VirusZ_II_English

COLLABORATORS			
	TITLE : VirusZ_II_English		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 27, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

VirusZ_II_English

1.1 Contents

VirusZ II 1.45

- English Documentation -

```
                    A list of supported Shell options.

                      Background
                    VirusZ's background features...

                        Menus
                    Usage of functions and preferences.

                      ARexx Port
                    ARexx commands understood by VirusZ.


                    Special Thanks
                    Just some hellos to good friends...
```

1.2 Important Note

```
                As there have been spread some VirusZ fakes in the past, I have ↩
                    included the
file  length  of  the  originally  released  program  version in the "About"
request.   If  you  are not sure whether you received a fake or an original,
extract your archive to a bootable disk first, switch off all your harddisks
in  the  BootMenu  and start VirusZ once from this disk (VirusZ requires the
commodities.library,  xfdmaster.library  and reqtools.library).  Now compare
the  length  in  the  "About" request with your file length.  If they match,
your  version  should  be  ok.  Otherwise don't install this version on your
harddisk.

If  you  want to get a 100% clean copy of the latest VirusZ version, see the

                Updates
                 chapter for addresses/phonenumbers.

If  you  have  some patches in your system that are not recognized by VirusZ
and  you  know  that  they don't do any harm, you can turn off the 'Check On
Startup'  option  in the Background preferences.  The surveillance mode will
then only report new changes after VirusZ has been already started.  If this
method  should work properly, you have to start VirusZ after all patches are
already  installed,  otherwise it would report the patches started later via
the surveillance mode.
```

1.3 Copyright & Disclaimer

```
The  entire  VirusZ  package  is  written and copyright © 1991-1999 by Georg
Hörmann  with  exception  of  the  reqtools.library  which  is  written  and
copyright © by Nico François who gave the permission to use it in any freely
distributable software package.

No parts of this package may be altered by any means (this includes editing,
reprogramming, crunching, resourceing etc.), except archiving.

The  author  is  in  no  way  liable for any changes made to any part of the
```

package, or consequences thereof as he is in no way liable for damages or loss of data directly or indirectly caused by this software.

1.4 Distribution

Neither fees may be charged nor profits may be made by distributing this piece of software. Only a nominal fee for costs of magnetic media may be accepted, the amount of US $6 shouldn’t be exceeded for a disk containing VirusZ. Non-commercial CD manufacturers like Fred Fish are allowed to put the package on their compilations, too. Outside a single machine environment, you are not allowed to reproduce single parts of the package, but you have to copy it completely. If any parts were already missing when you received the package, look out for another source to get your software in the future.

1.5 Updates

VirusZ is distributed in non regular times in Aminet. To get the really newest version, check either the Aminet itself or the Aminet CD’s.

You can find the program always in directory util/virus under the name VirusZ.lha.

1.6 Submissions

If you have found some new viruses and/or crunchers, send them to me right now. If you want your disks back, either enclose enough money for postage or German stamps. By now I had more expense than profit by sending all you folks your disks back. If you want me to continue my anti-virus work, don’t try to cheat me. Write to the following address:

Dirk Stöcker
Geschwister-Scholl-Straße 10
01877 Bischofswerda
GERMANY

stoecker@amigaworld.com

1.7 Introduction

This chapter provides a short overview of all features VirusZ offers.

First thing to mention is that VirusZ can be used as background program that checks memory and inserted disks for viruses. For this topic as for every other specific or global function, VirusZ offers a quite huge amount of user-definable settings to configure it as you like best.

Then you have the possibility to start different check mechanisms for files, sectors, vectors and bootblocks. This is what you usually do whenever you received new software from friends, PD-disks or from a BBS.

Finally, parts of VirusZ can be controlled via ARexx and thus enables the user to interact with it from programs like e.g. DirectoryOpus.

VirusZ has been designed as a 100% clean commodity without any system hacks, it follows the Style Guide conventions and provides you with keyboard gadget activation, great Reqtools requesters and many other useful features.

What VirusZ does NOT offer is localization and font-sensitivity. This has one major reason: One main purpose of VirusZ is and will be the background facility. Supporting different fonts or languages increases the disk and memory usage quite a lot, and that would make it unusable for people with just 1 MB of free memory.

1.8 Requirements

This version of VirusZ II requires:

- Kickstart 2.04 (or above)
- MC68000 (or better)
- commodities.library v37+
- reqtools.library v38+
- xfdmaster.library v37+
- rexxsyslib.library v33+ (for ARexx features)

1.9 Installation

Copy all libraries from the 'Libs' drawer of the VirusZ package to the LIBS: drawer of your system disk. You can use the 'Install Libs' script for the copy work.

Now you can either drag the VirusZ icon to your WBStartup drawer or add the following line to your 'S:User-Startup' file:

```
VirusZ [Options]
```

See the Shell Template section for a description of all supported options.

1.10 Shell Template

VirusZ supports the following Shell template:

```
CX_PRIORITY/N/K,CX_POPKEY/K,CX_POPUP/K,PUBSCREEN/K,FC=FILECHECK/K,
DE=DECREXEC/S,DD=DECRDATA/S,UE=USEEXTERN/S,EX=EXTRACT/S,ALL/S,AREXX/K,
```

QUIT/S

For more detailed information about Shell syntax, commodity usage and hotkey definitions, please consult the manuals shipped with your Amiga.

Please note that the ARexx interface commands described below require VirusZ to be active already. If it is not, it will first be started, the starter process will wait until the ARexx port appears and then the commands are sent to the port.

CX_PRIORITY:
Specifies the commodity priority of VirusZ's broker. Values may range from -128 to 127, default is 0.

CX_POPKEY:
Defines the hotkey used to pop up the main window.

CX_POPUP:
Tells VirusZ whether to pop up on startup or not.

PUBSCREEN:
Tells VirusZ to open its windows on the defined public screen instead of the Workbench.

FILECHECK:
This is not an option, but a shell interface to the ARexx command CHECKFILE. What makes it different from a simple call to CHECKFILE is that you can use wildcards of any kind and FILECHECK will list you all files that match with the given argument. The return code of this shell command is one of the following:
RC = 0 : Checking finished, no viruses encountered.
RC = 5 : One or more files are infected!
RC = 10: Error during file check.
Example: VirusZ FILECHECK "dh0:~(#?.info)" DECREXEC

DECREXEC:
This option is only of use with FILECHECK and tells VirusZ to decrunch executable files before checking.

DECRDATA:
This option is only of use with FILECHECK and tells VirusZ to decrunch data files before checking.

USEEXTERN:
This option is only of use with FILECHECK and tells VirusZ to use external slaves of the xfdmaster.library for decrunching too.

EXTRACT:
This option is only of use with FILECHECK and tells VirusZ to extract known archives and check their contents for viruses.

ALL:
This option is only of use with FILECHECK and tells VirusZ to scan all subdirectories that may exist in the given path.

AREXX:
This is not an option, but a shell interface to the ARexx port of VirusZ.

The argument given to this command will be directly sent to the ARexx port and the return code in the shell corresponds to the return code of the ARexx command.
Example: VirusZ AREXX "CHECKDIR dh0: SKIPDIRS"

QUIT:
This sends the ARexx command "QUIT" to the running VirusZ process and thus terminates it. This option is especially useful in scripts if you intend to check some files and remove VirusZ afterwards.

1.11 Workbench Tooltypes

For detailed information about Workbench tooltypes, commodity usage and hotkey definitions, please consult the manuals shipped with your Amiga.

VirusZ supports the following tooltypes:

CX_PRIORITY:
Specifies the commodity priority of VirusZ's broker. Values may range from -128 to 127, default is 0.

CX_POPKEY:
Defines the hotkey used to pop up the main window.

CX_POPUP:
Tells VirusZ whether to pop up on startup or not.

PUBSCREEN:
Tells VirusZ to open its windows on the defined public screen instead of the Workbench.

1.12 Background

In order to check inserted disks and memory in the background even ↩ if VirusZ
is working on e.g. a file check procedure, the backcheck is installed as a second task. This task does several things:

1. It scans through several memory locations and library/device vectors for all known viruses and resets any infected vectors.
2. It checks the bootblock of every inserted disk for viruses.
3. It checks the disk-validator of every inserted disk for viruses.
4. It is able to keep all system vectors under surveillance.

See the
Background
preferences for different settings.

1.13 Menus

```
                Whenever  VirusZ's  main interface is active, you will be able to  ↩
                   access the
following two menus:


                     Project               Prefs


                  File Check

                  File Check

                  Sector Check

                  Sector Check

                  Vector Check

                  Vector Check

                  Bootblock Lab

                  Bootblock Lab

                  Update Devices

                  Background

                  Show Brains...

                  Miscellaneous

                  About...

                  Archivers

                  Hide

                  Save Prefs

                  Quit
```

1.14 ARexx Port

```
                VirusZ has an ARexx port now.  The name of this port is 'VIRUSZ_II ↩
                   .REXX' and
it offers the following functions:


                       HIDE

                       QUIT

                    CHECKFILE
```

```
                     CHECKDIR
                Have a look at the example scripts in the ARexx directory if you ↩
                   want to get
an idea of the power of those few commands.

NOTE:   Starting  with  version  1.33, the CheckArc.vzrx script is no longer
        required  as  VirusZ can extract any archives itself now.  It's just
        still included as an example.
```

1.15 ARexx: HIDE

```
Syntax: HIDE

This function makes VirusZ close its main window and work in the background.
To get the interface back you have to use the defined hotkey or the Exchange
utility.
```

1.16 ARexx: QUIT

```
Syntax: QUIT

This function terminates VirusZ. All allocated resources will be released.
```

1.17 ARexx: CHECKFILE

```
Syntax: CHECKFILE File [DECREXEC] [DECRDATA] [USEEXTERN] [EXTRACT]

File is the filename (with path) of the file that should be checked.  Please
note  that  the  file  will  only  be checked and not repaired.  The options
DECREXEC and DECRDATA turn on decrunching of executable and data files.  The
USEEXTERN option enables the use of external slaves of xfdmaster.library for
decrunching too. EXTRACT turns on extracting of archives.

You will receive one of three results:
RC =  0 : Everything worked fine and file is not infected.
RC =  5 : File is infected by a virus!
RC = 10 : Error while checking file. This may be the result of a bad file
          specification, a bad option or an internal error (e.g. no memory).
```

1.18 ARexx: CHECKDIR

```
Syntax: CHECKDIR Dir [SKIPDIRS] [DECREXEC] [DECRDATA] [USEEXTERN] [EXTRACT]

Dir  is the directory that should be checked.  Please note that all files in
this  directory  will only be checked and not repaired.  Usually VirusZ will
scan  through any subdirectories that exist in the specified drawer.  If you
```

give the option SKIPDIRS, scanning of subdirectories will be skipped. The options DECREXEC and DECRDATA turn on decrunching of executable and data files. The USEEXTERN option enables the use of external slaves of xfdmaster.library for decrunching too. EXTRACT turns on extracting of archives.

```
You will receive one of three results:
RC =  0 : Everything worked fine and no files are infected.
RC =  5 : One or more files in the directory are infected by a virus!
RC = 10 : Error while scanning through directory. This may be the result of
          a bad Dir specification, a bad option or an internal error (e.g.
          no memory).
```

1.19 Special Thanks

There are several people I want to thank for supporting VirusZ:

* Flake/TRSI for viruses, patches, bug reports and the latest news from the net
* Jan Bo Andersen, Lars Kristensen and all the other guys at Virus Help - Team Denmark for viruses, translated docs and a fabulous support
* Holger Hesselbarth for patches, ideas and more
* Ralf Thanner for everything (what more should I say:-))
* Axel Folley for moral and financial support :-)
* Holger Wessling for his unbelievable fantasy
* Dave Jones for patches, viruses, ideas, bugreports and more
* Martin Huttenloher for MagicWB
* Martin Odaischi for dozens of viruses and financial support
* Heinz Lindner for resident tools and new Kickstarts
* Markus Stiebeling for bug reports and hints
* Rüdiger Prang for patches and TEX-Docs
* Steve/Silicon Designs 3003 for viruses and packers
* Jim Maciorowski for support, letters and donations
* all other folks that have contacted me in the past
* of course all users who already paid their shareware fee

1.20 File Check

Introduction

In the early days of the Amiga viruses, nobody thought about file or even link viruses. A good virus killer had to display the bootblock and check some vectors. But nowadays, the greatest danger doesn’t come from the bootblock, but from files. Therefore this quite unique file check has been created to check files for virus infection. It offers you several features which others lack. It can...

* decrunch data and executable files for checking (using xfdmaster.library),
* remove several viruses from one file in one step,
* remove linkviruses from any hunk of a file, not only from the first one,
* unlink so-called 4EB9-linkers (files will be split and then checked),

```
* strip hunk_name from executables (often used to hide viruses) AND
* extract any file-based archives (see
                Archivers
                 preferences)!!!
```

ATTENTION: Please never disable the decrunch option unless you have no other possibility because VirusZ can detect all built-in viruses ONLY in the decrunched state.

File Request

After selecting 'File Check' from the 'Project' menu, the first thing to appear is a file request. Here you (multi-)select the files and/or drawers you wish to check. If you want to select several entries, keep <SHIFT> pressed while selecting them. To select all entries, click on the 'All' button. Now click on 'Ok' to start or 'Cancel' to abort checking.

Output Window / Control Panel

Now a window opens that is separated in two parts. The bigger part is the output window which contains information about the files that are checked. The small part at the bottom is the control panel. By clicking on 'Stop', checking is interrupted and a request appears asking you to continue or to abort. If you select 'Continue', the request disappears and checking continues. By selecting 'Abort', checking is aborted and you can exit from the file check or select the next drawer/file by clicking on 'Check Again'.

Important Notes

The link virus removal code is absolutely reliable as long as infected files aren't damaged in any way. If the hunk structure is corrupted or anything else disables removing, VirusZ will tell you and then skip the file.
VirusZ handles the protection bits of files automatically, i.e. makes the file readable for checking and writeable for reparation. This is useful because you don't have to mess around with the Protect command in your Shell. Whenever there comes up a system request "Disk is write protected", VirusZ tried to change the protection bits. This access is not dangerous, so it would be best if you make your disks write enabled before checking.

Additional Hint

It may happen that a file is first infected and then crunched. If you want to save the cleaned file without having it decrunched, check it again with decrunching disabled.

1.21 File Check Preferences

```
                Skip Subdirectories
-------------------
```

Enable this option to make the file check ignore any drawers that may exist in a selected drawer.

Auto-Handle Viruses

If the file check detects a file that contains a virus, a request pops up to inform you which virus it was and asks you to either kill the virus or let it stay alive. With this option you can skip this request and kill any viruses automatically.

```
Test Without Save/Del
---------------------
```

If enabled, the file check only detects viruses, but doesn't try to save or delete the infected file. This may be useful with new disks you don't know the contents. Simply select all files, perform a file check and look at the output without being disturbed by requests. In fact it is useful for me to check through my virus drawers without aborting hundreds of requests.

```
Generate Report
---------------
```

This option makes it possible to create a text file that contains a copy of the text output you can see while checking. If enabled, a file request will appear after the file check is finished to ask you for the filename the report should be written to.

```
Auto-Save Report
----------------
```

If enabled, VirusZ doesn't ask for a path/filename to save the report to. It then simply uses the filename that is generated by default and the path entered in 'Default Report Path'.

```
Emulate ExAll()
---------------
```

This option tells VirusZ not to call the Kickstart ExAll() code, but to use an emulation instead. There are some Kickstarts around that have a bug in the ExAll() code, so if the file check doesn't report anything useful, turn on this option and try again. Nevertheless, try to avoid the emulation as much as possible because the real ExAll() command is a lot faster.

```
Decrunch Executables
--------------------
```

If this option is enabled, the file check decrunches executable files in order to check them for viruses.
ATTENTION: Keep this option enabled as often as possible. VirusZ can only detect viruses if the file is totally decrunched.

```
Decrunch Data Files
-------------------
```

If this option is enabled, the file check reads and decrunches data files in order to check them. This is useful for data files that actually contain executables, eg. XPK packed files.

```
Skip Crypted Files
------------------
```

If this flag is set, VirusZ will not ask you for passwords or keys if there appears a crypted file. This might be useful if you have protected these files yourself and know that there are no viruses in them. You don't have to respond to all the requesters then.

```
Use External Slaves
-------------------
```

This option enables the use of external slaves by xfdmaster.library. Please keep this option off as there don't exist external slaves at the moment that could be of use for file checking. The problem is that a lot of external slaves cause trouble like system crashes because of bad coding.

Extract Archives

This option makes VirusZ scan inside any archives that can be recognized and extracted with the information provided in the Archivers preferences.

Default Report Path

Enter the path where you want to save file reports to in this gadget. If auto-save is enabled, VirusZ uses this path for saving.

Amount Of Lines Displayed

This gadget contains the maximum amount of lines that will fit into the file check output window. Set to 99 on screens lower than 300 pixels and to smaller values on interlaced screens. Otherwise the scrolling will be too slow and decrease checking speed.

1.22 Sector Check

Select Drive

After selecting 'Sector Check' from the 'Project' menu, the first thing to appear is a drive request. Here you select the drive you wish to check. Only trackdisk units are supported, but checking should work with the new 1.76 MB disks too. Click on 'Ok' to start or 'Cancel' to abort checking.

Output Window / Control Panel

Now a window opens that is separated in two parts. The bigger part is the output window which contains information about the sectors that are checked. The small part at the bottom is the control panel. By clicking on 'Stop', checking is interrupted and a request appears asking you to continue or to abort. If you select 'Continue', the request disappears and checking continues. By selecting 'Abort', checking is aborted and you can exit from the sector check or select the next drive by clicking on 'Check Again'.

1.23 Sector Check Preferences

Auto-Repair Sectors

If the sector check detects an infected sector that can be repaired, a request pops up to ask you to either repair the sector or ignore it. With this option you can skip this request and repair any sectors automatically.

Check Without Repair

If enabled, the sector check only detects infected sectors, but doesn't try to repair them. Useful to get a quick overview over the sectors of a disk.

Amount Of Lines Displayed

This gadget contains the maximum amount of lines that will fit into the sector check output window. Set to 99 on screens lower than 300 pixels and to smaller values on interlaced screens. Otherwise the scrolling will be too slow and decrease checking speed.

1.24 Vector Check

Introduction

Mostly all viruses work in the same manner. Either they make themselves resident and/or corrupt some libraries or devices with their code. Therefore the vector check was designed to help you finding new viruses that can't be recognized directly by VirusZ yet.

Most of the vectors and entrypoints that will be displayed are only interesting for programmers, so I will try to avoid any explanations that confuse the average user.

VirusZ is able to display the names of library/device functions instead of printing just an offset message if you supply it with so-called FD files. These have been shipped with Workbench 1.2/1.3 Extras disks or can be found in most assembler and compiler packages. I haven't put them in the archive because of copyright reasons.

Output Window / Control Panel

After selecting 'Vector Check' from the 'Project' menu, a window opens that is separated in two parts. The bigger part is the output window which contains information about the vectors that are checked. With the scroll gadget at the right you can move the output up and down. The small part at the bottom is the control panel. By clicking on 'Refresh', the output will be refreshed. This is useful after clearing some vectors. If there is not enough memory to refresh, the vector check exits. With 'Exit', you normally leave the vector check. Use 'Prefs' if you want to change some settings while looking at the displayed information. If you leave the prefs window with 'Use', the display will be automatically refreshed.

What Can I See From The Displayed Information?

Well, every vector has a short comment right of it. As long as you can read 'Ok' there, everything is fine. Then it might happen that you read something like 'SetPatch', this tells you that the changes done to this vector are ok, because VirusZ recognized who did them.
But if you read '*** NON-STANDARD VECTOR ***', be alarmed. In fact, most of these unknown changes are nothing more than an utility like the well known 'PP Patcher'. But if you are sure that you haven't installed any system patches, this might be a new virus.
If '*** SUSPICIOUS ***' appears next to a process in the 'Suspicious Process Fields', this is very likely to be one of the new generation viruses that use these mechanisms to hide from normal vector checkers. Check your system carefully for misbehaviour and send me changed files as soon as possible.

Menu

There exists a menu called ’Clear’ in the vector check which offers you the possibility to clear certain vectors one by one or all together. The item names correspond with the respective vectors.
The ’Misc’ menu currently only offers one item: ’Save Report...’. This opens a file request where you may enter the name of the file to be saved. All output information displayed in the vector check window will then be written to this file.

1.25 Vector Check Preferences

Show ResModules

If enabled, the ResModules will be checked and non-ROM based modules will be displayed.

Show Exec Interrupts

If enabled, the exec interrupt table will be checked.

Show CPU Interrupts

If enabled, the CPU interrupt table will be checked.

Show Devices

If enabled, devices will be checked and non-ROM based function table entrypoints will be displayed.

Show Libraries

If enabled, libraries will be checked and non-ROM based function table entrypoints will be displayed.

Show Process Fields

If enabled, the tc_Switch, tc_Launch and pr_PktWait fields of every process are checked for suspicious entries.

Hide Known Patches

Normally the sector check displays known patches with their name after the patched entrypoints. If this option is enabled, known patches are skipped and will not be displayed. Useful to filter out modifications caused by SetPatch, LoadWB or other system commands.

Hide ’OK’ Vectors

If enabled, the vector check will not display ANY vectors marked ’OK’. This decreases the amount of printed lines drastically as long as there aren’t too much patches in the system.

Use FD For Offsets

If this is tagged, VirusZ tries to read the library/device function names from so-called FD files and displays them instead of saying 'Offset -1234'. Only if the function is not defined in the FD file (newer library version or reserved slot), the old offset message will be printed.

You may get these FD files from the WorkBench 1.2/1.3 Extras disk, or from almost every assembler/compiler package available for the Amiga. I can't include them in the VirusZ package for copyright reasons.

FD Path

This gadget contains the directory that holds your FD files. The files can be crunched with any data cruncher supported by xfdmaster.library and will be uncrunched while loading.

Amount Of Lines Displayed

This gadget contains the maximum amount of lines that will fit into the vector check output window.

1.26 Bootblock Lab

Attention

Be careful with writing to / installing your harddisk. I'm not reliable for your faults.

Drive / Display

There are two cycle gadgets in the bootblock lab, one on each side of the status line. The left one selects the drive you want to work with, the right one selects the display mode. Keyboard activiation of the drive gadget is <D> or <SHIFT-D> and <B> or <SHIFT-B> for the display mode gadget.

Name

Whenever there happens to occur an error, this will be stated in the status line. Then the name of the current bootblock in the buffer will be overwritten. By clicking on this gadget, the name is printed again.

Exit

Click to exit from bootblock lab.

Read

Reads the bootblock from the currently selected drive to the buffer. Only DOS disks can be read.

Write

Writes the current buffer contents to the bootblock of the selected drive. The disk type and the checksum will be corrected automatically.

Load

Opens a file request to select a bootblock file that should be loaded to the buffer. Only DOS bootblocks can be loaded.

Save

Saves the current buffer contents to a file. This is useful to backup important bootblocks of games etc.

Learn

This gadget will only be enabled if the bootblock in the buffer is neither a virus nor any other known bootblock. Then you are able to make VirusZ learn the unknown bootblock and give it a name. From now on, this bootblock will be reported with the given name and the background check will no longer report it as unknown.

Prefs

Opens the bootblock lab preferences window. Useful to change something without having to leave the lab.

Install

Installs a standard OS2 bootblock to the currently selected drive. The disk type will be corrected automatically.

Menu Functions

```
New Brain    - Removes currently loaded brain from memory.
Load Brain   - Loads new brain file to memory.
Save Brain   - Saves brain changes to file.
Merge Brains - Adds brain cells from a file to the currently loaded brain.
Edit Brain   - Here you can rename or delete brain cells.
```

1.27 Bootblock Lab Preferences

Ask Before Write Access

If enabled, a security request pops up every time you select 'Write' or 'Install' in the bootblock lab.

Read Inserted Disks

This enables the bootblock lab to read the bootblocks of inserted disks automatically. Useful if you intend to check a whole box of disks for bootblock viruses.

Install Uninstalled Boot

If enabled, 'Install' doesn't install a standard bootblock, but makes the disk non-bootable.

1.28 Update Devices

If you have some devices in your system that are only mounted on demand like recoverable ram-disks, MS-Dos floppies or SyQuest cartridges, they will not be detected by VirusZ after startup. To make these devices accessible for the Bootblock Lab, you'll have to update the internal device list with this function first.

1.29 Show Brains...

After selecting this function, a window will be opened with a list of all boot, file and link viruses detected by the current version of VirusZ. The second list shows all patches that will be recognized by the vector check.

1.30 Background Preferences

Check On Startup / Keep Under Surveillance

All the checkmark buttons under this headline have one thing in common: The first gadget switches the startup check on/off, the second (de)activates the surveillance mode. See below for the gadgets' meanings.

The memory scan routines recognize the same patches as the main vector check and therefore will not inform you of the changes done by these. If the info requester pops up to inform you about some changed vectors, go to the Vector Check and have a look at the changes if you want exact information.

If the bootblock check informs you about an unknown bootblock, go to the Bootblock Lab if you want to have a closer look at it.

ColdCapture

Checks for modifications done to the ColdCapture vector.

CoolCapture

Checks for modifications done to the CoolCapture vector.

KickTagPtr

Calculates a checksum over all KickTags in the list every time the check is performed and checks for differences to the last checksum.

CPU Interrupts

Scan through all hardware interrupt pointers relative to the vector base.

Exec Interrupts

Checks all the interrupt entries in the ExecBase.

Libraries/Devices

Searches for unknown patches.

Bootblocks

Checks the bootblock of every inserted disk.

Disk-Validators

Checks the disk-validator of every inserted disk.

Process Fields

Checks the tc_Switch, tc_Launch and pr_PktWait fields of every process for suspicious entries.

Known Viruses

Checks all memory locations for known viruses.

Surveillance Frequency

Enter the amount of seconds that should pass between two checks here. This frequency is used for the surveillance mode of all vectors and disks.

Report Known Bootblocks

Usually, bootblocks recognized by the brain are not reported (that's the main purpose of the whole brain system). But it may sometimes be useful to get those already known bootblocks reported anyway. If this option is enabled, that's excactly what will happen.

Check All Disks On Update

If the 'Update Devices' function from the main menu is called, it is likely that some devices may have changed. If this option is enabled, VirusZ scans all disks for such changes.

1.31 About...

Displays some information about VirusZ. You can see the file length your copy of VirusZ should have on disk at the bottom line.

1.32 Miscellaneous Preferences

Check Hunks On Startup

If enabled, the hunk structure of VirusZ will be checked on startup. An alert appears if there is something wrong (might be a link virus). Disable this option if you intend to crunch VirusZ with a file packer because most of these modify the hunks.

Requesters Follow Mouse

If enabled, all ReqTools requesters appear with the negative response under the mouse. If disabled, they pop up in the top left corner as usual.

Quit Immediately

If enabled, VirusZ quits without verification.

Install SnoopDos Task

If enabled, a task called ’SnoopDos’ will be created which doesn’t use any processor time, but prevents several trojan horses from doing any harm.

Pop Up On Startup

If enabled, VirusZ opens the main window on startup, otherwise it can be controlled via the Exchange commodity only.

Load Brain On Startup

If this option is enabled, the default bootblock brain (see below) will be loaded automatically on startup.

Close Main Window = Exit

If enabled, VirusZ quits when you click on the close-window button of the main window, otherwise it will act as if you selected the ’Hide’ item from the ’Project’ menu.

Center Main Window

If enabled, VirusZ’s main window appears centered at the top border of the screen. Otherwise it will use the coordinates that have been last saved. You can save the coordinates by moving the window to the desired position and then selecting ’Save Prefs’.

Activate On Startup

This option tells VirusZ to activate the main window on startup. This is useful for all users that don’t have VirusZ in the background all the time and want to start checking without activating the window first.

Hotkey

The default commodity hotkey used to pop up the main window.

Brain

The path and filename of the default bootblock brain. This will be used in the file requests of the bootblock lab and for loading the brain on startup (see above).

Devices

Enter all devices you want VirusZ to check here. They will appear in the BootLab in the same order as they are entered in the string gadget. If you enter a device name and VirusZ can’t find this device, it will be skipped. Thus you can enter all your devices, even if they are not always mounted. All names must be divided by a "|" character. Names are not case sensitive.

1.33 Hide

Makes VirusZ close its window and work in the background. You can re-open the window again by using the defined hotkey or via the Exchange commodity.

1.34 Archivers

Gadgets

New: Adds a new entry to the archiver list. This will only be taken to the list if all entered information is valid.

Delete: Deletes the selected entry from the archiver list.

Edit: You can edit the information of the selected entry again if it doesn’t work as you want.

Temp.Path: The path that should be used for temporary extracting should be entered here. Please note that any files in this drawer will be deleted during an archive check. Best thing to use is something unimportant like ’RAM:xyz/’ or ’T:xyz/’. If you don’t have much free memory and you want to extract big archives, use something like ’DH0:Temporary/’.

Archiver Information

Name: Enter a meaningful name for the archive type here (eg. LhA, LZX). This name also appears in the file check behind the filename of the archive.

Offset: This is the offset from the beginning of a file measured in bytes where the data should be searched for.

Data: The recognition data that determines whether a file is an archive or not. You can enter the data either as hexadecimal values ($xy) or as ascii strings (starting and ending with " or ’). Strings may contain several characters, hex values may only be in byte-range (one or two digits). Values must be separated with , . Overall length of data must not be larger than 16 bytes.

```
Example: LhA archives may be recognized with Offset: 2 and Data:
         * "-lh"         or
         * $2d,’lh’      or
         * $2d,$6c,$68   and so on.
```

If " or ’ should be used in a string, simply type "" or ’’ and it will be interpreted as single " or ’.

Please note that data will be translated to binary for internal use and may appear different from what you entered if you re-edit it later on.

Command: Enter the command line here that should be executed to extract the detected archive. Please use any options that an archiver provides to minimize the output of the program (eg. Quiet, No Messages).

The command line must contain the following parameters:
* %f: represents the filename of the archive to be extracted,
or
* %F: represents the filename of the archive with full path,
* %p: represents the temporary path to extract archive to,
or
* %P: represents the temporary path (ending with '/').

Example: C:LhA -mMIq x %f %P

Note that archiver programs must be called with their full path, because VirusZ doesn't support search paths set with the 'C:Path' command.

If a % should be used in the command line, simply type %% and it will be interpreted as single %.

Note that the archiver must be able to create destination drawers itself, as VirusZ doesn't do this automatically. This should be no problem at all as most archivers I know are capable of this.

If you have problems in creating a working single-lined command as I had with 'ZOO', simply create a script that does the work and enter a call to the script in the command line.

Example: C:Execute SCRIPTS:VirusZ_II.ZOO_Script %p %F

where 'VirusZ_II.ZOO_Script' contains the following commands:

```
.KEY PATH/A,FILE/A
C:MakeDir <PATH>
CD <PATH>
DH2:Tools/Zoo xqO <FILE>
```

If you still don't have any idea what I am talking about, simply copy the files in ENVARC/ and SCRIPTS/ to the corresponding drawers of your system, start VirusZ and edit the archiver paths to your needs. Everything should work fine then.

1.35 Quit

Terminates VirusZ. All allocated resources will be released. There will be no more virus checking.

1.36 Save Prefs

This saves the current settings to the file ’ENVARC:VirusZ_II.Prefs’ and the archiver list to the file ’ENVARC:VirusZ_II.Archivers’. From there, VirusZ will get its preferences on the following startups.