files                           0
Number of items updated in database             206

AGUARD: Database was changed. Write it to disk (Y/N/Esc) ?
```

When the user confirms entry in the database with the "Y" key, the completed update is reported on the screen:



```
AGUARD: Database file \AGUARD.DAT updated.
```

What should be done if there is no rational explanation for a change in the database content? In this case, it is likely that the files have been infected by a virus. The user can check this easily, because AGUARD offers the option of **testing for known viruses**. This is highly useful whenever files begin to change for no apparent reason. When AGUARD displays the table with all changes detected, the user may press "T" (Test for Virus) to check whether the file contains a virus known to the LGUARD and RGUARD programs. More than one file can be tested by pressing "Ctrl" and "T" together, in which case all files from the designated one to the end of the list are tested. The test may be interrupted at any time with Esc. During test–ing AGUARD works with the RGUARD program. Therefore the RGUARD.OVL and LGUARD.VPS files must be located in the same directory as the AGUARD.COM file. The following display shows the Vienna 648 virus detected in the selected file:



```
Alter-GUARD, ver: 7.00        (c) Pavel Baudis, ALWIL Software 1988-95

Warning: Tested file is infected by virus :        Vienna (2) - 648
                          I:\PROGRAM1.COM
                     Press any key to continue ...

I:\PROGRAM5.COM                                     Ok! modif: -T--C
I:\PROGRAM6.EXE                                     =V! modif: ---SC

                          I:\PROGRAM1.COM
```

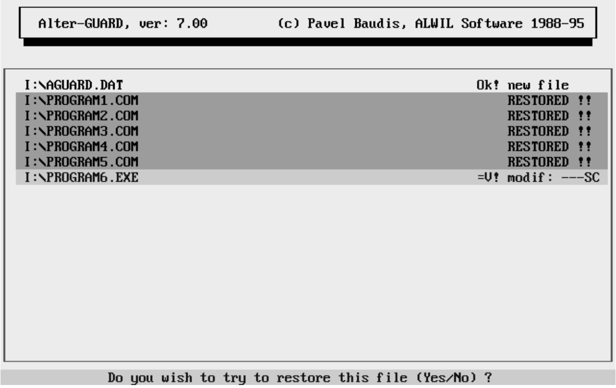
The presence of a virus is thus highlighted on the screen. If a virus was discovered in one of the files, "=V!" appears in the appropriate column. If no virus is detected, "Ok!" appears. The user may delete the infected file or attempt to restore it.

A **general restoration of infected files and removal of viruses** is one of the main advantages of the AGUARD program. For this it employs its capabilities as a program for checking the integrity of data. AGUARD has at its disposal a large quantity of information about original files. (New files which are infected with viruses cannot be restored by this method). When "R" is pressed, AGUARD tries several different methods for removing viruses and restoring the original file. The chief advantage here is that AGUARD is capable of deciding whether they have been successful, i.e. whether the file is in exactly the same condition as before infection. The "R" command cannot be used if the /F or /R options were selected or if the file is write–protected.

Important Note:

If you wish to restore files, you should turn off the computer, then turn it on again and start the system from the original system disk, then start the AGUARD program from the rescue diskette created during the installation of the AVAST!

system. Then you can be absolutely sure that no virus is active and the files can be easily restored by the above method.

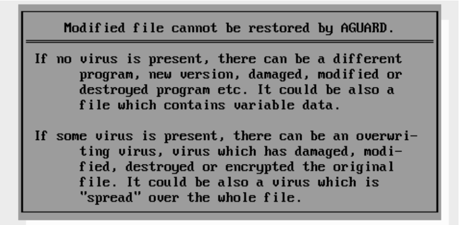


```
Alter-GUARD, ver: 7.00        (c) Pavel Baudis, ALWIL Software 1988-95

I:\AGUARD.DAT                                                Ok! new file
I:\PROGRAM1.COM                                                RESTORED !!
I:\PROGRAM2.COM                                                RESTORED !!
I:\PROGRAM3.COM                                                RESTORED !!
I:\PROGRAM4.COM                                                RESTORED !!
I:\PROGRAM5.COM                                                RESTORED !!
I:\PROGRAM6.EXE                                              =V! modif: ---SC

              Do you wish to try to restore this file (Yes/No) ?
```

Almost all viruses can be eliminated by this method. But there are a few exceptions. For example, overwriting viruses destroy the content of the original file so that it cannot be reconstructed and must be retrieved from a saved copy. Also files attacked by viruses which code the original file or are “scattered” throughout the entire host program cannot be reconstructed by AGUARD. Such viruses, however, are very rare at present, but unfortunately, One Half virus is among them.

On the other hand, **AGUARD is now able to remove new and so far unknown virus strains** without requiring constant updating. It is effective against polymorphous virus which are aimed especially against locating programs.

If AGUARD is unable to renew an original file, the following message appears:



```
     Modified file cannot be restored by AGUARD.

If no virus is present, there can be a different
      program, new version, damaged, modified or
      destroyed program etc. It could be also a
      file which contains variable data.

If some virus is present, there can be an overwri-
      ting virus, virus which has damaged, modi-
      fied, destroyed or encrypted the original
      file. It could be also a virus which is
      "spread" over the whole file.
```

The AGUARD program cannot directly uncover the ways in which a virus operates. However, it does enable the user to detect in time the changes caused by viruses. Changes which have been updated in the database will not be detected in the next running of the program. Therefore we recommend that AGUARD be run regularly (daily, weekly, etc.) and that the AGUARD.DAT file be saved separately on a diskette. Then the database must always be copied to the root directory of the appropriate disk before the program is run. This ensures that nobody has updated the database without authorization.

More detailed information on steps to be taken if the presence of a virus is suspected may be found in the chapter "Elimination of Viruses from the System", below.

File–GUARD

Another general antivirus program in the AVAST! package is FGUARD (File–GUARD). This program differs from the first two in a fundamental way. It is a memory resident program (occupying about 10.5 KB of operating memory) which monitors all attempts to modify specified files in real time (deleting or opening files for write, writing, renaming, cancelling the "Read Only" attribute).

Aside from its basic operation—monitoring the modification of files—FGUARD is able to trace destructive effects of viruses. It can monitor attempts to format a track, write directly on a disk through DOS or BIOS, and, on IBM PC/AT computers, the content of the CMOS configuration memory. It can also monitor attempts to provide a single step through interrupts 13h, 21h and 40h, by which some viruses detect an internal address which they then call up directly, thereby bypassing the antivirus protection. Single step is made possible by the Intel processor design, so that after each operation performed, control is passed to the interrupt 1 routine. This feature is currently used by debuggers; however it is also open to abuse. At this moment such a program exercises nearly absolute control over the processor operation. FGUARD can monitor changes in interrupt vectors 13h, 21h, 26h and 40h as well. Most changes in these vectors are made by memory resident programs, but they may also be made by viruses infecting the operating memory.

When any attempt at this type of operation is detected, a window appears on the screen with a message about action to be taken and further specification (filename, format track number, sector number, cluster, etc.). The user is prompted to confirm whether he wishes to take the action (delete file, format diskettes, etc.). If the user so specifies, the action will not be undertaken.

If the user only wishes to monitor computer operation and does not wish to answer the above questions, he may use another operation mode: REPORT mode, a report on the occurrence of an operation such as file deletion, track format, etc., is displayed in the upper right corner of the screen for at least 10 seconds.

FGUARD's operation can be temporarily suppressed in two ways: from the command line by selecting an option, or by pressing Ctrl+5 on the numerical keyboard. The activation of the FGUARD program is signalled by three ascending tones and its deactivation by three descending tones.

By selecting options the user can determine what actions FGUARD will monitor (see below). Always monitored are: a) read and write into the disk partition table sector, and b) write into the hard disk boot sector. In addition the user can select up to 10 file extensions to be monitored by the FGUARD program.

FGUARD is able to detect most virus strains on their first attempt to infect the operating system or other "healthy" programs. In this way the identification of hosts (programs responsible for virus spread) has become much easier. If a virus does not use the operating system services for its spread (i.e. all file operations are performed at its own expense, which results in a significant increase of the code size), or if it bypasses the operating system in some other way, it cannot be detected by FGUARD. FGUARD is also able partially to prevent destructive effects of viruses (disk format, write with DOS or BIOS, CMOS memory modification).

FGUARD is a memory resident program which should be executed in the AUTOEXEC.BAT initial batch file.

The user may specify programs not to be monitored by FGUARD. This is a useful feature, for example, if some application tends to cause false alarms. A list of such files may be saved in a special file, FGUARD.EXC, which must be located in the same directory as the FGUARD program itself. During installation (the first saving on the memory) it is necessary to specify with option /E that the program work with this special file of exceptions. In the FGUARD.EXC file each filename (without directory) is listed on a separate line. Names may not contain wildchars "?" and "*". Thus, if we wish not to monitor the files PROGRAM1 and PROGRAM2, we type:

```
PROGRAM1.COM
PROGRAM2.EXE
```

It should be mentioned that some programs open files for read and write incorrectly, even if the action involves reading only. This is especially true of programs written in Turbo Pascal. This of course can lead to false alarms in the FGUARD program.

An operation mode can be selected for any type of monitoring. Each option may be supplemented by one of three supplementary parameters: "+", "–", or "R". The character "+" denotes active monitoring in the interacting mode, the character "–" denotes inactive (switched off) monitoring, and the letter "R" denotes monitoring in the report mode. In addition the special switch /R+ can be used, setting all active monitoring in the report mode. /R– sets all active monitoring to the interacting mode.

FGUARD automatically recognizes a Novell network driver and is thus able to test programs run from network disks. To test disks on other computer networks it is necessary to install the program after the network drivers have been introduced.

Communication between a memory resident program and the user is very complex, especially if the current program uses one of the graphic modes or if it runs in a mode different from the real (the basic 8086 processor) mode. If FGUARD is run in Report mode, virus detection in graphic modes is signalled by a beep. In other cases the message is switched to text mode, then after communication, back to the original graphic mode. The program restores all colour and cursor settings, etc., but not the original content of the graphic screen. This must be restored in the appropriate application program. This phenomenon does not occur in the Windows environment if FGUARD for Windows, described in the following chapter, is installed.

Running the Program

```
FGUARD /H
FGUARD [ext1 [ext2..]][/I+][/F+][/D+][/B+][/C+]
   [/R+][/V+][/S+][/A+][/E][/X][/W][/3+]
FGUARD [ext1 [ext2..]][/I-][/F-][/D-][/B-][/C-]
   [/R-][/V-][/S-][/A-][/E][/X][/W][/3-]
FGUARD [ext1 [ext2..]][/IR][/FR][/DR][/BR][/CR]
   [/VR][/SR][/E][/X][/W]
```

/H or /?

FGUARD displays **instructions** for use.

Parameter ext1 ext2...

These parameters enable the user to request up to ten different **file extensions**, thus determining which types of files are to be monitored. The extensions are separated by a space. Extensions may contain the wildchars “?” and “*” for more files at once. By default COM, EXE, SYS, OV?, BIN and VPS files are monitored.

/I

The user may specify whether disk **track format monitoring** (interrupt 13h, 40h) will be active (/I+), only reported (/IR) or inactive (/I–). Hard disk formatting cannot be switched off permanently; it can only be suppressed until the program ends. I+ is the standard setting.

/F

The user may specify whether **monitoring of modification of specified file types** (interrupt

21h) will be active (/F+), only reported (/FR), or inactive (/F–). F+ is the standard setting.

/D

The user may specify whether **monitoring of direct write by DOS** (interrupt 26h) will be active (/D+), only reported (/DR), or inactive (/D–). Write onto hard disk boot sectors cannot be switched off permanently; it can only be suppressed until the program ends. D+ is the standard setting.

/B

The user may specify whether **monitoring of direct write by BIOS** (interrupt 13h and 40h) will be active (/B+), only reported (/BR), or inactive (/B–). Write onto hard disk boot sectors and disk partition table sectors cannot be switched off permanently; it can only be suppressed until the program ends. As far as physical disks are concerned, diskettes are denoted by letters (A,B) and hard disks by numbers (1,2). B– is the standard setting. At program start–up in Windows this parameter must be set to inactive mode!

/C

The user may specify whether **monitoring of CMOS memory content** for PC/AT computers will be active (/C+), only reported (/CR) or inactive (/C–). If monitoring is active, content of the CMOS memory is checked periodically (approximately once per second) and all changes are reported. If the user designates the modification as suspicious, FGUARD restores the original content of the CMOS memory. C– is the standard setting.

/R

The user may specify whether FGUARD will require a user input on detection of a suspicious action (R–) or whether only a **report** on such a detected ac–

tion will be displayed in the upper right corner of the screen (R+). By selecting this option, all active monitoring will be switched to the Report mode or to the interactive mode. R– is the standard setting.

/V

The user may specify whether interrupt 13h, 21h, 26h and 40h **vector change monitoring** will be active (/V+), only reported (/VR) or inactive (/V–). V– is the standard setting.

/S

The user may specify whether interrupt 13h, 21h and 40h **single step monitoring** will be active (/ S+), only reported (/SR) or inactive (/S–). S– is the standard setting.

/A

The user may specify whether the FGUARD program will be **active** (/A+) or **inactive** (/A–). If FGUARD is inactive, the only categories monitored are attempts a) to read and modify disk partition table and b) to modify the disk boot sector. Pressing Ctrl+5 in the numerical section of the keyboard has the same effect. Activation and deactivation is indicated by a beep. A+ is the standard setting.

/E

The user may specify that files (without directory) contained in the FGUARD.EXC **file will not be monitored**. If this option is deselected, no exceptions will be made.

/X

This switch **prohibits** any parameter and operation mode changes upon the following start–ups or when the Hot Key is used.

/W

This switch determines that the Novell network installation will be ignored. The network drivers may then be removed from the memory, however, FGUARD is in this case incapable of monitoring the network activity (operation).

/3[+ | –]

This switch enables to the user to specify permission or prohibition of a 32–bit access to the disk in the Windows environment.

These switches can be changed subsequently by running the FGUARD program with different parameters.

Exit Codes

When the installation of the FGUARD program is completed, FGUARD returns an exit code to the operating system. This code may later be tested either by another (parent) program or by using the IF ERRORLEVEL batch file command. FGUARD exit codes can only have the following values:

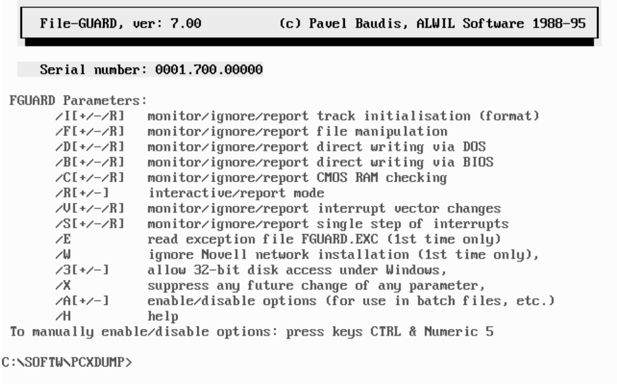
0	program installed in the memory
1	program installed earlier in the memory
2	error occurred; program cannot be installed in the memory
99	program displayed instructions for use

The exit codes thus defined may be used especially in command batches for branching at installation in the memory.

Examples of Use

```
FGUARD /?
```

The program displays simple help instructions of FGUARD.



```
 File-GUARD, ver: 7.00          (c) Pavel Baudis, ALWIL Software 1988-95

 Serial number: 0001.700.00000

FGUARD Parameters:
      /I[+/-/R]   monitor/ignore/report track initialisation (format)
      /F[+/-/R]   monitor/ignore/report file manipulation
      /D[+/-/R]   monitor/ignore/report direct writing via DOS
      /B[+/-/R]   monitor/ignore/report direct writing via BIOS
      /C[+/-/R]   monitor/ignore/report CMOS RAM checking
      /R[+/-]     interactive/report mode
      /V[+/-/R]   monitor/ignore/report interrupt vector changes
      /S[+/-/R]   monitor/ignore/report single step of interrupts
      /E          read exception file FGUARD.EXC (1st time only)
      /W          ignore Novell network installation (1st time only),
      /3[+/-]     allow 32-bit disk access under Windows,
      /X          suppress any future change of any parameter,
      /A[+/-]     enable/disable options (for use in batch files, etc.)
      /H          help
To manually enable/disable options: press keys CTRL & Numeric 5

C:\SOFTW\PCXDUMP>
```

```
FGUARD
```

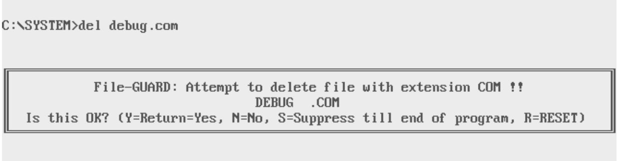
The program is run with standard parameters.

```
FGUARD /D- /C+
```

The program is run with direct write on disk with DOS de– activated, and monitoring of CMOS memory activated.

Here are some examples of actions detected by the FGUARD program:

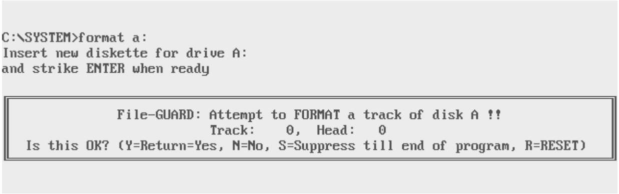
File deleted by DEL command:



```
C:\SYSTEM>del debug.com

           File-GUARD: Attempt to delete file with extension COM !!
                                DEBUG   .COM
  Is this OK? (Y=Return=Yes, N=No, S=Suppress till end of program, R=RESET)
```

In this case the user answers “Y” (Yes) if he wishes to delete the program, and “N” (No) if not.

Format diskette in the A: drive

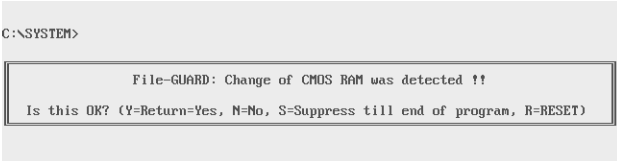


```
C:\SYSTEM>format a:
Insert new diskette for drive A:
and strike ENTER when ready

             File-GUARD: Attempt to FORMAT a track of disk A !!
                     Track:    0,  Head:   0
 Is this OK? (Y=Return=Yes, N=No, S=Suppress till end of program, R=RESET)
```

In this case the “S” response is appropriate, since otherwise the program would ask at each track format.

Change in CMOS configuration memory content:

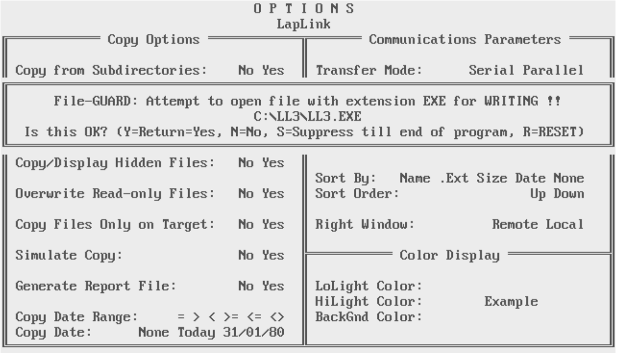
The CMOS configuration memory of AT computers contains basic information about the system, and the content of the memory is altered by the SETUP program. Under normal cir–cumstances there is no change in content.



```
C:\SYSTEM>

              File-GUARD: Change of CMOS RAM was detected !!

 Is this OK? (Y=Return=Yes, N=No, S=Suppress till end of program, R=RESET)
```

A program modifies itself

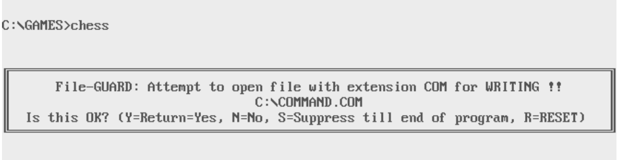
Some programs do not store their configuration in a special file. Instead they store it by modifying themselves. Any such attempt will be detected by FGUARD. For example the popu–lar Travelling Software communication program LapLink III permits modification of many parameters in a menu. At the moment the request is displayed to store the new configura–tion, the FGUARD window appears superimposed on the LapLink menu:



```
                               O P T I O N S
                                  LapLink
 ============ Copy Options ============  ======== Communications Parameters ========
  Copy from Subdirectories:    No Yes      Transfer Mode:      Serial Parallel

      File-GUARD: Attempt to open file with extension EXE for WRITING !!
                                C:\LL3\LL3.EXE
  Is this OK? (Y=Return=Yes, N=No, S=Suppress till end of program, R=RESET)

  Copy/Display Hidden Files:   No Yes
                                           Sort By:   Name .Ext Size Date None
  Overwrite Read-only Files:   No Yes      Sort Order:                 Up Down

  Copy Files Only on Target:   No Yes      Right Window:          Remote Local

  Simulate Copy:               No Yes      ============ Color Display ============

  Generate Report File:        No Yes      LoLight Color:
                                           HiLight Color:        Example
  Copy Date Range:     = > < >= <= <>      BackGnd Color:
  Copy Date:      None Today 31/01/80
```

In this case the appropriate answer is “Y”, since this is not a suspicious change.

File infection:



```
C:\GAMES>chess


      File-GUARD: Attempt to open file with extension COM for WRITING !!
                                C:\COMMAND.COM
  Is this OK? (Y=Return=Yes, N=No, S=Suppress till end of program, R=RESET)
```

In this example, the user decided to relax with a newly installed game. To his surprise, the game program attempts to interfere with the COMMAND.COM system. Without the FGUARD program the system would become infected, and from the COMMAND.COM file the virus would spread very quickly.

Besides the options “Y” (Yes), “N” (No), and “S” (Suppress checks to end of program), the user may chose the R (Reset) option, which will reboot the system immediately (as if the Ctrl+Alt+Del combination were used). This option may be used if further destructive activity could result immediately after the spread of the virus is halted. The user is prompted to insert the original system diskette and switch the computer off, then on again.

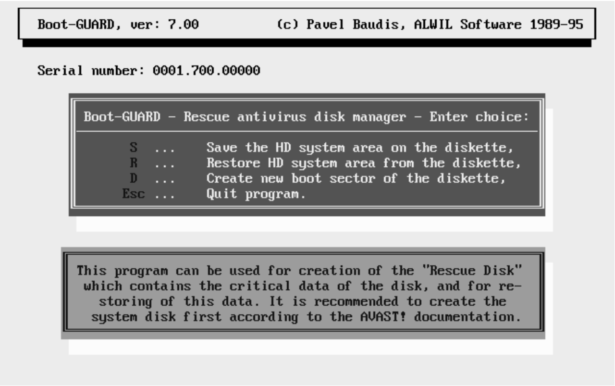
Boot–GUARD

The Boot–GUARD program occupies a special position in the AVAST! antivirus package. It is not intended for daily use, and in normal circumstances it only needs to be run once. Nevertheless it is of enormous value in antivirus protection, in that it makes it possible to store critical areas of hard disks on a diskette, which can then be used in case of necessity to restore the hard disks. These are the **system areas of the disk**, composed of the sector with disk partition table (sometimes called Master Boot Record) and the boot sectors. BGUARD should definitely be run during creation of the rescue diskette upon the installation of the AVAST! package as described in the respective chapter. Then, should these areas of the hard disk ever incur damage (most commonly from boot viruses infecting the hard disk), the original data can easily be restored to the disk. This step will completely deactivate and eliminate boot viruses from the hard disk.

BGUARD also performs the very important function of **eliminating boot viruses from infected diskettes**.

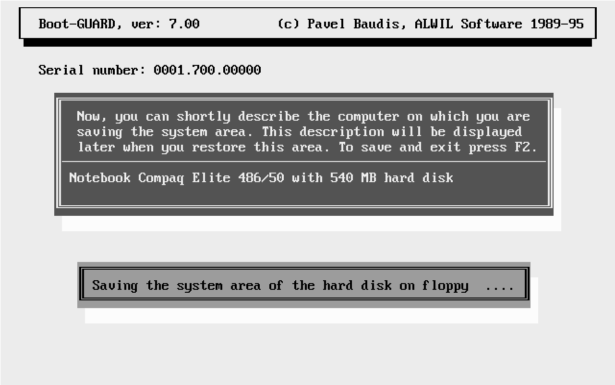
Running the Program

The BGUARD program is started from the command line in simple fashion. Further communication follows interactively upon user response. The following window appears on the screen:



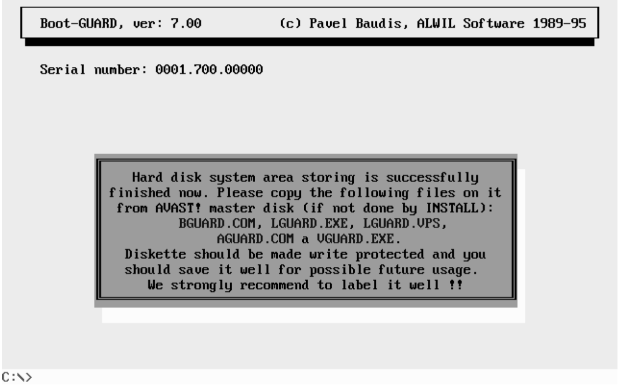
```
Boot-GUARD, ver: 7.00         (c) Pavel Baudis, ALWIL Software 1989-95

Serial number: 0001.700.00000

     Boot-GUARD - Rescue antivirus disk manager - Enter choice:

          S  ...    Save the HD system area on the diskette,
          R  ...    Restore HD system area from the diskette,
          D  ...    Create new boot sector of the diskette,
        Esc  ...    Quit program.

    This program can be used for creation of the "Rescue Disk"
     which contains the critical data of the disk, and for re-
       storing of this data. It is recommended to create the
     system disk first according to the AVAST! documentation.
```

The user may choose among three options: “S” to save the disk system area, “R” to restore the system area from the diskette, and “D” to form a new boot sector on infected diskettes. When “S” is pressed, the program asks for confirmation. Once this is given, the user must specify the drive in which he has inserted the system diskette prepared during installation of AVAST!.



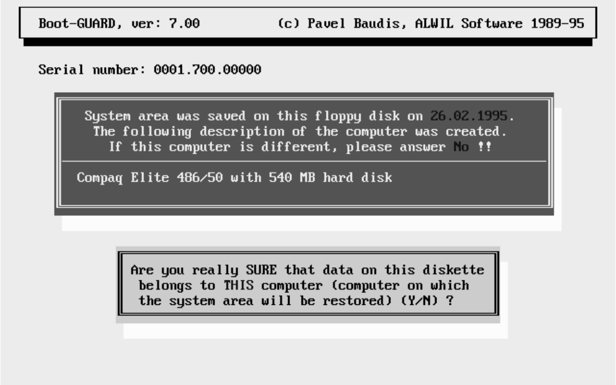
```
Boot-GUARD, ver: 7.00         (c) Pavel Baudis, ALWIL Software 1989-95

Serial number: 0001.700.00000

Now, you can shortly describe the computer on which you are
saving the system area. This description will be displayed
later when you restore this area. To save and exit press F2.

Notebook Compaq Elite 486/50 with 540 MB hard disk

Saving the system area of the hard disk on floppy  ....
```

The BGUARD program then saves the disk system area on this diskette. When the data has been successfully saved on the diskette, BGUARD prompts the user to store it in a safe place. This diskette should also be write–protected and duly labelled. Since the contents of system areas may vary with different computers (the particular configuration, disk type, number and size of logical disks, etc.), a special diskette should be prepared for each computer.



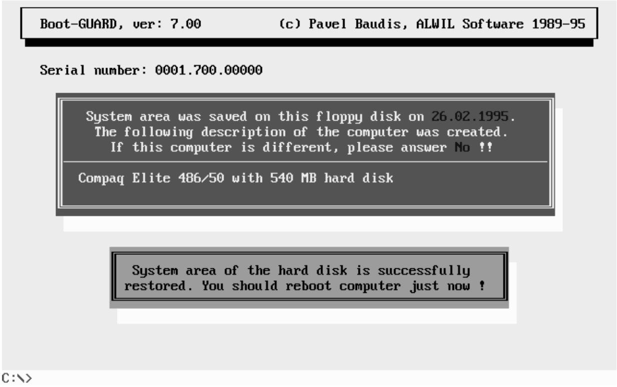
```
Boot-GUARD, ver: 7.00          (c) Pavel Baudis, ALWIL Software 1989-95

Serial number: 0001.700.00000

             Hard disk system area storing is successfully
           finished now. Please copy the following files on it
            from AVAST! master disk (if not done by INSTALL):
                  BGUARD.COM, LGUARD.EXE, LGUARD.VPS,
                       AGUARD.COM a VGUARD.EXE.
             Diskette should be made write protected and you
             should save it well for possible future usage.
                We strongly recommend to label it well !!

C:\>
```

If the system area of a disk has been altered (for example through the appearance of a boot virus), BGUARD can restore the data on the hard disk from the diskette. First the operating system must be booted from the original system diskette or from the diskette created at installation. Then the BGUARD program should be run from the rescue diskette and the option "R" (Restore) selected. BGUARD then asks for user confirmation to overwrite data on the hard disk. Then it asks whether the inserted diskette actually corresponds to the computer whose data is to be restored. Also it will display specifications that have been set upon production of your computer, together with the date of creation or last data saving.



```
Boot-GUARD, ver: 7.00          (c) Pavel Baudis, ALWIL Software 1989-95

Serial number: 0001.700.00000

        System area was saved on this floppy disk on 26.02.1995.
         The following description of the computer was created.
           If this computer is different, please answer No !!

       Compaq Elite 486/50 with 540 MB hard disk


              Are you really SURE that data on this diskette
               belongs to THIS computer (computer on which
               the system area will be restored) (Y/N) ?
```

If the active FGUARD program is installed in the computer, an attempt to write on the sector with disk partition table is reported. In this case, of course, the action is not suspicious and “Y” should be pressed. If there is more than one physical disk, all of them are restored. After the data has been restored, the system may be booted and operation can continue. The SGUARD and AGUARD programs can verify that the data is in fact identical to the original.



```
Boot-GUARD, ver: 7.00         (c) Pavel Baudis, ALWIL Software 1989-95

Serial number: 0001.700.00000

System area was saved on this floppy disk on 26.02.1995.
 The following description of the computer was created.
   If this computer is different, please answer No !!

Compaq Elite 486/50 with 540 MB hard disk

 System area of the hard disk is successfully
restored. You should reboot computer just now !

C:\>
```

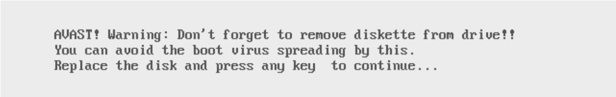
Important Note:

Whenever the configuration of the computer is changed (for example by adding a disk, changing the number or size of logical disks, etc.), **the current status must be re–saved on the rescue diskette!** This operation is usually performed with the FDISK program. Therefore the BGUARD program must be run after each such change.

Data cannot be stored if the disks are protected by an access control program such as the LOCKSUP component of our SUP protection system.

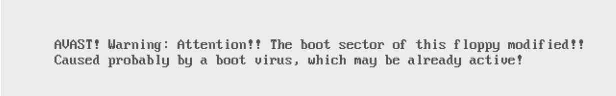
A third function of the BGUARD program is the **creation of a new boot sector on infected diskettes**. This is a task which cannot be performed by any other application of the operating system with the exception of the SYS program, which of course aside from overwriting the boot sector on the diskette also copies the system files, which is often not conven–

ient. The BGUARD program can overwrite the boot sector of **standard diskettes** and thus eliminate boot viruses without any previously supplied information. The code which it writes to the boot sector may, moreover, have a preventive effect. In case the user leaves this “treated” diskette in the drive and attempts to start the system from it, the following message appears:



```
AVAST! Warning: Don't forget to remove diskette from drive!!
You can avoid the boot virus spreading by this.
Replace the disk and press any key  to continue...
```

If at a later time such a diskette is infected with a boot virus and the system is started from it, the program can recognize the change and sends the following message:



```
AVAST! Warning: Attention!! The boot sector of this floppy modified!!
Caused probably by a boot virus, which may be already active!
```

Therefore it is advantageous to check all data disks with the BGUARD program. It can reduce the risk of infection by boot viruses and prevent possible damage.

Warning:

1) Since viruses often overwrite important parameters on the boot sector of diskettes, the BGUARD program works with standard format diskettes (360 KB and 1.2 MB with 5 1/4 inch diskettes and 720 and 1.44 MB with 3 1/2 inch diskettes, as does the FORMAT program). If the diskette has been formatted in a nonstandard way, or if an erroneous format is chosen, data might be lost from the diskette!
2) This test is not possible to use with system (booting) diskettes. With these the system command SYS must be used. **Such diskettes should always be write–protected anyway!**

BGUARD Switches

BGUARD only uses one optional switch, which determines the diskette processing:

/D

This switch determines, that the "restore" function for restoring of the system area of the infected diskettes will be initialized directly.

Exit Codes

When the BGUARD program ends, it returns an exit code to the operating system. This code may later be tested either with another (parent) program or in the command batch with IF ERRORLEVEL. BGUARD exit codes may have only the following values:

1	data restored
2	data saved
3	nothing done
4	error while working with floppy disk
5	error while working with hard disk
6	compare is OK
7	compare is not OK
99	help

Programs for Windows

It was quite recently that the number of users of the Windows graphic interface has rapidly increased. That is the main reason that led us to updating the AVAST! software by several programmes which are a full–purpose Windows applications. These programs work closely together and in some cases directly use the programs for the DOS environment.

File–GUARD for Windows

File–GUARD for Windows is a program which monitors attempts to alter files (by deleting, opening for write, renaming, suppressing the read–only attribute, etc.) in real time during a Windows session. It enables the user interactively to approve or prohibit modification or, if he wishes, to reset the computer. He can also monitor the status of the Boot sector of a diskette.

FGW works closely with the FGUARD and RGUARD programs for DOS. Without this cooperation the operation of this program in Windows might have adverse consequences (for example “deadlock” at the most inconvenient time).

Brief Description

FGUARD is intended for Windows 3.1 and newer versions and is able to work together with the DOS system. If the program is active, all activities monitored by FGUARD and RGUARD for DOS in the Windows environment (manipulation of files) or any suspicious activity (i.e. caused by viruses) are detected by the programs for DOS and sent on to the Windows system, where the information is processed by FGW.

DOS and Windows are two entirely different systems; interaction between them is no simple matter because a Windows program cannot be called up from DOS. Also some of the features of antivirus programs for DOS, such as monitoring changes in real interrupt vectors, make no sense in the pro–

tected Windows environment. Therefore if one of the programs for DOS detects a modification attempt on files, or a change in interrupt vectors, or any case in which it is impossible or unnecessary to call on FGW, such action is enabled without notification to the user.

If FGW is called up, a window appears on the screen asking the user to choose among several options. Further operation of the system is temporarily disabled until one of the options has been chosen.

If the user merely wishes to monitor the operation of the computer and does not wish to choose among the above mentioned options, he may use the Report mode, in which a report appears in the upper right corner of the screen specifying why FGW was alerted. This report appears for a period of 10 seconds with an accuracy of 0.1 seconds, depending on other programs being run, and is found in an independent window which always appears superimposed on other windows.

The operation of FGW, unlike the programs for DOS, can only be suppressed by setting the appropriate switch in the main window. In this program there is no “hot” key or warning beep. Such operating tools are not strictly prohibited by Windows, but they are inconsistent with Windows control methods, which are based on different principles.

By using the switches in the main window of the program, the user may select the activities to be monitored by FGW and the activities to be monitored on ending the Windows session.

It should be emphasized that FGW is fully dependent on the DOS programmes in the monitoring of all activities. If one of the programs for DOS fails to detect a signal or detects a false signal (i.e. incorrect file open for read), FGW may give a false alarm or fail to give an alarm.

Installation and Running the Programme

FGW is composed of two principal parts, the AVAST.386 file and the FGW.EXE file. The first, the so–called virtual driver, is a program which includes the information crucial for working together with DOS: it forms the link between the programs for DOS and FGW. The second is a standard application for the

Windows system, which contains the display routines and communication with the user.

All components of the AVAST! antivirus package which are necessary for working with Windows are automatically saved on the hard disk during installation. The installation itself is quite simple. Essentially it is sufficient to respond affirmatively to the appropriate question and the installation programme copies the necessary files by itself. After completion of the installation, it is necessary to run AVINST programme from the Windows environment.

During Windows start–up, then, FGW will automatically be the first program started, which we recommend as the ideal arrangement for its use. We further recommend that it be left active until the end of the Windows session—in fact that no action ever be taken to terminate it. However, it is inevitable that the FGUARD or RGUARD or both are present in the computer memory, otherwise, the FGW program will report an error message and fail to start up.

FGW also requires other files for its operation: the virtual driver (AVAST.386 file), files with program Help and several libraries. These files are automatically copied on the disk in the directory in which they should be placed.

FGW may be run in one copy only. An attempt to run a second copy will result in a warning tone, and the main window of the first copy will appear on the screen.

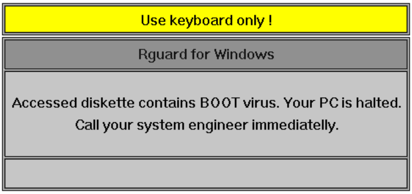
Command Line Parameters

During start–up the program inspects the command line, looking for parameters beginning with the characters "/" or "–" which may be written in upper or lower case characters. FGW version 7.00 supports one parameter on the command line which concerns following the activity of the RGUARD program and allows blocking of tasks which will elicit a reaction by RGUARD. This parameter is:

/SB Blocking a task involving entry on a diskette containing a boot virus.

/SE Blocking a computer if the programme currently run contains a virus.

If selected these parameters display messages stating the fact, and the Windows system takes no further action. But the system is not entirely blocked, so that the user may take steps to eliminate the cause of the warning.



Use keyboard only !

Rguard for Windows

Accessed diskette contains BOOT virus. Your PC is halted.
Call your system engineer immediatelly.

Standard and Extended Mode

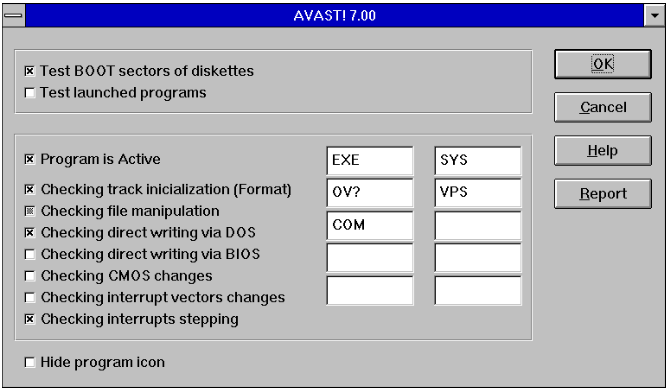
FGW supports the extended mode of the Windows operating system. It cannot be used in the standard mode. An attempt to run it in standard mode will result in a message stating the fact and the program will end. But this does not mean that while working in standard mode you cannot protect your DOS programs. The protection can operate in all its functions without problems. Unlike the extended mode, in the standard mode the program for DOS will switch the screen to the text mode when an event is detected, wait for the user's response, then switch back to the Windows system. After returning to the graphic screen of the Windows system, however, it is necessary to restore the screen, for example by maximizing a window then returning it to its original size. In extended mode FGW performs these operations automatically in the appropriate windows and communicates with the user in the normal Windows manner.

Program Control

Controlling the program is very simple. Any user who knows how to use Windows will be able to deal with FGW as well. The program is controlled in the same way as Windows: the mouse, the keyboard, or a combination of both.

Once the program is loaded, it appears as an icon in the bottom line of the screen. To make any changes in settings the program window must be called up by double–clicking on this icon. The main window appears and the settings can be changed.

The program's main window is divided into several areas. The upper portion contains RGUARD parameters, the lower portion those for FGUARD, and on the right and bottom edge are the normal Windows keys:



☒ Test BOOT sectors of diskettes
□ Test launched programs

☒ Program is Active
☒ Checking track inicialization (Format)
▨ Checking file manipulation
☒ Checking direct writing via DOS
□ Checking direct writing via BIOS
□ Checking CMOS changes
□ Checking interrupt vectors changes
☒ Checking interrupts stepping

EXE	SYS
OV?	VPS
COM	

□ Hide program icon

Setting Parameters for RGUARD

RGUARD works in two areas which may be installed independently. They are represented by two keys in the upper portion of the main window. If the appropriate part of RGUARD for DOS is installed in the memory, its key may be selected. If not, the key and its description are displayed as inactive (dark grey) and it may not be changed. Active keys may be selected or deselected. They are selected when the box is checked.

Setting Parameters for FGUARD

FGUARD parameters are generally set in the main window of the program. The basic key is placed first, by which FGUARD is globally selected or deselected. When the box is checked, the program is active and performs the selected functions; when blank, the program ignores all further parameters and is inactive.

Each switch for monitoring activity may be set to three positions. A blank box means monitoring switched off, a checked box means active monitoring, and a key displayed as inactive (in grey) means report mode. In addition a global key may be used for report mode, which switches all active keys to report mode, or all report keys to the active mode.

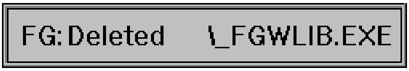
The keys may be set using the mouse or the keyboard as in all other programs for Windows. The meaning of the various settings is described in detail in the section on FGUARD for DOS.

Parameters of the FGUARD program may also be set using the editing boxes in which the user may select file types.

Report Mode

If the user only wishes to monitor operation of the computer and does not want to respond to the questions, he may use the Report mode, in which a report about monitored activity appears in the upper right corner of the screen. The report appears in an independent window superimposed on the other windows of the program being run and is displayed for approximately ten seconds.

The Report mode can be set for individual types of monitored activity separately or for all active monitoring globally using the report key:

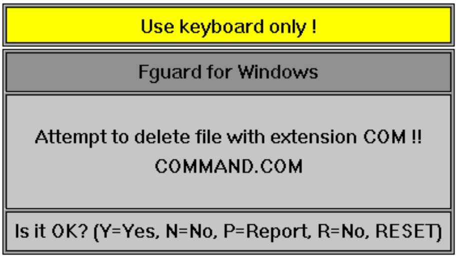


FG: Deleted   _FGWLIB.EXE

How to Respond to Warnings

If a monitored event occurs, a warning appears in an independent window. Since the warning was caused by a program for DOS (a program operating in the real mode of a 386 processor, during the display the system is interrupted: no other program functions, and the system waits for the user's response. To continue with his work, the user must press one of the keys listed at the bottom of the window. According to the response, the program will then permit the event, prohibit it, switch to Report mode or reset the computer. The upper portion of the window shows the name of the program which caused the warning, and in the main part there is a description of the warning.

FGW excludes the FGUARD for DOS option “Suppress checks to end of program” because of the multitask character of the Windows system. FGUARD and RGUARD reports may have the following form:

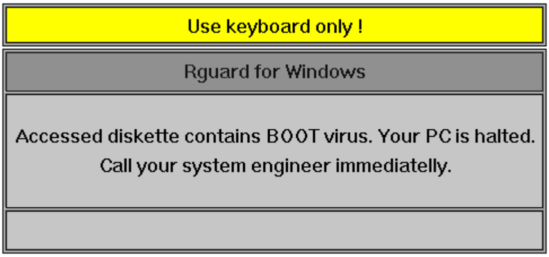


Use keyboard only !

Fguard for Windows

Attempt to delete file with extension COM !!
COMMAND.COM

Is it OK? (Y=Yes, N=No, P=Report, R=No, RESET)



Use keyboard only !

Rguard for Windows

Accessed diskette contains BOOT virus. Your PC is halted.
Call your system engineer immediatelly.

How to Respond to Warnings with Virtual Driver

The virtual driver, the AVAST.386 file, represents the basic Windows support. Its main task is to link the FGW.EXE program with DOS. If the FGW.EXE program is not installed in the system, the virtual driver takes over responsibility for warning the user. It uses the standard display for the Windows system intended for this purpose. An example of this message appears when you attempt to reset the computer from the keyboard in Windows 3.1 In detecting this event AVAST.386 is able to display the warning in the proper language form (that set for the program in DOS) and relay the message (of course in the text mode). The user, however, must respond with the standard Windows keys which vary according to the version of the system. In the English version the following responses are possible:

Y	Yes, allow, OK
N	No, prohibit, not OK
Esc	Reset computer
Enter	same as Y

In other versions other keys are used for the same responses.

During message display all other computer operation is suppressed. Thus not even the mouse functions. After the response is given, the content of the graphic screen is restored and the application functions according to the response.

Other Parameters

If you confirm parameter settings with the “OK” button or cancel them with the “Cancel” button, the main window of the program minimizes itself automatically to an icon. The icon’s form depends on whether activity is monitored actively or not and on whether both FGUARD and RGUARD for DOS are installed. Rules for the icon form are as follows:

F	FGUARD for DOS installed
R	Only RGUARD for DOS installed
closed lock	Program is active
open lock	Program is not active

Ending the Program

FGW is intended for continuous operation during the entire Windows session. If desired, the program may be eliminated from the memory. In this case, however, the possibility of interactively changing the program settings is lost. Any monitoring of activity selected at the ending of the program will also be active subsequently, and for messages the program will use the system message whose form can be seen for example during an attempt to reset the Windows system.

Versions incompatible. The program is unable to operate due to incompatibility with the supporting library (AVAST.386).

In order to operate, AVAST needs a library, which is supplied together with the used versions of EXE file or in some cases you may use a newer version of the library. The currently installed version is not compatible with the FGW program. The error was caused by incomplete installation or manipulation with the installed files. Repeat installation.

Program not installed. – FGUARD or RGUARD for DOS are not present in the computer memory. – the FGUARD or RGUARD and FGW are not mutually compatible. End the Windows system and run FGUARD or RGUARD.

In order to operate properly, FGW needs to have FGUARD, RGUARD or both programmes for DOS installed in the memory. If none of these is installed, FGW is unable operate. Another reason may be the incompatible versions, which might be caused by incorrect installation of the system only.

Internal DPMI error. Unable to allocate structures necessary to access the data. The program unable to operate. Recommendation: End your work in Windows.

FGW discovered an error in working with very sensitive structures of the protected mode of Windows. The only sense–making solution would be to save the data on a disk and terminate your Windows session.

Data modification internal error. The displayed data does not correspond to the data considered for the system protection.

FGW detected an error in saving the changes to internal data of the DOS resident programmes. There is a high risk that the saved data overwrote the important parts of other programmes and computer deadlock may occur.

Program initiation error. No timer necessary for operation is free. Please close some application and try again.

FGW needs a timer for its operation. The number of timers is limited. Without a timer, the program is unable to operate.

Error in creation of the REPORT window. The report will not be displayed.

FGW is for some reason unable to create the REPORT window with a particular message. The error occurred in the Windows system.

Memory allocation error. Unable to allocate DOS memory. Monitoring of the run programmes deactivated.

In order to operate, FGW needs to allocate memory on linear addresses, which correspond to the real mode of the processor operation. This memory is not large, but it was impossible to allocate it.

The programmes run will not be tested. The testing library (AWANTI.386) is not installed.

In order to execute antivirus testing, the supporting library AWANTI.386 needs to be installed. If not installed, or its installation fails, the programmes run cannot be tested.

Extended mode detection error. The program is not able to operate in a different mode. Please terminate your Windows session and use the “WIN/3” command.

FGW is able to operate only in the extended mode of Windows 3.1 or higher. If this mode is not activated, FGW will not run.

Locate–GUARD for Windows

Locate–GUARD for Windows is a locating antivirus program (of "scan" type) specially designed for use with Windows. It detects the same viruses as LGUARD for DOS and functions identically. Both programs are capable of finding a large number of known computer viruses including many mutations and polymorphic types.

LGW is primarily designed for continuous monitoring of the system on which it is run, but the program may of course be used for regular or irregular testing of your computer without the necessity to keep it operating on a constant basis.

The program is an implementation of the original LGUARD program designed for DOS into the Windows operating system taking advantage of a number of improvements and benefits, which this system offers. At the same time, however, it guarantees compatibility on the level of the number and types of detectable viruses and usage of the same parameters in both systems. The testing procedure is the same in principle but the actual implementation of the program utilizes all advantages of the 80386 processors, for which LGW is designed.

The program is designed in such a manner, so that it blocks the rest of computer operations to the least extent, and the aspect of its size and memory occupation comes only second. The program continuously monitors the activities of the mouse and the whole system. The discovered delays are used for its own operation, the user being unable to notice any delays of the rest of the applications.

Installation and Start–up

LGW, like the other programs in the AVAST! antivirus package, is installed with a program which is part of the package. The installation is ensured by AVINST program which creates the AVAST! group, from which the program may be run. In its operation the program uses the same database as LGUARD for DOS (LGUARD.VPS), containing virus definitions. This database must be placed in the same directory as the program itself. The program is run in the same manner as other programs for Windows. It does not recognize any param–

eters of the command line. For storing working parameters it uses the same configuration file (AVAST!.INI) as the other ALWIL Software programs for Windows. If this file does not al–ready exist, it will be created by the program upon start–up in the same directory from which it was started.

In connection with the set parameters, the program may start from two environments: the normal window environment or else in the background as an icon. A more detailed explana–tion of both possibilities will be found below under the descrip–tion of the program's parameters.

LGW can be run in one copy only. If there is an attempt to run a second copy, a warning beep sounds, and the main win–dow of the first copy is displayed on the screen.

What can be Tested with LGW

LGW is able to test all parts of your system one after the other, or independently according to the options chosen. For testing purposes the program distinguishes four areas:

- the system's operational memory
- the disk partition table (DPT) sector of the hard disk
- the boot sector of the individual disk
- the files on the individual disk

Each of these areas may be tested individually, or the user may select that all will be tested successively. If the first op–tion is selected, the program tests the area specified, then ends. If the second option is selected, all will be tested, beginning with the area selected. For example:

- memory
- DPT
- disk C:
- BOOT sector
- files
- disk D:
- BOOT sector
- files
- ...
- memory
- and so forth.

System memory

LGW tests 1024+64 KB of operation memory, which begins with linear address 0. This is the memory in which are found the DOS operating system and all memory resident programs and where critical data for the Windows system is stored, which must be available from DOS. Should you have further memory (and you surely do), this is not tested by the program, because no known viruses uses it, nor can any virus spread in it.

If you use a DOS window, its memory is emulated in two principal areas. The area common to all DOS windows is an area in Windows which contains DOS and resident programs installed before running Windows. The second area is created from the address space of Windows and is generally not found in a space of linear addresses under 1 MB. This second part of the DOS window memory is not tested. Therefore, if you should run an infected program here, LGW may not detect the virus in the memory.

DPT of Hard Disks

The program is only able to test two hard disks connected to the system in the standard way. If your system does not have two hard disks, the program recognizes the fact and tests only the one disk. LGW tests a maximum of two hard disks at once before testing the BOOT sector of disk C:.

Boot sector of Individual Disks

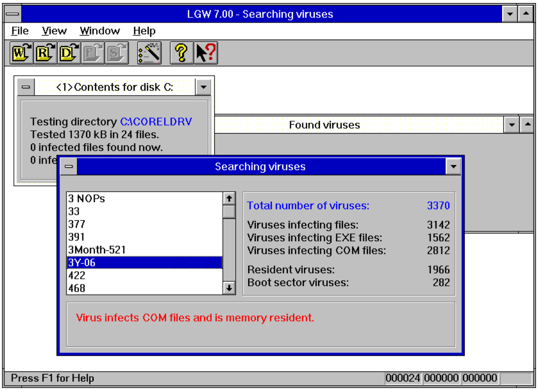
LGW tests the BOOT sector of each disk placed in the drive before testing its files. The program does not test the BOOT sector for RAM disks.

Files on Individual Disks

LGW tests files of individual disks according to the options selected. More detailed information about this is given below. The program is capable of testing files on all disks recognized by the operating system, including remote disks (Novell, ...). The program does not test files on CD–ROM disks, but the user may request such testing.

Program Windows

LGW is a so–called MDI application, which means that within the main window, several subsidiary windows (with varying content or for various purposes) may be displayed which are completely independent. The LGW program enables you to open any number of independent windows (the limitation being the local memory of the program where the data for these windows is stored), consisting of several different types. These are as follows:



Testing directory C:\CORELDRV
Tested 1370 kB in 24 files.
0 infected files found now.
0 infe

3 NOPs
33
377
391
3Month-521
3Y-06
422
468

Total number of viruses:	3370
Viruses infecting files:	3142
Viruses infecting EXE files:	1562
Viruses infecting COM files:	2812
Resident viruses:	1966
Boot sector viruses:	282

Virus infects COM files and is memory resident.

Press F1 for Help 000024 000000 000000

Working window

a window in which a selected sector of the system is tested

Report window

a window in which the REPORT file is displayed.

User definition

a window in which the user may edit the definitions of a virus

Detected viruses

a window listing viruses found

Viruses searched

a window listing viruses recognized by the program.

Of these five types, all except the working window may be opened in only one copy. The number of working windows is limited in practice only by the amount of local memory available (not the general memory of the system), and during normal operation of the program between three and five windows are used. Each window may be reduced to the form of an icon without influencing its effectiveness.

Working Window

A working window serves to report on the course of the testing. The window displays the directories, which were tested. It also includes the number of viruses detected in the respective window as well as the total number of viruses detected upon the given program start–up in all windows.

The working window is a basic part of the program. The first working window is opened at the start of the program and carries the number 1. Each of the working windows shows independently the course of the testing of a specific part of the system. Individual windows may test different areas and the same area at once (though testing the same area simultaneously only makes sense if you want to delay the processor).

How to Open a Working Window

A working window may be opened by selecting the appropriate option from the menu or by clicking on the appropriate key. In creating a window the program must form an independent working space within its local data. It may happen that at a given moment the program does not have sufficient space available and therefore does not create the data. In such a case the program informs the user and cancels the request.

Opening Further Windows

When the user elects to open another working window, the program assigns it the lowest free number that it finds. It does this by scanning the list of all working windows and determining the first free number. If for example the program is using three working windows, created one after another, it assigns them numbers 1, 2, and 3. If working window 2 should be deleted (leaving only windows 1 and 3 functioning), a new working window would be assigned the number 2. Each working window is absolutely independent of the others even though they use the same reference material for testing, the same configuration data for selecting positions for testing and the same operating routine. The program contains no means to enable one window to interfere with the data of another window.

Working Window Number 1

Window number 1 is created automatically at the start of the program. It is the one case in which a window is created without prompting from the user, and it draws its information from the configuration file (for example the types of files tested, possible attributes, etc.).

If the program closes window 1, it saves the state of the testing procedure in the configuration file. The program does this for each window number 1 (even if it is repeatedly closed and recreated). No other window records the state of its testing in this way when it is closed. It is important to realize that when you close working window 1 and then end the program, the next time the program is run, only that portion will be tested and the program will not test successively.

Only the working window automatically created at the start of the program is capable of continuing with the work according to the configuration data saved at the last ending of the program.

User–Defined Viruses

AVAST! comes with the LGUARD.VPS file, which describes individual viruses. This file is frequently updated, but even so the program offers the possibility for the user to create his own file of virus characteristics. The User–Defined Virus Window enables the user to display and edit this file.

The User–Defined Virus file may contain an indefinite number of virus definitions. Its content interpretation follows the rules as specified for the LGUARD program in the respective chapter.

If a character string is found to be invalid, the program continues interpreting the file until it finds a string which might be valid in the given place. This feature of the program may, in certain cases, lead to different virus definitions being mixed together. Therefore we urge great caution when using the user–defined virus file! Examples of virus definitions are found in the LGUARD program description.

Detected Virus Window

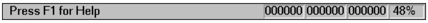
During testing the program may detect some viruses (which is, after all, its purpose). LGW reports each virus found in the memory and the first virus found in the files. A list of the viruses detected may be seen in the Detected Virus Window, which also reports where they were found. In contrast to the Report file, the content of this window is renewed during operation whenever the program finds a virus in the tested parts of the system. The window's contents cannot be edited or modified.

Viruses Searched Window

This window displays all viruses which the program is able to detect. The real number will of course be higher, since the list does not include all modifications. The window also shows some statistical data on the composition of the virus database, which enables the user to get information about each virus by selecting its name. The window's contents cannot be edited or modified, nor can its size be changed.

Status Bar

The program bar serves as a continuous display of important information during the running of the program and is located in the bottom part of the main window. It can be deleted by selecting the appropriate item in the menu.



Press F1 for Help | 000000 | 000000 | 000000 | 48%

The left side shows the currently selected menu option and changes as other options are selected. This space also shows what will happen if the key is pressed. The key description remains active until you click the mouse button. If you wish to cancel the operation caused by clicking the mouse, click again when the mouse pointer is not on the bar.

The right side of the bar shows the following program parameters:

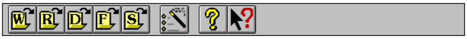
Indicator	Explanation
1	total number of files tested
2	total number of infected files
3	total number of virus types found
4	percentage of selected testing completed for active working window.

Program Control

This program is run in practically the same way as any other program for Windows. All standard actions carried out with mouse or keyboard can be done with LGW.

For quick supplemental help with some of the program's features, we have included the "bubble" Help (Tool Tips) which describes the work bar.

All program features may be accessed from the program menu, which changes depending on whether one of the windows is open and on which of the open windows is active. The main functions may be selected from the work bar which is displayed just below the program menu. The various keys have the same value as the corresponding menu entries. The keys that are inactive at any given moment are shown in light grey and cannot be selected.



The keys on the work bar are as follows, from left to right:

- open a new working window; LGW displays a dialogue box to determine the parameters for this window.
- open REPORT window,
- open user–defined virus window,
- open virus detected window,
- open virus search window,

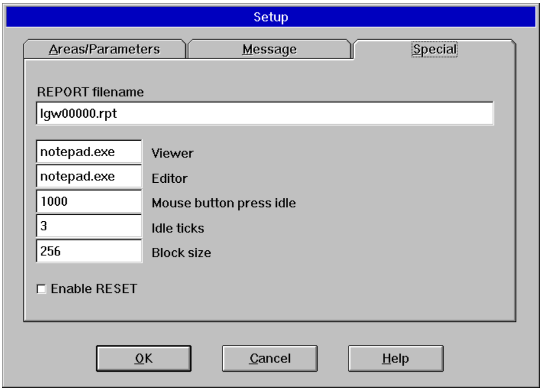
- program configuration; LGW opens a dialogue box in which all program parameters can be configured,
- open window with copyright information and virus character–istic database
- change cursor for help with individual program features

Setting Program Parameters

LGW does not recognize any parameters on the command line. All necessary program parameters are read from the configuration file AVAST!.INI. If this file does not exist, or if some of the parameters are not present in the file, the standard settings are used, which in the great majority of cases guarantees the optimal functioning of the program.

LGW has two places where working parameters may be requested. They are the configuration dialogue box and working windows 2 and higher.

Configuration Dialogue



REPORT filename
lgw00000.rpt

notepad.exe Viewer
notepad.exe Editor
1000 Mouse button press idle
3 Idle ticks
256 Block size

□ Enable RESET

The configuration dialogue serves to set the program's working parameters. These are divided into several groups. Incorrect settings may cancel the operation of the program. Therefore it is essential to exercise caution in setting the parameters.

The dialogue box uses all the standard means of control by keyboard or mouse. It is controlled by the "OK" and "Cancel" keys as follows:

The "OK" key sets the parameters as they appear in the dialogue box, and the box disappears. The "Cancel" key cancels all changes to the configuration parameters, and the box disappears. Clicking on "Help" brings help for the configuration dialogue. "Esc" has the same effect as "Cancel".

The following parameters may be used for configuration of the program:

Test memory

When selected, the memory will be tested from linear address 0 to 0x10FFFF, which represents real mode memory (DOS – 1 MB) and HMA (64 KB). If the program receives a command to test the memory and this option is not selected, the memory will not be tested. The program automatically generates a request to test the memory before testing the C: disk. During start–up the program tests the memory even if this option is deselected.
Default: Selected

Test DPT

When selected, the DPT of the hard disk is tested. Since the size of the DPT on a hard disk is 512 bytes, both disks are tested at once. If the system has only one hard disk, the program determines the fact and tests just the one disk. If the program receives a command to test the DPT and this option is deselected, it will not be tested. The program automatically generates a command to test the DPT after the memory test, even if the memory test has been omitted. If it is unable to read the DPT, the program shows an error message and continues running.
Default: Selected

Test BOOT Sector

When selected, the Boot sector of each disk will be tested. If the program receives a command to test the Boot sector and this option is deselected, it will not be tested. The program automatically generates a command to test the boot sector before testing files on the given hard disk. If it is unable to read the Boot sector, the program shows an error message and continues running. The program does not test the BOOT sector of the RAM disk.
Default: Selected

Test files

When selected, files on accessible sectors of the system disk will be tested. If the program receives a request for testing files and this option is deselected, the files will not be tested. The program tests files on local and remote disks available to the user at the moment and distinguishes disks C: through Z:. The program ignores any CD–ROM disks that it may encounter. If when starting a new window the user requests disks A: or B:, the program tests the disk in the appropriate drive.
Default: Selected

Test network files

This option ensures that the testing status of network disks will be promptly switched, while the program automatically recognizes the disk type. If selected, and if the local disks files are tested, the remote disk files will be tested as well. If deselected, the remote files will not be tested.
Default: Selected

Test SYS files

When selected, files with the SYS extension will be tested. If deselected, these files will be ignored.
Default: Selected

Test HID files

When selected, files with the HID extension will be tested. If deselected, these files will be ignored.
Default: Selected

All files

When selected, all files will be tested whatever their extension. When deselected, only files with extensions COM, EXE, SYS, BIN, and OV? will be tested.
Default: Deselected

Full Files

When selected, complete files will be tested. If deselected, only the first and last 8191 bytes will be tested.
Default: Deselected

Ignore virus target

When selected, the program will test for all viruses in all files without regard to the type of file or virus parameters (i.e. including Boot viruses). When deselected, COM (EXE) files are only tested for viruses that can attack this type of files. Other types of files are tested for both viruses.
Default: Deselected

Write REPORT

When selected, information will be saved in a REPORT file. In this case the file contains information about detection of a virus or other important information including a statistical report on the system test. If deselected, nothing is saved in the REPORT file.
Default: Deselected

Create new REPORT

If selected, a new file will be created upon saving the REPORT file during the program start–up or upon the change of configuration data. The program au–

tomatically generates its name according to the name of the last saved file. If deselected, the data are saved in the existing file.
Default: Deselected

Write also non–infected files

If selected, the program will write into REPORT file also files, in which it does not detect the presence of known viruses. If deselected, the program does not write the non–infected files.
Default: Deselected

Work in Background

When selected, the program automatically changes to an icon on start–up and works with low priority in the background. For determining priority see the parameters Operation Frequency and Number of Unused Ticks. When deselected, the program will operate in the foreground as a standard window with high priority. Selecting this option influences program start–up. While running it is possible to switch between icon and normal window as long as the program runs as an icon with low priority.
Default: Deselected

Report of detected virus

This editing window contains the information which will appear on the screen when a virus is found. This report appears if a virus is found in the memory and also when the first virus is found in the other tested system sectors. This report may be edited and replaced with another text. The text is displayed in the Virus Report Window, along with the sector where the virus was found and the virus name. What is displayed is what the user selects in this window. The report of detected virus may also be provided with a voice output, if the current installation of the Windows system supports it. The configuration of

the sound report can be set in a standard manner in the control console.
Default: “Virus Detected!!!”

Enable RESET

When selected, the user may reset the computer. When deselected the user may not reset the computer by clicking on Reset in the Report of Virus Detection box. If this is tried, a warning appears. Of course the program cannot in any way influence the hardware reset button or counteract turning the machine off.
Default: Deselected

Name REPORT file

The editing windows enables the user to name the REPORT file. The program automatically generates new names according to other configuration parameters. From the specified value, the parameters valid are the creation path, first three characters of the name and the type (extension) of the file. The remaining five characters of the name are reserved for the program.
Default: “lgw.rpt”

File Scanner

An editing window, which enables to determine the name of the programme together with a list of its parameters, which will be run if you wish to display the REPORT file. The name of the REPORT file is not specified, the program uses the current name.
Default: “notepad.exe”

Editor

An editing window, which enables to determine the name of the programme together with a list of its parameters, which will be run if you wish to display or edit the User–defined virus file. The name of the

file is not specified, the program uses the current name.
Default: “notepad.exe”

Duration of Dwell in Tests

The editing window enables to set how long the program will wait after each mouse movement. The program monitors the intensity of user’s work session adjusting its own operation to minimize the disturbance to the user’s activity. The dwell duration is set in milliseconds.
Default: 1000

Number of Unused ticks

The editing window enables the user to set the frequency of generating testing reports in case the Windows system awaits user input. If the Windows system awaits user input and is performing no operation (even in its own environment), a system event with a continually increasing number is called up. If at any time Windows is processing some event, this numbering is regenerated from the number 0. The number appearing here represents the number of called up events which will be ignored. If the number 0 appears here, the program will generate a testing report whenever the Windows system has nothing to do.
Default: 3

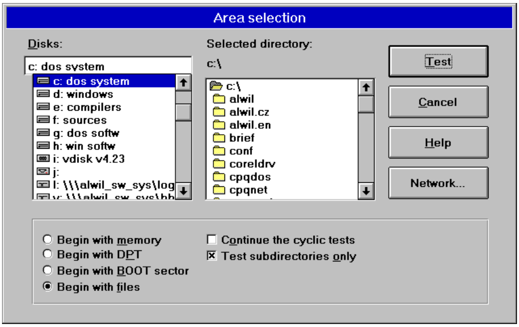
Size of the Tested Block

The editing window enables to assign the maximum size of the block in bytes, which will be tested without any interruption. This parameter serves to eliminate troubles with dwells (delays), when the computer does not respond to any user’s input. The values of 16 and around this number should be suitable even for the slowest systems and the value of 512 and above is suitable for the systems mounted with the 486/66 processor and above. Reducing this parameter will cause slow down of the testing speed. The “0” value denotes that the whole block that was read is tested all at once. The value different from “0” means that the read block of data is tested in parts with maximum size assigned in this editing box.

Default: 512.

New Working Window

The dialogue for choosing sectors for testing appears whenever the user requests the creation of a new working window. It serves to set the parameters for this window and only in the event of its successful completion can a new working window be created. The dialogue box uses all standard means of commu–nicating information by keyboard or mouse. The “Esc” key has the same effect as the “Cancel” key.



Area selection

Disks: c: dos system

- c: dos system
- d: windows
- e: compilers
- f: sources
- g: dos softw
- h: win softw
- i: vdisk v4.23
- j:
- l: \\\alwil_sw_sys\log

Selected directory: c:\

- c:\
- alwil
- alwil.cz
- alwil.en
- brief
- conf
- coreldrv
- cpqdos
- cpqnet

○ Begin with memory
○ Begin with DPT
○ Begin with BOOT sector
● Begin with files

□ Continue the cyclic tests
☒ Test subdirectories only

Pressing the “OK” key saves all parameters to the program’s internal data and their transfer to the new working window. The dialogue box disappears from the screen. Pressing the “Cancel” button can–cels all changes made in the parameters and cancels the new working window. The dialogue box disap–pears from the screen.

Test memory, Test DPT, Test BOOT sector, and Test files

The keys for testing memory, DPT, Boot sector and files serve to choose which area of the system will be tested as the first (or the only area) in the new working window. For the "Test BOOT sector" and "Test files" keys, the locations are determined from the editing box below containing the names of the disk and directory.
Default: "Test files" selected

Also test further sectors

If selected, other system sectors will also be tested along with those chosen. The program will test the selected sectors first. If the user chooses to test a directory, the content of this directory will be tested as the program reads the directory structure of the given hard disk (i.e. the information it receives from the DOS system). All directories placed before the chosen directory will not be tested, and no message about this fact will appear. As long as this option is deselected, the working window will test only the selected part of the system and upon completion of the test the window will close automatically. The automatic closing of unneeded windows speeds up the program and reserves more memory for further windows.
Default: Deselected

Test only subdirectories

This option enables to test only that part of the directory tree related to a selected directory. After completion of testing this area, the working window will be cancelled in a similar manner as if only one area is tested.
Default: Deselected

Help

LGW provides a contextually sensitive Help which reacts to the standard F1 key.

If you need help with any of the pictured elements, use Shift–F1 or the appropriate key on the command bar, which changes the shape of the cursor. As long as the cursor is changed, it is sufficient to click with the left mouse button when the cursor is on the appropriate element, and the correct Help will appear.

Messages, Error, Warnings and Queries

The Working window No. X was closed due to error in program. The error was specified by the previous report.

One of the working windows was closed due to unspecified error. It is an informative report, which was specified by another report with a more precise error definition.

Unauthorized computer reset. If you need this authorization, you must manually select the respective option in the parameter setting dialogue.

Resetting the computer is an activity, which must be authorized in the configuration data of the programme. If this report is displayed, the user is currently unauthorized to perform the RESET function.

The REPORT file NAME does not exist. Unable to open its window. If you want the program to write the REPORT file, you must select the respective options and assign the file name in the setting parameter dialogue.

The program is unable to display the non–existing file.

Your virus definition file is too old. Please check with your supplier about the upgrade option.

If you have used your LGW program for too long, the virus definition file may be outdated. If it is older than six months, LGW displays this report. Please check with your supplier for the newest upgrade.

Nothing to test. The programme settings indicate that the program has nothing to test. If you need to test a certain part of the system, change the program settings.

The configuration data setting shows that the program has nothing to do. In this condition, it only sits in the memory, thus being useless. Change the configuration data settings or end the program.

The definition file is not in use. The user definition are specified in the SYSTEM.INI file and currently are not used. I will use NAME.

At the moment of selecting editing the user virus definition file, no file is defined. A new file is created.

The configuration file is not saved. The program detected an error in saving the changed parameters to the disk. The changes are not saved and accepted.

The changed data are not saved in the configuration file. The error was not caused by LGW program.

Overwriting the content of the REPORT file. If you use the editor instead of the scanner, do not overwrite the content of the REPORT file because some collisions could occur resulting in LGUARD structure overwriting.

This warning appears if an attempt to display the REPORT file is made. If you use the editor to do so, the manual data writing may damage the file structure thus causing data losses.

Error in writing into the REPORT file. The program detected an error in writing into the REPORT file, which it is unable to remedy. The data storing is deac–tivated. You may retry the procedure by selecting the respective option in the configuration dialogue.

If an error occurs in writing the data in the REPORT file, its updating is immediately deactivated, but the program continues in testing. The most likely cause of this is insufficient space on the disk.

Testing the file within the network. The network files form a subset of the accessible files. It is impossible to test the network disks files and have the option of testing the files in general deselected at the same time.

If you request that network files be tested, you must select the option of testing all files in general. Change the configuration data setting.

The NAME file was deleted. You cannot work with a nonexistent file. If you deleted it erroneously, we are very sorry, but the file is gone forever.

It is a naked truth that working with deleted files is an uneasy task to tackle and under LGW it is even an impossible one.

File renaming error. Unable to rename the file to the same name. The operation is useless.

Truly, the existence of two files with the same name is impossible.

Program is unable to create the internal data. Some other data, which occupies space in the local memory is necessary for operation of the programme. The programme does not have sufficient space to create them. Your request is not accepted.

This report displayed as a response to the require–ment of opening another working window. The re–quirement cannot be met.

Insufficient memory. The program does not have sufficient memory space. The required operation may not be executed and is cancelled. If you insist upon its execution, close some window of the programme and try again.

If you really insist upon your requirement, follow the above advice.

Data display error. For some unknown reason, the program is unable to display the necessary data. The window which caused the error will be closed.

The error source cannot be specifically located. By all means, the data could not be accessed and displayed.

REPORT file opening error. The NAME file does not exist or is non–accessible. It cannot be displayed.

Unable to display the REPORT file. It either does not exist or another software blocks the access to it.

Unable to allocate the timer, which is necessary for the programme operation. Close some application and try again.

The program needs a timer for its operation. The number of timers is not limited. If you do want to start LGW, you must close some of the run programmes.

Error in reading DPT. Under general circumstances, you can read DPT, but you do not have this option on the network or CD–ROM discs. There is also hardware or software which prevents reading of DPT. DPT will not be tested.

The access to DPT might be very difficult.

Error in reading BOOT sector of disk X. Under general circumstances, you may freely engage in reading the BOOT sector, but there is certain hardware or

software which prevents reading of these disk areas. BOOT sector will not be tested.

The access to the BOOT sectors of the disks may prove very difficult.

Error in opening the NAME file. The program detected an error in the attempt to open the file. This file will not be tested. Continue in work?

If the file is discovered, but it cannot be opened, the program displays this report. It is up to the user to decide whether to continue to test other files or not.

Error in writing the NAME file.

A number of errors may occur upon writing the file, the exact source of which is difficult to identify. By all means, check the size of the free space on the disk.

Access to the NAME file is prohibited.

There is a software which disables access to the selected files. If you do not use such software, check the disk integrity.

NAME file reading error. The program detected an error in reading the file.

A number of errors may occur upon reading the file, the exact source of which may be difficult to identify. Check the disk integrity.

Insufficient memory for the requested operation. The program does not have enough of local memory to execute the requested operation.

The main storage area is not equivalent in 16–bit Windows. It is divided into local and global ones. The size of the global memory may be influenced by closing some of the running applications, by increasing the virtual memory or by expanding RAM.

NAME file creation error. The program detected an error in an attempt to create the requested file.

The LGW program needs to create a file from time to time. If it fails in doing so, it means that it found an error in access to the disk. Check the disk integrity.

Programme start–up error. Unable to run the NAME file. It is very likely that this program does not exist or you do not have access to it.

The right to access the program may be taken away by another software application such as SUP for instance, or the network operating system. Check whether the given programme exists and whether it is accessible.

BOOT diskette X reading error. Under general circumstances, you may freely engage in reading the BOOT sectors of diskettes. Make sure that you have the floppy inserted in the drive and try again.

The right to read the BOOT sector may be taken away by another software application such as SUP for instance. Check whether this is the case with the network administrator.

Wrong area selection. It is not possible to test the BOOT sector of the disk X:. Select another area to be tested, or to be tested first.

It is not possible to test the BOOT sectors of all types of storage media. If the program is unable to read the respective BOOT sector, this report will be displayed.

Error in locking the global memory. The allocated global memory may not be locked for the program.

This is a serious error, which may result in collapse of the Windows system.

This is a very serious error of the Windows system, which may cause the computer deadlock and data losses. The report must not be ignored.

The program is unable to use the global memory. An error occurred during initiation of the data necessary to allocate the global memory. This is a serious error which may result in collapse of the Windows system.

This is a really serious error of the Windows operating system, which may cause computer deadlock and data loss. The report must not be ignored.

The program is unable to use the timer. This error occurred after freeing the original timer, so that the program is unable to continue in operation. This error cannot destabilize the Windows system, but the program is ended.

If you request to change the time constants of the programme, the timer reallocation error may occur. The error results in immediate termination of the programme, but no further consequences need to be feared. All memory was freed and you may try to run LGW again.

The virus definition file is not present in the memory. The programme operation requires that the AWANTI.386 library is installed in the memory upon the start of the Windows system.

The AWANTI.386 library contains the core of anti–virus testing of all data of the AVAST! system. If not installed in the memory, it makes no sense, that LGW operated any further.

VxD No. X error. An error occurred in testing the data. The place of its origin is in the critical area. End the program and rerun it.

The error originated in the critical area of the programme. It is a wonder that the system keeps operating. By all means, close the application and try the testing again.

The polymorphic virus testing error. The program is unable to test the NAME file for presence of polymorphic viruses for unknown reasons. The file will not be tested.

It is not very clear, why the respective file cannot be tested for presence of polymorphic viruses. If this report is displayed, contact your dealer or directly ALWIL Software company.

Extended mode detection error. The program is unable to operate in another mode. Terminate your Windows session and use "WIN/3" command.

The LGW program requires an extended mode of operation of the Windows system.

Really? Are you sure that you want to test files on CD–ROM disk? Of course, it is possible but you will certainly admit that it is not a common requirement. The testing procedure may take very long.

The CD–ROM testing procedure may take very long, but it is a true that the virus could exist on the CD–ROM.

Delete the NAME file? If you answer YES, the file will be deleted without the possibility of restoring it. If you are not really sure, do not execute the operation.

If you delete the file by LGW programme, you cannot ever restore it. Even the computer expert would not be able to help since a completely destructive way of deletion is applied.

The NAME file being created already exists. If you answer YES, the existing file will be overwritten without the option of reverting the action.

If you create a file of the same name, it will completely destroy (overwrite) the original file.

Sum–GUARD for Windows

SGW is a program designed for quick test of changes as implemented in the individual files. Its principal purpose is to promptly test the basic (primarily the system ones) files, such as for instance MSDOS.SYS, IO.SYS, COMMAND.COM or WIN.COM. The program is not of course limited to the listed files, but in fact any file can be placed within the program "testing competence", if it is accessible at the moment of the program start–up.

Program Installation

SGW is installed during the AVAST! system installation. The installation itself occurs in two steps. The basic installation programme asks whether you want to install also the programs for Windows. If your answer is affirmative, the files which are necessary for operation under Windows are installed as well.

The actual installation in the Windows system consists in running the AVINST installation programme, which is a standard part of the AVAST! package and is in the ready–state after installation of AVAST! system for DOS.

Using the Programme

The operation principles of the program are very simple. The program reads the files specified on the command line or in the configuration file of the AVAST! system, calculates their check–sums and matches them with the data found earlier or specified on the command line together with the file names.

The program is able to process several various parameters on the command line and one special parameter. The parameters which are unknown to the program, are ignored without displaying any error messages. If the program does not find a file specified on the command line, it is ignored too.

The general command line of SGW looks as follows:

```
SGW [file [sum] [file [sum]...] [/|-][A|a]]
   [[/|-][D|d] [/|-] [M|m]]]
```

where the file represents the full name of the file to checked, sum represents the checksum in decimal form. The checksum needs to be filled in providing you know it. If you do not know it and write the command line for instance in the following form:

```
SGW C:\command.com c:\io.sys c:\msdos.sys
```

the program SGW will calculate the checksum for each file and reports an error, as it did not find any number to which the calculated checksum could be compared. The above–specified command line makes sense only in case you include one of the /A or /D switches in it. If you know a checksum of the individual files, you may specify the same command line in the following form:

```
SGW C:\ command.com 12345 C:\io.sys 23456
```

The meaning of the individual switches is as follows:

/A

adds all files found on the command line into the configuration file. If a checksum is also specified in the command line, it is added too.

/D

deletes the files found on the command line of the configuration file. The specification of the checksum has no influence upon further operation.

/M

switches the program into interactive mode of operation, processing all other parameters of the command line first and after processing, it displays the configuration dialogue.

/A and /D parameters cannot be specified on the command line at the same time. If such thing happens, the program reports error and terminates the programme operation.

If the /A parameter is used, and no checksum for any or all files is specified, these files are added into the configuration file and the checksum is calculated upon the first programme start–up. This fact is not reported in any way, and occurs completely without the user input.

All switches of the command line may be preceded by "/" or "–" characters and may appear both in lower or upper case.

The program accepts one parameter, which is not assigned from the command line but from the features of the icon of the run programme, which determines whether the programme should be run in minimized form or not. If the program is started by selecting the option "Run Minimized", it checks the status of files specified on the command line and specified in the configuration file in this succession. If this option is deselected, the program considers this state the same as if /M parameter was specified on the command line and is run in the interactive mode. Upon installation, two icons of SGW programmes are created. One of these icons, saved in the AVAST! group has this parameter unidentified and running of this programme results in displaying of the configuration dialogue. In the start–up group, this parameter is selected and the programme checks the set parameters and the configuration dialogue is not displayed.

Working Files

In order to operate properly, SGW requires a few files, some of them being the necessary start–up prerequisite (dynamic libraries) and some to read and write (configuration file). If this file does not exist, the program creates it in the directory, from which it was started.

The program uses AVAST!.INI directory to save the necessary data. This file has a standard format of all configuration files of the Windows system and is located in the directory, from which it was started. The program uses the [SGUARD] group for its operation.

FileXXX parameter

The FileXXX parameter is designed to test the names of the tested file and its checksum. The "XXX" set of characters is substituted by a number in the real configuration file. For instance:

```
File1=c:\command.com,12345
```

From the example, it is apparent that the number replacing the “XXX” characters is not preceded by zeroes or blank characters. The checksum is specified immediately after the file name and is separated by a comma.

The numbering of the individual items is not important, it is actually possible to leave out some of them. In using the interactive mode setting, SGW automatically numbers the individual items in the upward manner. If some if the items are deleted, the program automatically uses the vacated numbers.

If you manually edit the content of the control file, it is absolutely necessary to comply exactly with the same parameter format. The program does not anticipate deviations from the hereby described rules. The items, which the program is unable to identify are ignored without displaying any reports.

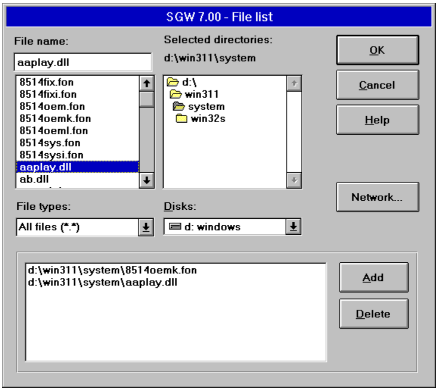
NumberFiles parameter

The NumberFiles parameter is designed for saving of the number of files to be checked. The number is saved as a standard digit in the decimal form.

If you use the interactive mode setting, SGW will automatically maintain the parameter value. If you manually edit the configuration file, it is necessary to bear the following facts in mind.

The program tests only that number of files, which is specified in the NumberFiles parameter, these files being processed in the numerical order. All other files are ignored.

If the NumberFiles parameter specifies a higher number, than as specified in the configuration file, the program reports an error and offers parameter resetting.



SGW 7.00 - File list

File name: aaplay.dll

8514fix.fon
8514fixi.fon
8514oem.fon
8514oemk.fon
8514oeml.fon
8514sys.fon
8514sysi.fon
aaplay.dll
ab.dll

Selected directories: d:\win311\system

d:\
win311
system
win32s

File types: All files (*.*)

Disks: d: windows

d:\win311\system\8514oemk.fon
d:\win311\system\aaplay.dll

Program Control

The SGW program control is the same as in other programs running under the Windows system. The program does not use any special operation methods and does not require any special or unusual knowledge.

Working with the interactive setting dialogue

If the /M parameter or a special parameter are used upon the start–up of the programme, the dialogue for setting the interactive mode to tested programmes is displayed.

The dialogue is a modified version of the standard file opening dialogue of the Windows system. Its modification consists in adding the bottom section, where the list of names of the files, which are tested upon LGW start–up and two options used to work with this list are found.

The unlimited number of files can be added into the list. The only limitation is caused by the Windows system, which does not enable unlimited quantity of elements in the list.

Working with the error messages and reports dialogue

The error messages or reports are generated by the programme as a response to events, which seriously affect the programme operation or are important to inform the user.

This type of dialogue permits only one answer and the program waits until the user inputs this answer. Other running programmes of the Windows system are not affected by this wait and operate in regular manner.

Program Origin

The SGW program in its 7.0 version is the first implementation for the Windows system. It is a part of the AVAST! system 7.00 and above and the numbering of its upgrades is the same as the numbering of AVAST software.

SGW derives its origin from the SGUARD for DOS, which is a part of the AVAST! system, version 3.0 and above. In its work, it uses the same internal algorithms to calculate the checksums of files, so the numbers, which you may come across upon using the programme or handling the working files, are the same, which SGUARD for DOS produces.

The program has the same capabilities as its DOS brother with several changes implemented.

1. The program operates exclusively under Windows system.
2. The program contains interactive setting of the tested files.
3. The program does not contain the option of testing the checksum of the memory, as this ability is useless in the Windows system.
4. In its current version, the programme does not contain the option of testing the system areas of the individual disks.

Operation and Start–Up Requirements

In order to operate, SGW requires the following prerequisites:

1. Microsoft Windows 3.1 or the newer versions of this operating system.
2. The system with 80386 SX and better processors or with a compatible one.

Operating in the standard mode, the programme does not create any window, but despite all this, it finds out whether it is run with a requirement for display in the standard form with an open window, or whether it is to be run in the icon's form.

The program may be run by all methods, which are accepted by all normal 16–bit EXE programmes.

Messages, Warnings, Errors and Queries

SGW is able to display several various types of warnings and error messages, displayed in the various forms. Their list and detailed information of the possible options are specified below.

Wrong checksum. The “ANYFILE” file has a configuration checksum of 99998. The controlling checksum in the configuration file is 99999. Is this change all right?

This warning is displayed if the checksum of the file does not correspond to the data specified on the command line or in the configuration file. The warning appears in two types. In case of the command line file check, the report is displayed as warning. In case of the configuration file check, the report has the form of a query, where you can determine whether the new checksum is alright (change of the file is legal) or not. The report stating the incorrect checksum may be provided by the voice output, if the current installation of Windows supports it. The configuration of the sound report can be set in the control panel using the standard procedure. If the change is all right, the new value of the checksum is saved in the configuration file.

Delete “ANYFILE” file in the configuration file? If you delete this file, it will not be tested anymore.

This warning is displayed in the moment, when some file of the list specified in the configuration file is requested to be deleted. This warning has the form of a query, whether you want to delete this file for sure and with permanent effects. If you answer affirmatively, the file will be deleted out of the directory and will not be tested until it’s set in the configuration file again.

Insufficient memory. I am sorry, but the program is short of memory. The requested service cannot be executed.

This error displays once the program comes across insufficient memory to pursue its work tasks. SGW uses only a minimum amount of memory to operate (around 10 KB), which should always be available. If the program displays this error, please contact your AVAST! dealer, or directly ALWIL Software and inform them of this fact.

File opening error. The program is unable to open ANYFILE file. The file will be removed from the list.

SGW needs to open each of the tested files. The program opens these to read them only and at the same time assigns an attribute which enables reading of this file by another processes, but disables writing. If the tested file is already opened in the exclusive manner or the operation system is unable to access this file for any other reasons, this error report displays. We recommend that you check whether the tested file really exists and whether it is not being used by another running application.

The files to be tested not found. The AVAST!.INI file parameters are not correct. Rewrite the configuration file?

The program displays this type of error as a response to a wrong content of the configuration file. It is the case, when the NumberFiles variable contains a higher number of files to be tested than the configuration file actually contains (FileXXX variable). This error report is displayed in the form of a query, whether you want to rewrite the configuration file. Prior to display of this error, a considerable delay in the programme operation may occur, which is signalled by the change of the cursor to the shape of an hourglass. This delay is caused by intense scanning of the configuration file and searching for the missing files to be tested.

Wrong structure of the command line. The command line must not contain "/A" and "/D" switches at the same time.

This error is displayed as a response of the programme to a wrong structure of the command line.

Alter–Guard for Windows

AGW is designed to test the integrity of the content of all available disks. The program represents the fundamentals of the system protection against virus attack. If you use this program regularly and in an appropriate manner, you have a high probability that you will succeed in saving most of the infected files.

System Installation

The AGW program is installed during the installation of the AVAST! system. The installation itself occurs in two steps. The basic installation program first asks whether you want to instal the programs for Windows as well. If you answer affirmatively, the files which are necessary for work in the Windows session are installed too.

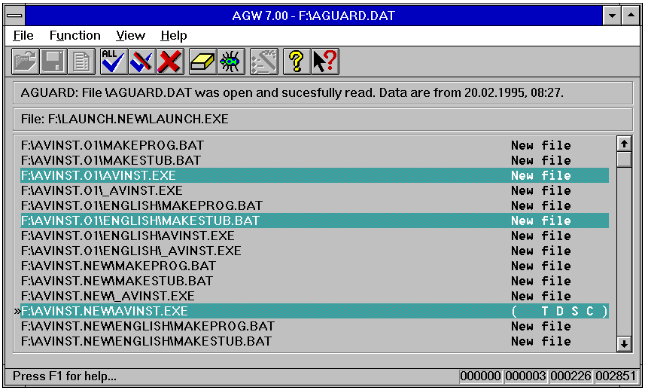
The actual installation in the Windows system consists in running the AVINST installation programme, which is a standard part of the AVAST! package and is ready after AVAST! for DOS is installed.

Using the Programme

AGW is designed to monitor the changes to content of the individual files on disks, which are at a given moment accessible. The program monitors the changes of all parameters, which are characteristic for a given file and is capable of identifying any change executed to their content.

Upon the start–up, the program reads the configuration file and finds whether it does not find the specification of areas to be tested. If this is so, the program automatically starts to test the first of the detected areas. If this is not so, the program awaits input for testing through a slightly modified dialogue for file opening.

For each of the tested areas, FGW opens a file with data already saved. The actual work consists in reading each file in the tested area and comparing the currently discovered data to the data saved upon the last program running. If the data equals, the file was not changed. The user may select several areas to be tested at once, while their number is only limited by the Windows system.



```
AGUARD: File \AGUARD.DAT was open and sucesfully read. Data are from 20.02.1995, 08:27.

File: F:\LAUNCH.NEW\LAUNCH.EXE

 F:\AVINST.01\MAKEPROG.BAT                          New file
 F:\AVINST.01\MAKESTUB.BAT                          New file
 F:\AVINST.01\AVINST.EXE                            New file
 F:\AVINST.01\_AVINST.EXE                           New file
 F:\AVINST.01\ENGLISH\MAKEPROG.BAT                  New file
 F:\AVINST.01\ENGLISH\MAKESTUB.BAT                  New file
 F:\AVINST.01\ENGLISH\AVINST.EXE                    New file
 F:\AVINST.01\ENGLISH\_AVINST.EXE                   New file
 F:\AVINST.NEW\MAKEPROG.BAT                         New file
 F:\AVINST.NEW\MAKESTUB.BAT                         New file
 F:\AVINST.NEW\_AVINST.EXE                          New file
»F:\AVINST.NEW\AVINST.EXE                           (  T D S C )
 F:\AVINST.NEW\ENGLISH\MAKEPROG.BAT                 New file
 F:\AVINST.NEW\ENGLISH\MAKESTUB.BAT                 New file

Press F1 for help...                     000000 000003 000226 002851
```

If any of the parameters is changed, the program writes it out to the main area of its working window together with designation of the type of change. The changed files can be further processed by the program tools, which are detailed in the Program Control chapter.

After termination of testing of some of the areas, the changed data may be for selected (or all) files declared correct and saved in the data file. The data of the unselected files will remain unchanged.

The program operates in the system delays and its design practically excludes any influence upon the working speed of other applications. The only and very small delay that can occur may stem from the access to files and occupation of a large portion of the disk buffer memory, which cannot thus be used by other programs.

Program Parameters

In its version 7.0, AGW does not recognize any parameters represented on the command line. All parameters, which it finds here, are ignored without any warning displayed.

All parameters, which the program needs to save in the period in between two start–ups of the program, are saved in the AVAST!.INI configuration file. AGW uses only a part of the parameters, which are written in this configuration file. This data is found in the [AGUARD] and [AGW–Param] groups.

(AGUARD) group

The AGUARD group contains the basic parameters for the program operation, which do not determine its configuration. These are mainly the parameters necessary for the communication of two instances of the programme.

In the [AGUARD] group, the following parameters can be found:

Temp

The TEMP variable serves for the purposes of communication of the individual programme instances. Its value changes between the individual program start–ups. It is important that its content is not changed during running of the programme by another programme (editor), as important data losses may occur.

[AGW–Param] Group

The [AGW–Param] group contains parameters, which are important for saving of the program configuration between its two start–ups. Manual changes to individual parameters are not recommended due to possibility of their incorrect combination and resulting in data loss.

The following parameters can be found in the [AGW–Param]:

Extension1 – Extension10

These parameters serve to save the tested file types. They save the text strings of 1 – 3 characters in the format, which corresponds to the DOS file type format. In order to keep the compatibility with the AGUARD for DOS version, you may only save ten types of files. If you need to test more types, you must use the characters which are designed to assign more types at one time (“*” or “?”) following the DOS system rules.

TestingAreas

The “TestingAreas” parameter serves to save the areas to be tested, which you want to test regularly and automatically. These areas will be tested upon each program start–up. The value of the parameter is represented by a string of characters.

AutomaticStart

The “AutomaticStart” parameter contains a logic value of 0 or 1, which determines whether the program automatically test the areas to be tested, listed in the “TestingAreas” parameter, without the requirement of the user input. If this parameter has a value of 0, the user is invited to input the areas to be tested. If the parameter has a value of 1 and the “TestingAreas” parameter contains the currently valid area to be tested, the program will automatically start the testing procedure.

FastCheck

The “FastCheck” parameter contains a logic value of 0 or 1, which determines, whether the program will test the contents of the individual files. If the parameter has a value of 0, the file contents will be tested. If the parameter has a value of 0, the file contents will not be tested. In this case, the program serves to perform fast check of the deleted files or attribute changes. If the content of the individual files is not tested, the program still permits the saving of the changed data into the file.

IgnoreArchive

The “IgnoreArchive” parameter contains a logic value of 0 or 1, which determines whether the program will take the changed archive bit of the file into account. If the parameter has a value of 0, the change to the archive bit will be considered a change of the file. If the parameter has a value of 1, the archive bit will not be evaluated. If any of the files change more parameters than just the value of the archive bit, its change will be displayed.

NoSubdirs

The “NoSubdirs” contains the logic of 0 or 1, which determines whether the program will also process the files in the subdirectories of the selected area to be tested. If the parameter has a value of 1, no files in subdirectories will be tested. The value of 0 means that files in the subdirectories will be tested.

ReportMode

The “ReportMode” parameter has a logic value of 0 or 1, which determines whether the program will be able to write the modified data into the data file. If it has a value of 0, the program operates in the standard mode. If it has a value of 1, it will operate in the so–called REPORT mode, in which it is not able to implement any change to the existing data file.

Program Control

The program control does not involve any special procedures or elements which would differ from the control tools of the majority of all standard programmes running under the system. The program can be controlled by the keyboard and mouse, while the procedures and control procedures are the standard ones.

Program Menu

The AGW menu combines all of the programme functions. It can be controlled by the keyboard or mouse, without any difficulties.

The non–accessible menu items are displayed in grey. The accessibility of the individual items are controlled by the current programme status.

File Menu

The File Menu contains the following items:

Select Areas...

The "Select Areas..." item serves to pop up a dialogue, which gives us options of choosing the areas to be tested. A detailed description of the dialogue can be found in the respective chapter. The option can be selected directly by a short–cut key "Ctrl–O".

Save Data

The "Save Data" serves to save the modified and selected data into the AGUARD.DAT file. It is important to realize, that only the selected items will be saved into the data file. The option can be selected by a short–cut key "Ctrl–S".

Delete File

The "Delete File" item serves to delete the selected files on hard disks. It is important to realize that the once the file is deleted it is gone forever. The file restoration may require the presence of an operating system expert. This option may be directly selected by the "Del" short–cut key.

Setting

The "Setting" item serves to pop up a dialogue, in which the individual working parameters of the program can be set. The set parameters are saved in the AVAST!.INI configuration file.

End

The “End” item serves to terminate the program. This option may be directly selected by “Alt–F4” short–cut key.

Function Menu

The Function Menu contains the following items:

Select All

The “Select All” item of the menu serves to select all files, which are at the moment of selecting this function displayed in the main part of the program window. The items, which are not currently visible and are accessible through the shift bars, are selected as well. The selected items are displayed in reverse video (inverse colours). The individual files can be selected by the mouse left button double clicking.

Cancel Selection

The “Cancel Selection” item of the menu serves to cancel selection of all items, which are at the moment of use of this function selected regardless of the selection method. The item selection can be cancelled by double clicking on the mouse left button.

Virus Test

This item serves to test the selected files for presence of viruses. Only the existing files, which have not been so far tested are tested. The test results are displayed visually and are included in the file information, which is accessible by clicking the right mouse button. The infected files are displayed in red.

Create Report

The “Create Report” item serves to create the so–called Report files, which contain information of the current data status of the currently tested area. The files are created in the directory, from which AGW was run. The more detailed information on Report files are described in the respective chapter.

Display Menu

The Display Menu contains items, which select and deselect display of the individual program bars. If the respective item is selected, the corresponding bar is displayed. The program contains the Status Bar and Work Bar, sometimes called “Tool Bar”.

Help Menu

The Help Menu contains the following items:

Index

The “Index” item of the menu enables to select the file with help. A direct selection of the item is possible by using a short–cut key “F1”.

Copyright

The item of the menu called “Copyright” enables to display the “Copyright” dialogue with brief program information.

Tool Bar

The Work or Tool Bar serves to provide fast access to the program functions offered. The individual options (buttons) displayed on the Tool Bar visually display functions, which they represent. Their functions are the same as the function of the corresponding items of the menu. Some options have no direct equivalent among the menu items and serves to simplify the control of the program using the mouse.

ESC Key

The ESC key serves to interrupt a longer–lasting continuous operation of the program, for the completion of which the user for some reason cannot wait any longer. The user may interrupt the testing of the individual area, searching for the deleted files or waiting for the data file saving procedure.

The ESC key or the respective button of the work bar may be used directly to interrupt any of the above–mentioned activities.

Contextual Help

The contextual help serves to provide fast information related to AGW programme. The use of a mouse simplifies in a great deal the selection of the program section, on which you can obtain help. Just click on the respective button of the tool bar and then click on the selected program area. If the help related to a given area of the program exists, it will be immediately displayed.

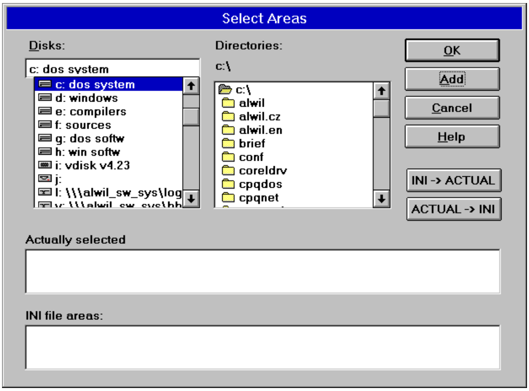
Status Bar

The Status Bar serves to display the program status information and the tested area status information. The whole Status Bar is divided into two main parts. In the left one, information on the items selected by a mouse or the selected buttons of the Tool Bar.

In the right part of the Tool Bar, there are 4 count indicators, which display the number of the infected files, the number of the selected files, number of the processed files and the number of all files.

Program Dialogues

The dialogues of the AGW program represent separate windows, which display or request selection of a certain information. Their designation varies and so does their appearance.



Select Area Dialogue

This dialogue serves to select areas to be tested and is one of the main elements of the programme.

The dialogue represents a modified version of a standard open file dialogue. Unlike the standard dialogue, it excludes the list of files in the selected directory but includes two editing windows, in which the selected areas to be tested and two buttons, which serve to facilitate manipulation with these editing windows are displayed.

Disks

The list display the currently accessible local and remote disks, from which it is possible to select the areas to be tested.

Directories

The list displays the currently accessible directories, which can be selected to be tested. Right above the list, the currently selected directory is displayed.

Just Selected

The "Just Selected" editing window contains a list of areas to be tested after pressing the OK button. An area (directory) can be added into the list with the help of the ADD button or directly by writing in the text.

Values for INI File

The “Values for INI File” editing window contains the list of areas to be automatically tested upon the program start–up if you select the respective switch in the Setting dialogue. The editing window may be directly edited or processed using the buttons designated for this purpose.

OK

The “OK” button informs the program of the fact, that you have selected the areas to be tested and the program may commence its execution. If the “Just Selected” editing window is empty, the directory, which is currently selected is tested and its name is displayed right above the list of currently selected directories.

Add

The “Add” button serves to add the selected directory into the “Just Selected” editing window. After adding the item in the editing window, the program checks whether the area being added is already described by the previous areas or not. If it is, the new area is not added and the user will be informed of this fact.

Cancel

The “Cancel” button informs the program of the fact, that the user changed his mind and does not want to continue in the work performed so far. The button deletes the window and all modifications in the editing window will be disregarded.

Help

The "Help" button serves to display the help instructions related to AGW. Once the help is called up, this page will be displayed.

INI–>Actual

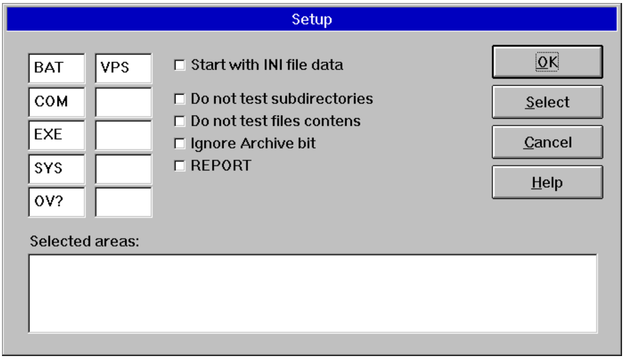
The "INI–>Actual" button serves to copy the content of the editing window "Values for INI file" into the "Just Selected" editing window.

Actual–>INI

The "Actual–>INI" button serves to copy the content of the "Just Selected" editing window into the "Values for INI file" editing window.

Setting Dialogue

The "Setting" dialogue serves to set the program parameters, which may modify its operation. The set parameters are saved in the AVAST!.INI configuration file.



File Types

The AGW program enables testing of up to ten types of various files. If you want to test more types, you may use special characters (wildchars), for example the "*" type means that the program will test all files. The file type can be up to 3 characters long. If you write more characters, the program will advise you of such a fact.

Selected Areas

The “Selected Areas” editing window contains the list of directories, which are automatically tested upon the program start–up, in case you have selected the “Data for testing from INI”. The content of the editing window is displayed in the dialogue for selecting the areas to be tested in the “Values for INI file” editing window. The editing window may be directly edited or the individual areas may be designated for testing using the Select Areas dialogue.

Data for Testing from INI

The switch determines, whether the AGW program upon started up automatically test the areas saved in the AVAST!.INI configuration file or whether the program will wait until the tested areas are selected manually.

Do not Test Subdirectories

The switch determines, whether the AGW program will also test the subdirectories of the selected directories. If you want to test the whole disk completely, you have to leave this switch selected and select the ROOT directory as the area to be tested.

Do not Check the File Content

This switch determines whether the program will check for changes of file content. In testing the individual files, the tested elements are: change of time and date of creation, change of the length and attributes. The file content may and may not be tested. If it is not tested, the program operation is much faster, but you will not obtain complete information. At the same time, if you do not test the file content, you will not have the option of restoring the original file content.

Ignore Archive Bit

The switch determines whether the program will ignore the change of the archive bit in the attributes of files or not. If the switch is selected, the change of content of the archive bit will be ignored. If several parameters of a file are changed, the change of the archive bit is displayed together with other changes.

Report Mode

The switch determines whether the program will be able to save the test results into the data file. If selected, the program will not be able to modify the data file and the results can be only viewed on the screen.

OK

This button informs the program of the fact, that the set parameters are correct and are to be used for further operation of the program. The dialogue window disappears from the screen and the parameters are saved in the configuration file.

Select Areas

This button will pop up the Select Areas dialogue and after having done so, places the selected areas in the editing window.

Cancel

This button informs the program of the fact, that the parameters are incorrectly set and are not be used any further. The dialogue window disappears from the screen and the parameter changes are not saved.

Help

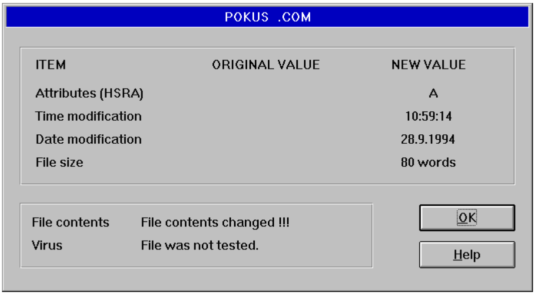
This button displays the help file containing information of the AGW program displaying this page.

Copyright Dialogue

The Copyright Dialogue serves to display very brief information on the program version and copyright, which pertain to the program.

File Status Dialogue

The File Status Dialogue is designed to display the description of changes to files, which are displayed in the main part of the working window. The dialogue displays an easy–to–follow table listing the respective changes, which occurred over the time. If the file was deleted or created, the respective column in the table is empty.



ITEM	ORIGINAL VALUE	NEW VALUE
Attributes (HSRA)		A
Time modification		10:59:14
Date modification		28.9.1994
File size		80 words

File contents	File contents changed !!!
Virus	File was not tested.

Message, Warning or Error Dialogue

The Message, Warning or Error Dialogue is designed to provide a simple display of the respective information. At the window heading, there is a text, which describes the program and type of message, which is displayed. Also, the text is displayed in various colours to assign importance to the various messages.

The normal message is written in a standard black, warning is light blue, error is purple and a very serious error is light red. The dialogue of this type offers only one response option, which is the affirmative “OK”, by which the user acknowledges having read the information and being aware of it.

The program is interrupted for the time of the dialogue display. Other programmes of the Windows system are not af–

fected. The dialogue of this type is used by all AVAST! programmes.

Query Dialogue

The Query Dialogue is designed to display a query and ask the user for an answer. At the window heading, there is a text, which specifies the program and type of message displayed, which is always in this case a query. Also, the text is displayed in various colours to assign importance to the various messages.

The normal message is written in a standard black, warning is light blue, error is purple and a very serious error is light red. The dialogue of this type offers only two response options – the affirmative "YES" selected by pressing the respective button and negative "NO".

The program is interrupted for the time of the dialogue display. Other programmes of the Windows system are not affected. The dialogue of this type is used by all AVAST! programmes.

Mouse Buttons

AGW makes use of the mouse similarly as most other programmes running under Windows do. However, some tools and procedures were implemented to simplify the program control thus accelerating the work.

Left Button

The left button has a special meaning only for the main part of the working window. A simple click of the mouse can be used to set the keyboard cursor on the item at which the mouse points. A double click of the left button will select the given item or deselect it.

Middle Button

The middle button is not always a part of all mouse types and its simulation is not standardized, so the operation of this button is not implemented in the AGW program.

Right Button

The right button of the mouse is used to display the file status dialogue. The dialogue is displayed immediately once the right button is pressed once (click). The right button is active only if the mouse cursor points at some line containing the file change description.

Main Part of the Window

The main part of the window is an area, in which the modified files are displayed. The number of these files may be higher than the window can accommodate at one time. In such a case, a shift bar is displayed at the right window edge, by which the list of files can be moved up or down.

Each of the displayed items consists of several parts. On the left side, there is a file name with full path. On the right hand side, the file status as it was modified from the last program start–up is displayed.

The main part of the window can be operated using a mouse or keyboard. The description of the mouse operation is dealt with in the respective chapter. If you prefer keyboard, you can use it too. For this purpose, a keyboard cursor is implemented, being displayed as "»" right next to the file names. The cursor can be operated using the standard keys.

Report Files

The report files can be created using this option. The AGW program as well as AGUARD for DOS create two files, with the name AGW–[disk name].ALL and AGW–[disk name].DIF. The files are created in the directory, from which the program was run and their content consists only of the text in the 1250 code page either in Czech or Czech without diacritic marks depending on the program operation mode.

The format of files is always the same and in future will not probably be modified, so the output may be processed by another program.

The program is interrupted for the time of the report files creation. Other programmes of the Windows system are not interrupted too, mainly due to this system design.

Program Origins

The version 7.00 of the AGW program is the first implementation for the Windows system. It is a part of the AVAST! system version 7.00 and above and its numbering is the same as the one of AVAST! system versions.

The AGW program derives its origins from the AGUARD for DOS program, which is a part of the AVAST! package version 1.0 and above. In its operation, it uses the same internal algorithms to detect file changes, so the results that you may see when you use this program are the same as those produced by AGUARD for DOS.

The program has the same abilities as its DOS version except for several updates.

1. The program exclusively operates under Windows system.
2. The program enables interactive setting of the tested areas.
3. The program does not enable restoring of the modified files.

Operation Requirements

In order to operate, AGW needs the following:

1. Microsoft Windows 3.1 or a newer, 16–bit operating system.
2. A computer with at least 80386 SX processor or a fully compatible one.
3. The AWANTI.386 library needs to be correctly installed in order to test files.

Messages, Warnings, Errors and Queries

AGW is able to display several types of warnings and error messages in various forms. Their detailed list and information on the options are specified below.

Repeated area test. The “ANYAREA” area is already covered and it does not have to be added. The repeated testing of the same data only slows down the processor and has no further meaning. The item will not be added.

> In selecting another area to be tested, AGW discovered that this area was already placed among those to be tested. Such a setting is useless and the new area thus will not be added.

The selected disk is a CD–ROM disk. The program cannot be used to test this medium. The requirement was cancelled.

AGW needs to keep the data file in the ROOT directory of the tested disk. The writing to the medium in the CD–ROM drive unit is not possible and therefore AGW refuses to work with this disk.

The configuration file is not saved. The program detected an error in saving of the modified parameters to a disk. The changes are not saved and are not accepted.

The program detected an error in saving of the configuration file to a disk. The error is not caused by the programme, but by the operating system or a disk error. By all means, the changed data are not saved to the disk.

The selected directory is too large. The area of the memory, which maintains the list of selected directories to be tested is not unlimited. The selected directory already exceeds the given area. The request cancelled.

AGW has only limited possibilities as to the saving of the working data in the memory. These possibilities are not limited by the 16–bit structure of the Windows system (memory segmentation), but despite all of this, the selected area to be tested exceeds the program capacity.

Disk saving error. The program is unable to read the system area of the disk X. The system are is not tested.

AGW was unable to read the system area of a disk. This fact is most likely caused by another program, which prohibits access to these areas.

Testing exception. The disk X is operated by the STACKER. Its system area is constantly modified. The system area of the disk is not tested.

In its operation, the STACKER program keeps writing its internal data in the BOOT sector of the disk, which it operates. AGW detects these changes and report them. As these changes are not suspicious and continuous, the program detects the presence of STACKER on the tested disk and if it finds such a program, it does not test the BOOT sector.

Delete the selected files? If you answer YES, all of the selected files will be deleted with irreversible effect from the hard disk. Are you sure you want to proceed?

AGW enables the deletion of the selected files. The deletion is, however, an irreversible function and that is why AGW writes this message.

Insufficient memory. The program does not have sufficient space, most likely in the local memory. The requested operation cannot be executed and is cancelled.

AGW is limited by the Windows system and the size of the local memory is insufficient. The operation selected by the user exceeds the system capabilities.

Insufficient global memory. The program has not enough space in the global memory. Please end some application and try again. The requested operation is cancelled.

AGW uses the main memory of the Windows system to perform operations that demand large memory. If it is insufficient, the operation will not be executed.

Tested file access error. The program detected an error in handling some file. The area testing is incomplete!

For some reason, AGW cannot read a part of the tested file. This error was caused by the Windows system or a disk error.

Incorrect version of the opened file. The program is unable to process the information saved in the file due to incompatible versions. Delete the AGUARD.DAT file and try again.

AGW writes to the data file the program version number which created this data file. As the time goes on, the structure of the file changes and the program is able to decide whether the data file is compatible with the file or not.

Wrong content of the opened file. The program is unable to process the information saved in the file due to its unauthorized change. Please restore the AGUARD.DAT file from the backup copy and try again.

AGW writes into the data file information, which later serves to check the file consistence. If another program changes the content of this file, AGW cannot use it.

File opening error. Unable to open ANYFILE file. The file will not be tested.

AGW is unable to open the file to be tested. The file exists but another program prevents its opening by opening the file for its own use or by otherwise preventing access to it.

File search error. In searching the file, a general error occurred. The program is unable to continue to test this area.

AGW discovered a general error in searching the file. This error is caused by the Windows system or by a hard disk error.

Memory access error. The program is unable to access the formerly allocated memory. The requested operation cannot be executed.

AGW is unable to access the formerly allocated memory. This memory should be accessible but for some unknown reason is not. This error originated in the AGW program or in the Windows system.

Database restoring error. This error originated during internal processing of changes, which you wished to save on the disk. If it occurs more frequently, call ALWIL Software and ask for assistance.

During the saving of the data on the disk, it is necessary to modify the data format so that it is compatible with the AGUARD for DOS program. Such a format modification should occur without any troubles. If this is not so, please contact directly ALWIL Software and consult the problem with the person in charge.

The changed data was not saved to the disk. An error occurred during saving of the data and the data could not be saved. The original file remained unchanged.

AGW detected an error in saving of the data file to the disk. The error originated in the Windows system or on the hard disk. The most frequent cause for this may be insufficient space on the target disk.

Unable to delete the requested files. Your request cannot be executed due to a memory access error.

When identifying the file deletion request, AGW detected an error in access to the database located in the memory. This error originated in the AGW program.

Antivirus test error. The ANYFILE file will not be tested for presence of viruses due to insufficient memory or file access error.

AGW is unable to test the selected file for presence of viruses. In order to enable this procedure, it is necessary to read it into the main memory. If this memory is not available or the file is not accessible, the testing cannot be executed.

REPORT file opening error. The program detected an error upon working with the ANYFILE file. The Report files are not complete.

AGW detected an error in creating the REPORT file. This error originated in the Windows operating system or on a disk. The most likely cause of its origin is insufficient memory on the disk.

The system area content change. The system area of the disk X was changed. Is this change all right?

This query is displayed as a result of the changed content of the system area of the tested disk. Its change, if not justified, almost always means that the system was attacked by a virus or implies another serious damage to the computer operation.

Save the changes to a disk? After confirming this command, the selected files will be declared correct and the changed database will be saved to a disk. If you answer NO, the program asks about the option of deleting the saved data.

AGW saves into the data file only those changes, which the user selects.

Cancel the read data? If you answer YES, the detected data will be deleted and the database on the disk will

remain unchanged. If you answer NO, the program will return to processing of the changed files list.

The process of reading the data about the tested area may a time–consuming task. AGW warns you before you cancel the hard results of your work.

Close application in operation? The request to end the program in the middle of work is not very common. If you are troubled by the program window, reduce it to a size of an icon.

Ending the program in the middle of work is not a very common request.

Close the application without saving of the data? Your request is not common. If you really want to end the program without saving the data, select the YES button.

Ending the program in the middle of work is not a very common request.

This page is intentionally left blank.

Viruses and the Computer Network

Computer viruses and computer networks require a separate chapter. This is not because networks are special targets of viruses or greater damage is done here, but because in a system with many users damage can spread more quickly, and elimination is often more complicated than with independent computers. Also the risk of reinfection is much greater.

Computer network protection depends on checking the identity of the users (defined for example by name and password) and assumes that each user is given a certain degree of access. One of the problems faced by a computer network, of course, is that while a user may be identified, the same may not be true of all the programs he or she runs. Such programs will nevertheless enjoy the same degree of access to the network as the user. For example, if a user with minimal access installs a program in the one directory to which he has access and runs it, that program has no access to other areas of the network. But if the same program is run by a user with Supervisor status, the same program gains unlimited access.

In practice it is necessary to make use of the types of protection offered by the network. What are these? Network files, like DOS files, for example may be given certain attributes. The difference is that there are far more network attributes and they cannot be so easily changed. Moreover, there is the "execute only" attribute (absent in DOS), which permits the protected programs to be run but not read, copied or modified. Similar attributes may be specified for user access to directories or files. For some programs, however, this idea is not feasible. A number of programs for example cannot be write–protected because they write their own configuration to themselves. In any case the "execute only" attribute should be used with caution because it can cut both ways: it is true that programs thus protected cannot be infected by viruses; but nei–

ther can they be checked by antivirus programs. And should they be infected before installation, the virus would be difficult to trace. There are also problems with certain programs: if a program contains overlay, which reads itself, it will not function. And there are such programs. There are other factors that emerge in this connection, such as the creation of a variable path. This determines the directory in which programs to be run are sought. Under no circumstances should a network contain a "public" directory in which anyone may save anything. On other hand, we recommend that the network manager set up a user with supervisor's rights without the right of writing. This user can be used by the manager for testing without endangering further files with infection. In any case, the supervisor must devote appropriate attention to this sort of problem.

How do viruses spread through a network? Experiments have been done to determine how various viruses are capable of spreading in networks and to what extent networks are resistant to them. It has been shown that this often depends on the moment the viruses become active – in Novell, for example, before, after or between the IPX, NETx and LOGIN programs.

In general viruses may be divided into four groups according to their importance for computer networks:

- viruses which operate normally
- viruses which do not operate in networks
- viruses which are in conflict with the network
- viruses created especially for networks

Most viruses belong to the first group, and against them the correct configuration of network protection is of utmost importance.

The second group certainly includes all classical boot viruses. These are able to attack individual network terminals and even the server under certain circumstances, but they are not capable of spreading throughout the network from one computer to another simply because the network does not offer such low–level services as BIOS. Other viruses closely connected with DOS also belong in this group, since they are unable to work with remote devices. These include the "tunnelling" viruses, which seek an entry point deep in the operating system in order to bypass whatever antivirus protection is being used. But in doing so they will often bypass the network software as well.

The third group includes viruses which can only be active while the network is inactive. The most common case is a virus which calls up an idle function of the operating system for its recognition in the memory—a function which turns out to be in use by the network at the moment.

The fourth group includes viruses which are able to make use of the possibilities offered by networks. Such viruses are far more difficult to create than DOS viruses, and so far this group consists of just two virus strains. The first is the GP1 virus, which monitors functions for LOGIN and then tries to move to a determined area by using the name and password. The second is the Yankee.Login virus, first detected and analysed in Czechoslovakia, which does something similar but using slightly different means. From the file which it has infected, it determines whether the LOGIN.EXE program is being run and if so, absorbs the coded name and password.

No virus known so far is able to make more sophisticated use of network operation modes. Whether such a virus might exist and how it might bypass network protection systems has been a subject of controversy for more than two years. The Novell firm insists that such a virus is not possible in practice. Certainly it has not yet been discovered. But in this field there can never be one hundred percent certainty.

AVAST! and Network Support

AVAST! is of course able to work with computer networks. It has been developed and tested especially for the Novell network, but in practice it works with a number of other types of networks. All the AVAST! programs work with networks: the Novell network automatically recognizes the memory resident FGUARD and RGUARD locating programs which are capable of reporting detection of known viruses to selected users.

It is important that AVAST! be used in consultation with our firm. The number of work stations for example must be determined before the package can function. Our firm offers dis–

counts for multiple use of programs, and there will certainly be a type that suits your needs.

It is important that the network supervisor correctly assess his needs and then adhere to the virus protection program worked out in advance. It is especially important to insure that all users are trained to use the antivirus programs correctly and that they always have the latest versions at their disposal.

The CHKAVAST Program

The first problem can be solved with a very simple program: CHKAVAST is able to test whether FGUARD or RGUARD are installed in the memory and to display the result with an exit code. The program can function in any circumstance, but it is especially useful to run it while registering network users. Users without one or both of the programs installed in memory can be excluded from the network.

CHKAVAST parameters

The CHKAVAST program has just two parameters to determine which programs will be checked:

F

This parameter determines that **the presence of FGUARD** in the workstation memory will be checked.

R

This parameter determines that **the presence of RGUARD** in the workstation memory will be checked

If both options are selected, or if both are deselected, the presence of both memory resident programs will be checked.

```
CHKAVAST ISTIME hh:mm
```

Serves to determine the system time; The return code is set to 1, if the system time is ahead the given time and to 0 if the time is behind or equal.

CHKAVAST ISDAY [NEW] [OLD] [<day>] [<date>]

Serves to determine the system day and date settings and to branch commands in batch files if the condition match. The return code is set to 0 if the condition matches and to 1 if it does not match. The NEW parameter determines whether the computer has been booted for the first time this day, and OLD has the opposite meaning.

In addition to these parameters, the days of week (MON, TUE, etc.) and dates (a pair of two–figure numbers denoting day and month) may be selected. If either of these numbers equals zero, they denote any value, so that for example 00–08 means any day in August and 15–00 means the 15th day of any month. The form of entering the date must conform to the current country code setting (European or US) Parameters could be combined; the final result will be correct only if all the conditions for each match.

Example:

```
chkavast isday mon new
if errorlevel 1 goto label1
echo At beginning of week do all the tests!!
lguard c:\*.* /s /e*
:label1
... isday 13-00 fri
if errorlevel 1 goto label2
pause Today is Friday the 13th. Good luck!
:label2
```

CHKAVAST ISMONTH [NEW] [OLD]

Serves to determine whether the computer has been booted for the first time in the month. The return code is set to 0 if the condition matches and to 1 if it does not match. The NEW parameter determines whether the computer has been booted for the first time in the month, and OLD has the opposite meaning.

Example:

```
chkavast ismonth new
```

```
if errorlevel 1 goto label1
echo Let's make month backup!!
:label1 ...
```

CHKAVAST ISWEEK [NEW] [OLD]

Serves to determine whether the computer has been booted for the first time in the week. The return code is set to 0 if the condition matches and to 1 if it does not match. The NEW parameter determines whether the computer has been booted for the first time in the week, and OLD has the opposite meaning.

Example (the same as for ISDAY but better, because it works fine even if the computer was not turned on on Monday).

```
chkavast isweek new
if errorlevel 1 goto label1
echo At the beginning of week do all the tests!!
  lguard c:\*.* /s /e*
:label1 ...
```

CHKAVAST ISYEAR [NEW] [OLD]

Serves to determine whether the computer has been booted for the first time in the year. The return code is set to 0 if the condition matches and to 1 if it does not match. The NEW parameter determines whether the computer has been booted for the first time in the year, and OLD has the opposite meaning.

Example:

```
isyear new
if errorlevel 1 goto label1
echo Happy New Year!!
:label1 ...
```

CHKAVAST uses the file AVAST.DAT which is located in the same directory as the program itself. It uses this file to determine the correct answer for NEW and OLD commands. Parameters ISDAY, ISWEEK, ISMONTH and ISYEAR are quite independent; for example on Monday morning both ISDAY NEW and ISWEEK NEW are true.

Exit Codes

The CHKAVAST programs sets an exit code to the operating system. This code may later be tested either by another (parent) code or in the command batch with IF ERRORLEVEL, or in the LOGIN Script network. The CHKAVAST exit codes may assume only the following values:

0 The program(s) is (are) installed in the work station memory.

1 The program(s) is (are) not installed in the work station memory.

Use of CHKAVAST in LOGIN Script

To test for the presence of the memory resident programs, first create the special command batch TSTAVAST.BAT, which will have for example the following content:

```
@echo off
rem batch TSTAVAST.BAT called up from Login Script
chkavast
if errorlevel 1 goto end
echo RGUARD and FGUARD not installed.
echo You may not be connected!
logout
:end
```

The following line must be at the end of the user's Login Script:

```
#TSTAVAST
```

CHKAVAST.COM and TSTAVAST.BAT files must be placed in the LOGIN directory. If the LOGOUT.EXE program is not placed on the path, its location must be specified.

Updating AVAST! Programs in the Network

It is very important to assure the distribution of updated versions of antivirus programs. Larger organizations usually have plans for such distribution to each user. In a computer network this may be a relatively simple procedure, even if they should reach not only the server but all work stations (for example for tests before registration with the network). Such distribution can be organized as follows:

- The server should set up a public directory, for example \AVASTNEW\.
- New versions of the AVAST! package (or its appropriate parts) may be placed here (for example a new VPS file).
- Then after the registration of a user the following command may be given from the work station:

```
replace X:\AVASTNEW\*.* C:\AVAST! /U /R
```

"Replace" is a standard MS–DOS component. It is a program which copies files requested in the first parameter only if they are newer than the files corresponding to the second parameter. The letter X corresponds to the letter assigned by the network to the disk.

This simple procedure insures that the program facilities are updated at all work stations at the moment a user is registered. Details of the Replace system program can be found in the MS–DOS documentation.

Description of Some Computer Viruses

The next paragraphs will describe some of the viruses. Updates to this user's manual may be contained in the READ.ME file on the distribution diskette.

Virus 534 (W–13)

Virus 534 is a very simple virus which first appeared in 1990. It begins to spread the moment an infected program is started. Infected programs are lengthened by 534 bytes. The virus infects COM files in the current directory, and if no such files are found, it then looks in the root directory of the current drive.

No destructive effects have so far been attributed to this virus.

Its virus flag is a change in the date of the last modification of the file, which always appears as the 13th month.

Virus 648

Virus 648 (Vienna, PC–Boot), perhaps the earliest virus reported in Czechoslovakia, first appeared in 1988 and has been among the most widespread. It begins to spread the moment an infected program is started. The virus infects COM files, increasing their length by 648 bytes. It uses the PATH system variable, so that it spreads quickly throughout the system, and thanks to its knowledge of PATH it attacks the most frequently run programs.

It completely destroys one out of eight programs that it finds: at the beginning of the program it writes system boot instructions, and the original content of this part of the program is destroyed. The length of the destroyed program does not increase.

Its virus flag is a change in the time of the last file modification, which appears as 62 seconds (this value is not listed in the DIR command).

The main symptom of the virus is that when some programs are run, the system is rebooted, or else there is "computer deadlock".

Virus 744

Virus 744 appeared in 1990. It is a modification of virus 648. It begins to spread the moment an infected program is run. It attacks COM files and increases their length by 744 bytes. It uses the PATH system variable so that it spreads quickly throughout the system, attacking the most frequently run programs.

It destroys one out of eight of the programs that it finds: at the beginning it writes a nonsensical instruction and destroys the original beginning. The length of the destroyed program is not increased.

Its virus flag is a change in the time of the last file modification, appearing as 30 seconds. When some programs are run, computer deadlock follows.

Virus 897 (April 1st)

Virus 897 (April 1st, SURIV 1.01) appeared in 1990. It is a memory resident virus attacking COM files along with the COMMAND.COM system program. The virus begins to spread the moment an infected program is run. It spreads by opening, on the same disk where the infected file is found, a working file called TMP$$TMP.COM, in which it writes itself and the infected program.

Its effects appear in the message "APRIL 1st HA HA HA YOU HAVE A VIRUS" which is displayed on April 1st and followed by deadlock. From 2 April to 31 December the message "YOU HAVE A VIRUS" is displayed when a program becomes infected.

The virus flag of 897 is the "SURIV" string ("virus" backward) appearing in infected COM files. Symptoms of the vi–

rus's presence are the messages displayed and computer deadlock.

Virus 1339 (Vacsina)

This virus first appeared in Czechoslovakia in 1989. It is a memory resident virus whose spread does not begin the moment an infected program is run. It installs itself in the memory, from which it monitors the running of programs. It attacks COM files, which it lengthens by 1339 bytes. It modifies some EXE files so that they are changed to the COM type (though they have the extension EXE). Their length is increased by 132 bytes, and when they are next run they may become infected. 1339 is one of a group of viruses which are able to work together and sometimes modify each other.

Virus 1339 has shown no destructive effects. Its symptom is a certain code in the operating memory and at the end of an infected program.

Virus 1560 (Alabama)

Virus 1560 (Alabama) appeared in the fall of 1990. It is a memory resident virus infecting EXE files and the DEBUG and SYMDEB programs. It spreads when programs are run or files opened. It does not attack the program or file that is being run, but other EXE files in the current address. Its peculiarity is that it can survive in the memory when the computer is reset using Ctrl+Alt+Del.

After 674 generations of the virus a message appears on the screen: "SOFTWARE COPIES ARE PROHIBITED BY INTERNATIONAL LAW" and "Box 1055 Tuscambia ALABAMA USA". The computer then shuts down.

Its virus flag, like that of the 648 virus, is that the time of the last modification is set to 62 seconds.

Virus 1618 (Mixer 1A)

Virus 1618 (Mixer 1A) appeared in 1990. It is a memory resident viruses attacking EXE files longer than 8192 bytes. It spreads upon execution of programs.

The virus re–codes characters sent to the serial or parallel ports. After a certain number of generations, and when 50 minutes have elapsed since its installation, keyboard reset is disabled; ten minutes later a bouncing ball appears on the screen, as with the Ping–Pong virus.

The virus flag is the “MIX1” string at the end of infected programs. Its principal effect is to cause problems with the printer, including garbled printouts.

Virus 1701 (Cascade)

Virus 1701 appeared in 1989 and was very widespread. It attacks COM files, lengthening them by 1701 bytes. It spreads very quickly. When the system is turned on the virus installs itself in the memory, where it monitors programs executed. If a COM program is executed, the virus checks whether it has already been infected and if not, it attacks it. Thus its spread is not connected with the execution of an infected program.

Its effects are relatively harmless. In 1988 it caused characters to fall downwards on the screen. Their number progressively increased, so that work became impossible. If the date of the system does not fall between 1 October and 31 December 1988 the virus has no effect.

The 1701 virus flag is its code length. The virus determines whether the program begins with a jump instruction and whether the jump is a certain length from the end (a jump at the start of the virus).

In systems outside the period mentioned above, the presence of the virus is difficult to detect. The virus destroys no data, and programs continue to work correctly.

Virus 1800 (Dark Avenger)

Virus 1800 (Dark Avenger, Bulgarian Virus, Sofia Virus) appeared in 1989. It is a highly dangerous memory resident program which attacks COM and EXE files. It spreads rapidly because it attacks programs not only the moment they are run, as do most viruses, but also during other operations (creating or closing files, determining attributes, renaming, etc.). Infected programs contain the texts: "–Eddie lives...somewhere in time!" and "This program was written in the city of Sofia (C) 1988–89 Dark Avenger". The virus modifies the disk boot sector.

The destructive effect of the virus is that after each sixteenth program run from the given disk, an incidental cluster is written in its code and the original content is destroyed. The lost data is exceedingly difficult to reconstruct.

The virus does not have its own flag. Its presence is detected from its complete code in the files tested.

Virus 1813 (Friday the 13th)

Virus 1813 is perhaps the most widespread of all. It first appeared in Czechoslovakia in the fall of 1989. It is a memory resident virus which attacks COM and EXE files. Because of an error that it contains, it attacks EXE files repeatedly. It has been dubbed the "political" virus because it is said to have appeared first in Israel in May 1988 on the eve of that country's 40th anniversary. The virus does not attack the COMMAND.COM system program.

About thirty minutes after its installation in the memory the virus opens a window on the screen and begins to slow the operation of the computer. Each Friday the 13th it erases all files retrieved.

Its flag is the string "MsDos" in infected COM files.

Except on Friday the 13th the virus's only effect is to slow the operation of the computer.

Virus 2881 (Yankee Doodle)

Virus 2881 (Yankee Doodle) first appeared in Czechoslovakia in the autumn of 1989. It is a dangerous memory resident virus with a large number of mechanisms for disguise and self–defence. For example it is capable of correcting its own code and, under certain circumstances, exiting programs that it has infected. It belongs to the same group as Virus 1339. It attacks COM and EXE virus the moment they are run.

It modifies the Ping–Pong virus whenever it finds it, and under certain conditions it plays "Yankee Doodle" at 5:00 p.m.

Its flag is a certain code at the end of the infected program.

Virus 2928 (Yankee Doodle)

This is an earlier version of the preceding virus. It is 47 bytes longer and only differs in that it always plays "Yankee Doodle" at 5 p.m.

Ping–Pong Virus

The Ping–Pong virus appeared in July 1989. It spreads very quickly. It does not attack files but rather the disk boot sector. When infecting a disk it overwrites the boot sector with its own code. Moreover when it finds a free cluster on the disk it marks it as flawed, then writes the second part of its own code onto it along with the original content of the boot sector. When the system is booted from an infected disk the virus installs itself in the memory (decreasing it by 2 KB) and monitors access to disks. When uninfected disks are read, they become infected as described above. The virus spreads on the hard disk during booting from infected diskettes, which can happen by mistake if they are left in drive A: and Ctrl+Alt+Del are pressed. If this happens, the diskette should be removed immediately and the system rebooted using Ctrl+Alt+Del. In this way the hard disk will not be infected.

Under certain circumstances a "ball" (a character with the code 07) begins to bounce across the screen.

The virus can be modified by Viruses 2881 and 2928 so that after the 255th system boot is ceases to be functional.

Stoned Virus

This is a boot virus which does not infect files but the boot sector on diskettes and the disk partition table (DPT) on hard disks. Its presence is indicated by a decrease of 2 KB in the operating memory (which can be determined for example by PCTOOLS). In infected areas (i.e. the boot sector on diskettes and the DPT on hard disks), the following texts appear: "Your PC is now Stoned" and "LEGALISE MARIJUANA". The first message is usually displayed on the screen when the system is booted from an infected diskette. The virus overwrites one area of the infected medium (track 0, head 0, sector 7 on the hard disk, and track 0, head 1, sector 3 on diskettes). Since the original content is overwritten, the result might be damage or data loss.

In the memory, the Stoned virus infects only diskettes in drive A:. Each operation with unprotected diskettes may lead to the spread of the virus! It can reach the hard disk whenever the system is booted from an infected diskette, which need not be a system diskette. The Stoned Virus does not respect the BPB format (BIOS Parameter Block) in the diskette boot sector, so that data in this block is garbled, which may lead to damage and data loss.

Virus 2967 (Yankee Doodle)

This is a modification of the Yankee Doodle Virus 2881. It is a memory resident virus which attacks COM and EXE files when they are run. It lacks most of 2881's mechanisms for disguise and self defence. But it is able to monitor the LOGIN.EXE program, a component of the Novell network, and gather the coded names and passwords of network users. The virus appeared in Czechoslovakia in the spring of 1991.

Virus 1575 (Caterpillar)

Virus 1575 is a memory resident virus infecting COM and EXE programs at the moment of their retrieval (for example with the DIR and COPY commands). Two months after infection, the virus appears on the screen in the form of a caterpillar which moves from left to right and top to bottom, moving characters on the screen.

Bloody! Virus

The Bloody! Virus is a memory resident virus infecting disk system areas: the boot sector on diskettes and the disk partition table on hard disks. At the 128th system boot from an infected hard disk, the message "Bloody! Jun.4, 1989" is displayed (the date of the Tienanmen Square massacre). Loss of data can occur from older IDE hard disks!

Michelangelo Virus

This virus first appeared in Czechoslovakia in September 1991 as an unknown strain. It attacks the system area of disks: the boot sector of diskettes and the disk partition table of hard disks. It was discovered to have been derived from the Stoned Virus and is in no way remarkable except in its effects: it can be very dangerous. At each start–up it tests the date in the computer and on 6 March it overwrites the content of the disk from which the computer was started. The virus reads the date directly from the computer clock (DOS is not yet active on start–up, and the date cannot be detected in the operating system). Therefore disks in computers without battery–powered internal clocks (such as the IBM PC/XT) cannot be overwritten.

Stoned (2) Virus

Stoned (2) (NoInt, Arc Hub) is a boot virus which does not infect files but disk system areas, especially the boot sector on diskettes and the DPT on hard disks. It is derived from the Stoned Virus and can be recognized since it reduces the oper–

ating memory by 2 KB. It overwrites one area of the infected medium (track 0, head 0, sector 7 on hard disks, and track 0, head 1, sector 3 on diskettes). The original content of these areas is overwritten, sometimes resulting in modification or loss of data. The virus contains the string “ARC HUB 8A”.

The Stoned (2) Virus employs the “Stealth” technique: instead of reading or writing the infected sector, the original sector is read or written. Thus, when the virus is active it cannot be detected by some antivirus programs.

In the memory the Stoned (2) Virus only infects diskettes in the A: drive. Any operation with an unprotected diskette thus contributes to the spread of the virus! The virus can infect the hard disk whenever the computer is booted from such a diskette, which need not be a system disk. The Stoned virus does not recognize the BPB (Bios Parameter Block) in the boot sector of diskettes. Therefore the data in this block is garbled, which may result in damage or data loss.

Virus 1376 (Helloween)

This virus appeared in Czechoslovakia at the beginning of 1992 and was obviously created here. It is memory resident, infects COM and EXE files, but does not infect certain antivirus programs, including those of Czechoslovak origin. It tests the date in the computer and on 1 November displays the following message:

```
Nesedte porad u pocitace a zkuste jednou delat neco rozumneho!
          ********************
 !! Poslouchejte HELLOWEEN - nejlepsi metalovou skupinu !!
```

(Don’t just sit at the computer. Do something rational for a change! Listen to HELLOWEEN, the best heavy metal group!!)

It then resets the computer. The virus has no further effects.

DIR II Virus

The DIR II Virus is unique. It is 1024 bytes long and is marked by the fact that although it infects COM and EXE files, it does not use them to modify other files. The virus occurs only once on infected disks. It originated in Bulgaria and has been fairly widespread in 1992.

When an infected program is executed, the virus installs itself in the memory and attaches itself to device drivers so that it is activated whenever any disk operation is performed. It uses the Strategy and Interrupt functions. Once installed, it starts the host program and ends normally. The memory resident virus then monitors access to disks and controls the Build BPB function (for the correct functioning of the CHKDSK program) and infects disks and directories.

The virus's infection mode can be divided into two parts. The first insures infection of the entire disk. The virus reads the last cluster on the disk, writes itself to this cluster and marks it in the FAT table as occupied. If this cluster is included in a file, the file is overwritten and destroyed. However, this is the only permanent damage that this virus can cause. The second part of the infection process leads to the modification of directories. The virus operates in the directory entry containing information about where the file begins on the disk (the first cluster pointer, FCP). The virus changes this parameter for all EXE and COM files, so that all programs begin with the virus code. The original value is coded and stored in a free (reserved) place in the directory entry. In this way the virus infects all files in the given directory simultaneously and spreads very quickly. The virus only checks extensions, not filenames, so that it even infects deleted files(!) The virus continually switches the FCP entries between the original and modified values, so that the system is able to continue operating. One side effect of this feature is the appearance of some Stealth characteristics.

If the virus is active in the memory, the computer runs normally. However, if the system is booted from a blank diskette, all infected files are modified to a length of 1024 bytes, and the CHKDSK program reports millions of errors (all programs begin at the same place). Infected diskettes have the same effect in virus–free computers.

Once the virus has been detected, infected files can only be backed up with difficulty. If the virus is in the memory, infected programs can be stored. If the virus is inactive, the entire disk is garbled. The VGUARD program does not remove the virus from the disk because this can be done quite simply by any user: when the virus is active, all COM and EXE files in all directories must be renamed (for example to *.CO and *.EX). The virus itself then restores respective directory entries to their original state. The same technique can be used to rescue programs on diskettes. Finally, the computer must be booted from the original diskette and all files renamed once again. The CHKDSK program can remove the cluster occupied by the virus. A program still containing the virus after completion of this operation is the probable source of infection.

Virus Jack Ripper

After activated this boot virus is installed under the level of 640 KB of main memory, decreases the remaining size of free memory with 2 KB, redirects the interrupt vector 13h and tests if the computer's hard disk has been infected. If not, the virus writes its code in the disk partition table (DPT). The second part of it is saved into the sector 8, head 0, track 0. The original DPT is located in the sector 9, head 0, track 0. The virus then writes into the memory the original DPT sector and hands over the control to it.

The virus monitors during writing in or reading from diskettes if it has not been attacked already. If not, the virus is saved into its boot sector and the last but one sector in the root directory. The original boot sector is saved into the following sector. Any operation with an unprotected diskette thus leads to its infection and further spreading of the virus.

Jack Ripper uses the stealth technique. If the virus is active, it monitors the requirements for reading and writing of the sector. Upon an attempt to read the DPT the original DPT is offered, upon an attempt to write into DPT the operation is not performed. When reading sectors 8 or 9 only zeros are read.

The virus contains the character string "(C) 1992 Jack Ripper". This string is coded both on hard disk and diskettes.

The harmful activities of this virus are very insidious. The virus when written into the sector changes with the probability of 1:1024 two randomly selected words in the written sector.

And as data is mostly written, this may lead to cumulation of unexplainable errors.

This virus is detected by LGUARD, RGUARD and AGUARD. It can be removed by FDISK/MBR (from DOS 5.0 version), as well as the BGUARD program (provided you have saved the original contents of the disk). From diskettes it can be removed by BGUARD.

Virus J&M (JiMi)

Virus J&M (Hasita) is a boot virus. After its activation it is installed in the end of the main memory and the remaining part of it is decreased by 2 KB. The computer hard disk is then tested for infection. Provided there is no infection, the virus saves its code into DPT and the original DPT is saved in the sector 6, head 0, track 0. The virus redirects the interrupt vector 13h, reads in memory the original boot sector and hands over the control.

The virus tests floppy disk operations, and if the floppy disk has not been infected, the virus writes its code into the floppy disk's boot sector. The original boot sector is transferred to the sector 14, head 1, track 0.

The virus contains the character string J&M. This string is not coded and the virus uses it for its identification.

After it is activated the virus tests the topical date. If the date is 15th November, it attempts to format track 0, head 0 of the first hard disk. If the controller allows this activity, the virus overwrites itself together with the original DPT.

This virus is detected by LGUARD, RGUARD and AGUARD. It can be removed using FDISK/MBR (from DOS 5.0 version), as well as by BGUARD (provided you saved the original hard disk contents in advance). It can be removed from floppy disks using BGUARD.

Virus One Half

This virus is memory residential, multi–partite, tunnelling, stealth and polymorphous. After activated the virus steps interrupt of 13h up to the DOS segment. Then it attempts to in–

fect the DPT. If successful, the virus saves its body in the last 7 sectors of the zero track, the original DPT is saved in the 8th sector from the end of the track and ends its activities. In case the infection of hard disk results in failure, it becomes residential immediately and attacks COM and EXE files longer than 1000 bytes. Files are attacked when these are run, opened or renamed, i.e. both on hard disks, floppy disks and network disks. The virus tests filenames and resigns from attacking files SCAN, CLEAN, FINDVIRU, GUARD, NOD, VSAFE, MSAV and CHKDSK.

After the system booting from infected hard disk the virus reserves the last 4 KB of RAM, is installed in the reserved memory and becomes residential. Now it attacks only floppy disk or network disk files. Files attacked are lengthened by 3544 or 3577 bytes. Infection symptom is a certain dependence between the date and the time of file origination.

The author of One Half was probably inspired by the Bulgarian virus Commander Bomber. Likewise, in this virus there is a decoding loop spread in ten sections randomly located in the original file. Separate sections are mutually connected by two types of jumps and random single–byte instructions are added to these. The whole structure of the decoding loop has two purposes. First, the files infected are without virus decoding in the memory dysfunctional and ordinary methods cannot be employed to remove the virus, while the virus is difficult to be detected using a text string.

Damage caused by this virus can be immense. After each system booting the virus xores the last two tracks of every surface of the hard disk with a random number generated upon installation in DPT. The number of the last coded track is saved in its DPT. After coding half of the disk the following report may appear depending on the date:

```
"DIS IS ONE HALF... PRESS ANY KEY TO CONTINUE".
```

Virus continues coding. The first third of disk is not coded by the virus.

One Half uses the stealth technique. If the virus is active in memory, xoring of disk and lengthening attacked files cannot be detected.

The virus can be removed from DPT both by the system program FDISK/MBR (from DOS 5.0 version) and BGUARD program. Attacked files should be deleted and replaced by those on backup copies. Prior to removal of virus from the disk, it is advisable to make backup copies of important data while the virus is still active in memory. Otherwise, the data can be placed in the coded part of the disk and through the removal of the virus from DPT a loss of the data may follow.

Virus Tremor

Tremor is a polymorphous virus attacking COM (length 8192 to 55039 bytes) and EXE (length 8192 to 1048576 bytes) files and is memory resident. When running an infected program the virus is first decoded and then tests the current date. If the period after infection is shorter than 3 months, or the file is in another directory than the one in which it was attacked, the virus modifies its own code and does not show any audio and visual effects. The virus tests its presence in the main memory using the interrupt of the 21h function 0F1E9h. If the virus is already active in memory or the DOS version lower than 3.30, control is handed over to the program attacked. In case the virus is not in the memory, it is installed in XMS or UMB memory. If none of the aforesaid operations is performed, it is installed on the top of the basic main memory, while it allocates 4288 bytes in the memory. Using 01h interrupt the virus tests possible presence of debuggers, finds the 21h interrupt address used for direct calling of the system core. Interrupt addresses 21h and 15h are redirected to unused area of MCB of the host program and then it goes directly to its residential section. Program specified by 'COMSPEC=' variable is infected, mostly COMMAND.COM and the original program is run.

Tremor uses stealth techniques. If the virus is active, it monitors the activity of the system and information that might lead to its detecting is given to the system in an altered form. For example, upon a request about the length of a file the original length of attacked files is reported etc. Upon program start–up the virus tests if the name starts with CH, ME, MI, F2, F– , SY, SI and PM. If so, changes are made in the memory allocation, resulting in CHKDSK returning as if correct values of the size of free memory. The virus does not attack programs starting with the characters SC, CL or HB. The virus also tests if the second and third character in the given filename are RJ. If so, it gives the system true information on files. It means that archives ARJ contain the virus, while ZIP files do not. Likewise, copies of clean and infected files created through the system COPY do not contain the virus, while copies made through Norton are infected. If the virus detects the presence of an antivirus program FLU–SHOT+, the file is not attacked and resigns to show any indication of itself. Tremor also tests the presence of the antivirus program VSAFE of DOS 6.00. Using special functions of 13h interrupt the virus can also put VSAFE into inactive state and activate it again after attacking the file.

The virus opens a file and reads the last twenty bytes. Then it decodes the bytes and if these contain the word “DEAD” and the date of the file is increased with 100 years the virus takes the file as already infected. Otherwise the virus adds its coded copy in the end of the infected program, redirects the initial jump or changes the value in the EXE file extension and runs the initial program.

Infected files are extended with 4000 bytes, the date of infected files is increased with 100 years. When calling the 15h interrupt the virus writes the given message. When calling the 21h interrupt the virus shifts the whole screen to the left and to the right by one character.

```
-=> TxRxExMxOxR was done by NEUROBASHER / May-June’92, Germany <=-
- MOMENT - OF - TERROR - IS - THE - BEGINNING - OF - LIFE -
```

The virus can be removed either by deleting all infected files and replacing these by those on original diskettes or using the AGUARD program (provided you use it regularly).

Virus 17.11.1989 (Pojer)

This virus was developed in Czechoslovakia. When run it is first decoded and detects its presence in the main memory. If it is not present in memory, it is installed in a standard way in the end of the main memory. Then the virus redirects the vector of 21h interrupt (DOS service) and in February, July, September and December on odd days it also redirects the timer (vector 1Ch). Eventually, it runs the host program.

This memory residential virus then monitors the function of program running and infects COM and EXE files run. It contains a table with names of programs it does not infect. These are SCAN, CLEAN and similar programs, on the whole 8 anti–virus programs. At the moment of infection the 24h interrupt vector is redirected (critical error), resulting in no system error messages upon unsuccessful attempts to infect a program. Infected files are longer with 1919 bytes, the virus is coded in a very simple way.

If the attacked program is run on 17. November, or 6. February (why?), the virus shows together with sound effects the following message and than continues its normal activity. Apart from that on days its own routine for timer (see above) is installed, in the top left corner of the screen it shows space and “oblong” repeatedly.

```
** BRAIN 2 v1.40 **
WARNING ! Your PC has been WANKed !
Viruses against political extremes, for freedom and parliamentary democracy
>> STOP LENINISM, STOP KLAUSISM, STOP BLOODY DOGMATIC IDEOLOGY !!<<
```

It is a very primitive virus, the author of which does not know much about politics, English as well as programming and morals. The virus contains a number of basic errors in all of the aforesaid areas.

Virus Civil Defense

This virus was most likely originated in Russia. It is designed so that it can work only on 286 and higher class computers, which it also tests.

The virus infects the DPT and EXE files. When starting–up an infected computer the size of memory is decreased by 7 KB and interrupt vectors are installed for timer, keyboard, printer and later for DOS. The 13h interrupt vector is installed as most other boot viruses do. Then DOS functions are monitored and upon searching files (Find First and Find Next) it infects EXE files on diskettes A or B. On floppy disks it first reads and then writes the boot sector. This serves as a test of write–protected disk. Infected programs secure spread of the virus from one computer to another. After running an infected program hard disk is tested for infection, if the hard disk is clean, the virus's boot sector is written, as well as the original DPT of the sector and other 12 sectors with the virus. Eventually, the virus removes itself from the infected and run program. The symptom of this virus in an infected file is the time of the last modification set to the value of 54 seconds.

This memory resident virus performs rather difficult activities, based on the "age" of the virus ranging from zero to five. Age 2 equals to approximately 275 hours of active virus in computer, age 5 equals to more than 375 hours. The virus counts the length of its activity in minutes, while this number is saved in a reserved area of CMOS memory. With increased age the interference activities of the virus are intensified. Let us describe what activities the virus performs in its 5th age. Three LEDs on the keyboard are flushing, a great number of errors of keyboard and syntax errors are simulated. Sometimes characters are not shown, sometimes more characters are generated. When pressing the function keys Soviet songs are played, when pressing Scroll Lock, screen turns red, and the message "Ha ha ha, slava KPSS, narod i partyia yedini, priviet ot GKCP" is shown, the former Soviet hymn is played and the virus resets the computer. When resetting (Ctrl Alt Del) together with audio effects on blue background a long Russian poem is shown signed by E. Letov (a "Grazhdanskaya oborona", this is where the name of the virus comes from, while the name of the virus and the version 1.1 are in the code too).

The virus also returns the DOS 2.00 version, so that many programs cannot work correctly. When running programs the following message is shown with the probability of 1:15 "Formatting disc c: complete. Format another ? (y/n)" and waits for a response. If the answer is Yes, the virus informs the hard disk has been formatted and then reads the sectors on the disk for a while. Do not format! If the answer is No, the virus writes a message "Eh, kak zhal, ved mye tak khatyeloss etto sdyelatt" and continues its activities. Twenty minutes after the computer is booted a yellow "piston" appears on the right that, accompanied by the sounds of a song, pushes the text to the left.

The axis of the piston bears a message “Vas privyetstvuyet virus CDV ver. 1.1...” Apart from these effects the virus also monitors what is printed on printer and certain Russian words replaces with other words.

V–Sign Virus

This boot virus is called V–Sign according to its manipulation routine. It has several interesting features. The virus takes up two sectors on disk and does not save the original sector, it only saves into it its own short loader, which is, after activation of the virus, overwritten by the original contents. Moreover, the virus contains (as one of the few boot viruses) slightly polymorphous features. In a cycle it replaces some loader commands so that these have different order.

When booting the system from an infected medium the virus loader first reads two sectors with the virus body in memory, allocates 2 KB of memory just under the level of 640 KB and saves itself in the memory. Then it modifies the 13h interrupt vector (disk operations), renews the original contents of the boot sector and hands over the control.

The virus monitors 13h interrupt and during operations read and write it can spread. If on the hard disk any sector on track 0, head 0 is read, the presence of the virus is tested upon the following operation. As for floppy disks the virus tests the first byte of the FAT table and thus specifies the diskette type, which is necessary for allocating a position to save itself.

V–Sign has another remarkable feature. Upon its installation in memory it tests the presence of Stoned boot virus in memory, while it can steel from it the original value of 13h interrupt and rewrite it in memory. Moreover, if it detects that the hard disk has been infected by itself already, it attempts to infect the sector in which Stoned saves the original boot sector. It is possible that by removing Stoned by some anti–virus programs following re–infection with V–Sign can take place. Removing one virus using another is not any news, but the V–Sign method is rather unique.

The manipulation routine consists in showing a big V on screen, composed of semi–graphic signs. The image is delayed, so that it can appear gradually. Then the program is cycled so that it cannot continue and the computer must be reset.

The manipulation routine occurs only rarely, i.e. only when 64 floppy disks have been successfully infected. However, as the counter is zero–set upon each installation of the virus in memory, the aforesaid number of floppy disks would hardly used during one session. Similar situation can occur upon formatting a large number of floppy disks or making backup copies of large disks on diskettes.

Other Viruses

New viruses are constantly appearing. They are reported in "Screen" magazine published by our firm, and also in the READ.ME file on our distribution diskette.

This page is intentionally left blank.

Removal of Viruses from the System

At this point we should like to offer some advice on what to do if you discover viruses in your computer. Research indicates that millions of users throughout the world have already met with computer viruses. Certainly nobody is immune.

The first thing to remember is not to panic. A hasty or ill–considered action can cause even more damage than the virus by itself.

If you are not fond of taking risks, you will already have backed up all important data and programs, so that even a total destruction of the content of your disk would be less than catastrophic. Moreover, you have almost certainly discovered the presence of a virus before it has done its worst. An active virus can be detected by noticing anything unusual in the behaviour of the computer: graphic or acoustic effects, strange error messages, unexpected disk behaviour, errors in program operations, and so forth. A surer method of protection is offered by the AVAST! package. Thus it is of crucial importance **to use the AGUARD program regularly**, and at **every configuration change** on the hard disk to **store the new data** with the BGUARD program!

If your SGUARD, AGUARD or FGUARD programs discover a mismatch which cannot be explained rationally, such as changes in file length or content, unusual disk operation, changes in the operating memory, etc., you should first use LGUARD and VGUARD to determine if one of the known viruses is present in the system. LGUARD should handle most today's viruses.

Once you have determined that a virus is indeed present, you must boot the system from the original system diskette that came with your computer or from the system diskette created during the installation of AVAST!. **This diskette must be write–protected** to insure that the virus is not active in the memory. Then it is important that all programs be run not from the hard disk but only from verified diskettes.

The best way of removing viruses is to delete infected programs and reinstall them from original distribution diskettes. Unfortunately, this is not always possible. Another method is to overwrite infected programs from the backup diskette. However, one infected program should be saved on a specially marked diskette for later use in identifying virus strains.

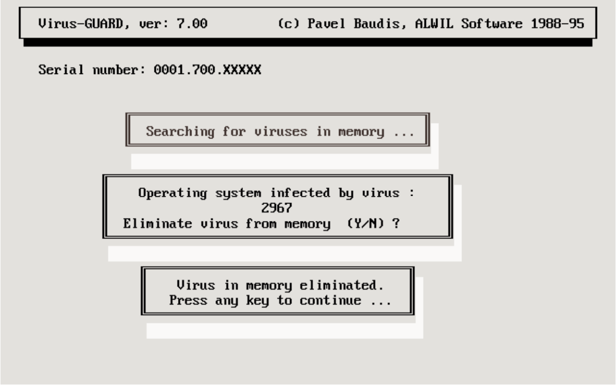
Removing Viruses with the VGUARD Program

A further possibility is to remove the virus with the help of the VGUARD program. Since VGUARD is able to locate some of the most widespread viruses in the Czech and Slovak Republics, it will probably catch whatever may appear in your system. If this virus has been located with the VGUARD program, then VGUARD should be used to remove it from infected files. Programs that have been destroyed by viruses must be reinstalled from original diskettes or backup copies.

However, it is usually better **to remove viruses from the system by using a general program**. The procedure is as follows: first insert the AVAST! working diskette that was created during installation of the package and write–protected. Then run the VGUARD program:

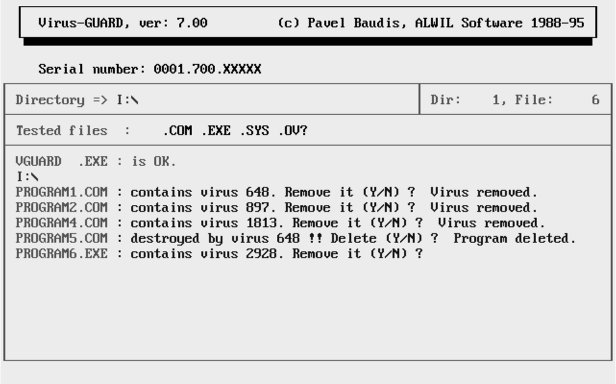
```
A:
VGUARD C:
```

First the VGUARD program scans the operating memory for resident viruses. If any such virus is found, it must be eliminated:



```
Virus-GUARD, ver: 7.00        (c) Pavel Baudis, ALWIL Software 1988-95

Serial number: 0001.700.XXXXX

Searching for viruses in memory ...

Operating system infected by virus :
2967
Eliminate virus from memory  (Y/N) ?

Virus in memory eliminated.
Press any key to continue ...
```

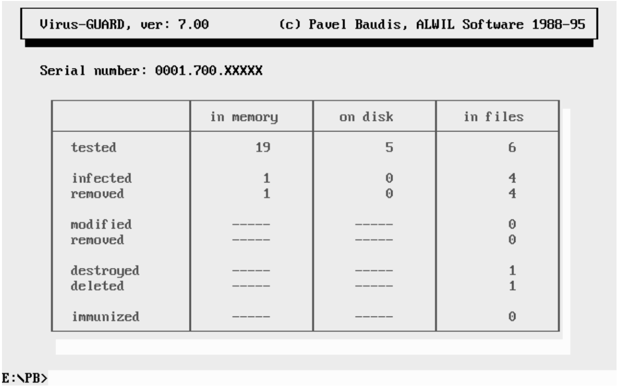
Once the virus has been eliminated from the memory it poses no further danger. Then the program checks the disk system area (which in this example is found to be in order) before proceeding to test the files:



```
Virus-GUARD, ver: 7.00        (c) Pavel Baudis, ALWIL Software 1988-95

Serial number: 0001.700.XXXXX

Directory => I:\                                 Dir:    1, File:    6
Tested files  :    .COM .EXE .SYS .OV?

VGUARD  .EXE : is OK.
I:\
PROGRAM1.COM : contains virus 648. Remove it (Y/N) ?  Virus removed.
PROGRAM2.COM : contains virus 897. Remove it (Y/N) ?  Virus removed.
PROGRAM4.COM : contains virus 1813. Remove it (Y/N) ?  Virus removed.
PROGRAM5.COM : destroyed by virus 648 !! Delete (Y/N) ?  Program deleted.
PROGRAM6.EXE : contains virus 2928. Remove it (Y/N) ?
```

The names of files that have been infected, modified, destroyed or immunized are listed. The user is prompted to remove viruses from infected and modified files and to delete destroyed files. We strongly recommend this action.

When VGUARD ends, it displays a table of results:

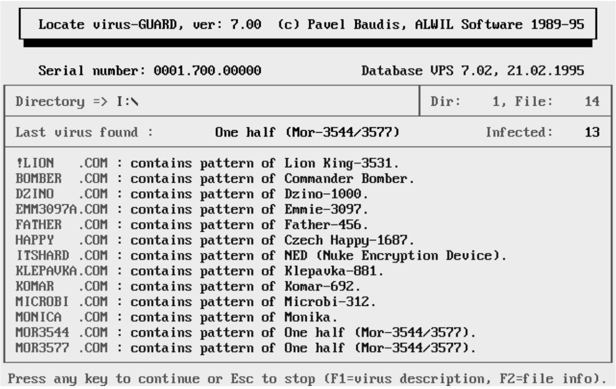


```
 Virus-GUARD, ver: 7.00        (c) Pavel Baudis, ALWIL Software 1988-95

 Serial number: 0001.700.XXXXX

   ┌─────────────────┬──────────────┬──────────────┬──────────────┐
   │                 │  in memory   │   on disk    │   in files   │
   ├─────────────────┼──────────────┼──────────────┼──────────────┤
   │  tested         │          19  │           5  │           6  │
   │                 │              │              │              │
   │  infected       │           1  │           0  │           4  │
   │  removed        │           1  │           0  │           4  │
   │                 │              │              │              │
   │  modified       │       -----  │       -----  │           0  │
   │  removed        │       -----  │       -----  │           0  │
   │                 │              │              │              │
   │  destroyed      │       -----  │       -----  │           1  │
   │  deleted        │       -----  │       -----  │           1  │
   │                 │              │              │              │
   │  immunized      │       -----  │       -----  │           0  │
   └─────────────────┴──────────────┴──────────────┴──────────────┘

E:\PB>
```

This table shows that one virus has been located in the memory and removed, and that the disk system area was in order. Of twenty files checked (including VGUARD.EXE), four were found to be infected and one destroyed. The virus was removed from the infected files, and the destroyed file was deleted.

General Virus Removal

There are a number of advantages to general virus removal. General methods work for unknown virus strains and are of–ten able to check the results of their work. The same is true of known strains, so that the following situation could occur:



```
 Locate virus-GUARD, ver: 7.00  (c) Pavel Baudis, ALWIL Software 1989-95

 Serial number: 0001.700.00000             Database VPS 7.02, 21.02.1995
┌─────────────────────────────────────────────────┬──────────────────────┐
│ Directory => I:\                                │ Dir:    1, File:   14│
├─────────────────────────────────────────────────┴──────────────────────┤
│ Last virus found :         One half (Mor-3544/3577)      Infected:   13│
├────────────────────────────────────────────────────────────────────────┤
│ !LION    .COM : contains pattern of Lion King-3531.                    │
│ BOMBER   .COM : contains pattern of Commander Bomber.                  │
│ DZINO    .COM : contains pattern of Dzino-1000.                        │
│ EMM3097A.COM : contains pattern of Emmie-3097.                         │
│ FATHER   .COM : contains pattern of Father-456.                        │
│ HAPPY    .COM : contains pattern of Czech Happy-1687.                  │
│ ITSHARD  .COM : contains pattern of NED (Nuke Encryption Device).      │
│ KLEPAVKA.COM : contains pattern of Klepavka-881.                       │
│ KOMAR    .COM : contains pattern of Komar-692.                         │
│ MICROBI  .COM : contains pattern of Microbi-312.                       │
│ MONICA   .COM : contains pattern of Monika.                            │
│ MOR3544  .COM : contains pattern of One half (Mor-3544/3577).          │
│ MOR3577  .COM : contains pattern of One half (Mor-3544/3577).          │
└────────────────────────────────────────────────────────────────────────┘
 Press any key to continue or Esc to stop (F1=virus description, F2=file info).
```

Should something like this happen to you (though certainly involving smaller numbers) you should copy the suspect program on a diskette and contact ALWIL software to have the virus analysed. You should do the same if you are convinced that your system has a virus even though VGUARD and LGUARD failed to detect anything.

For general virus removal it is then necessary to boot the system from a "clean" system diskette (preferably the one created during installation of AVAST!). Then with the AGUARD program you can easily discover any files which have been suspiciously modified. AGUARD can determine whether modified programs contain any of the known viruses. AGUARD also enables you to restore the original file, thereby removing viruses (see AGUARD description). We recommend that the whole process be followed by a second running of the AGUARD program so that you can determine if all files have in fact been restored to their original condition. This is one of the great advantages of the program.

The BGUARD program can also be used for general removal of boot viruses. It is capable of restoring the original content of system areas of hard disks from the diskette and of overwriting boot viruses found here. Then AGUARD can be used to determine if the system areas are truly in their original state. BGUARD can also remove boot viruses from diskettes.

Reporting the Discovery of Viruses

In many countries the discovery of viruses must be reported to the police. Such information is highly useful. It is important (for our firm and for the user alike) to know just what viruses have been found and how widespread they are. Therefore we have prepared a simple questionnaire found in the form of a file on the distribution diskette. A copy accompanies this documentation. We will be grateful if you could take a moment to fill it out and send it to us in the event that your system is attacked by an unknown virus. This information will be treated in strictest confidence and will be used only for statistical survey purposes. We thank you in advance for your cooperation in this important matter!

This page is intentionally left blank.

The AVAST! Built–in Protection

All programs of the AVAST! antivirus program contain a special mechanism by which they can check for their own modification by a virus. At start–up they first test whether they themselves have been changed, and if so they display the following message:

```
VGUARD.EXE
WARNING: This program was modified (maybe by some virus??) !!

Press any key to continue....
```

The user is thus warned in time. If any key is pressed, the program continues running. The virus is probably active in this moment however.

We sincerely hope that all computer viruses will steer clear of your system. If they do not, we trust that our programs will help you to overcome all the problems that they can cause.

This page is intentionally left blank.

Appendix

Caution:

ALWIL Software can assume no responsibility for any damage caused by a computer virus.

Rights of Users of our Programs

– In case of program malfunction or function at variance with this handbook or the READ.ME file on the distribution diskette, the user is guaranteed consultation in person, by telephone or in writing.
– Users may take advantage of our AVS antivirus service at low rates, paid for one year in advance.
– To expand the scope of the AVAST! package users should contact us at the address below.
– In case of virus attack, users have the right to a free consultation with ALWIL Software personnel in person, by telephone or in writing.

Our address:

ALWIL Software tel.: (+42 2) 782 20 50
Prubezna 76 fax: (+42 2) 782 25 53
100 31 Praha 10 BBS: (+42 2) 782 25 50
Czech Republic Internet: baudis@alwil.anet.cz

This page is intentionally left blank.

Index