DiskSafe

Thomas Richter

COLLABORATORS

	TITLE : DiskSafe	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Thomas Richter	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

DiskSafe

1.1 DiskSafe Guide

DiskSafe Guide

Guide Version 1.24 DiskSafe Version 1.32

WARNING: The "LOGFILE" option of the previous releases was broken. Sorry about that. _____

Table of Contents

© THOR-Software

Thomas Richter

Rühmkorffstraße 10A

12209 Berlin

Germany

EMail: thor@einstein.math.tu-berlin.de

WWW: http://www.math.tu-berlin.de/~thor/thor/index.html

1.2 The THOR-Software Licence

The THOR-Software Licence (v2, 24th June 1998)

This License applies to the computer programs known as "DiskSafe" and the "DiskSafe.guide". The "Program", below, refers to such program. The "Archive" refers to the package of distribution, as prepared by the author of the Program, Thomas Richter. Each licensee is addressed as "you".

The Program and the data in the archive are freely distributable under the restrictions stated below, but are also Copyright (c) Thomas Richter.

Distribution of the Program, the Archive and the data in the Archive by a commercial organization without written permission from the author to any third party is prohibited if any payment is made in connection with such distribution, whether directly (as in payment for a copy of the Program) or indirectly (as in payment for some service related to the Program, or payment for some product or service that includes a copy of the Program "without charge"; these are only examples, and not an exhaustive enumeration of prohibited activities).

However, the following methods of distribution involving payment shall not in and of themselves be a violation of this restriction:

(i) Posting the Program on a public access information storage and retrieval service for which a fee is received for retrieving information (such as an on-line service), provided that the fee is not content-dependent (i.e., the fee would be the same for retrieving the same volume of information consisting of random data).

(ii) Distributing the Program on a CD-ROM, provided that

a) the Archive is reproduced entirely and verbatim on such CD-ROM, including especially this licence agreement;

b) the CD-ROM is made available to the public for a nominal fee only,

c) a copy of the CD is made available to the author for free except for shipment costs, and

d) provided further that all information on such CD-ROM is redistributable for non-commercial purposes without charge.

Redistribution of a modified version of the Archive, the Program or the contents of the Archive is prohibited in any way, by any organization, regardless whether commercial or non-commercial. Everything must be kept together, in original and unmodified form.

Limitations.

THE PROGRAM IS PROVIDED TO YOU "AS IS", WITHOUT WARRANTY. THERE IS NO WARRANTY FOR THE PROGRAM, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IF YOU DO NOT ACCEPT THIS LICENCE, YOU MUST DELETE THE PROGRAM, THE ARCHIVE AND ALL DATA OF THIS ARCHIVE FROM YOUR STORAGE SYSTEM. YOU ACCEPT THIS LICENCE BY USING OR REDISTRIBUTING THE PROGRAM.

Thomas Richter

1.3 About DiskSafe

"DiskSafe" is a tiny dos.library patch that, uhm, keeps disks safe - from invalidation by an accidential reset.

If you hit the reset key combination on the keyboard, the amiga usually stops all disk IO operation and does not update the disk, usually leaving it completely damaged. When booting again, the filing system tries to repair the damage - this is quick and O.K. for disks, but takes long for big HDs (usually around 20min per GB) and is thus not acceptable.

"DiskSafe" installs a patch that completes all disk IO before the reset actually is allowed to occure and thus leaves the disk validated, even when you hit reset within a disk IO operation. But in order to make this working, some special hardware must be present, which Commodore build not into ALL amigias, read the requirements !

However, DiskSafe 1.18 offers an "replacement" for the reset key that works for all Amigas. Check the Configuration section of this guide how to do that.

Starting with release 1.10 DiskSafe can be setup to protect the ColdReboot() library function as well, hence protecting the system from accidential software resets.

Release 1.12 introduces again new features: First, you may ask DiskSafe for a log file, listing all files that have been saved. Then, an additional key sequence has been defined to reset the computer without saving the disks. Finally, a better protection mechanism agains other calls writing to the disk has been included.

To understand better what exactly DiskSafe does, read the background .

1.4 Requirements

DiskSafe tries to cancel the reset signal until all disk IO has finished. To do this, some special reset logic must be present on your amiga, which Commodore in their infinite wisdom forgot to add to all computers...

It is safest to test DiskSafe first, before using it, because I can't give any waranty if this reset logic is installed in your computer. Up to my knowedge, it is present in:

o) the newer A1000

o) the A2000 (A to C) series

o) the A3000 to A4000, and the A1200 (thanks to the reports)

but is not present in *some* (depending on revision)

o) old A1000's

o) A500's

o) A600's

I haven't tested this program for the A3000 and A4000, but I guess they have the necessary hardware. Nevertheless it is worth a try, since C= manufactured quite a lot revisions of the A500's which differ in various ways.

Anyways, DiskSafe offers a work-around for this lack: A new reset keyboard sequence can be installed on request with the RESETKEY keyword, check the Configuration chapter of this guide.

Continue reading here:

A Short Test

and to run the full test, try:

Complete Test

1.5 A Short Test

How to test the Reset Logic needed by DiskSafe

To test the Reset logic of your computer, a tiny test program called "ResetTest" is included in the "Extras" drawer of this archive. Here is how it's done:

- Make shure DiskSafe is NOT installed.

- Open a shell window.

- Start the ResetTest program from the "Extras" drawer. A window should pop up.

- Hit the reset key combination:

- If you see a countdown running from 10 to 0, then printing

** POOF **

and finaly your amiga resets, the reset logic works and DiskSafe will work.

- If your computer resets immediately, without any countdown, the Reset Logic is broken and DiskSafe will fail. However, you may ask DiskSafe for a replacement reset sequence that is always safe - check the Configuration section.

If you found your reset logic o.k., you should run the complete test .

1.6 The Complete Test

How to test DiskSafe

First read the following steps, all at once and make shure you understand them. Some steps must be done FAST, and you can't continue to read this manual. THIS TEST HAS CHANGED AGAIN, so RE-READ and RE-EXECUTE IT!

- Take a new disk and format it, or use an old one which you no longer need. MAKE SURE THAT NO DATA YOU'LL NEED LATER ON IS CONTAINED ON THIS DISK SINCE IT MIGHT GET DESTROYED BY THE RESET - if DiskSafe fails to operate.

- Install DiskSafe. Copy the "CheckRoot" program of the "Extras" drawer to a safe place! Don't use the test disk for storing this program cause this disk might get damaged!

- Run DiskSafe with:

DiskSafe df0: logfile=RAM:log chunksize=8192

or, if no reset logic is present, use

DiskSafe df0: logfile=RAM:log chunksize=8192 RESETKEY

- Insert the disk likely to be trashed in your first diskdrive.

- Open a shell window.

- Locate a big (200K or more) bunch of data from the shell. DiskSafe itself is too small for testing... Every data will do it!

- Enter a copy command like the following:

copy file to df0:foo

where file is the name of the test file. Then press RETURN to start the copy operation. After the disk has started spinning, wait a while and

o) HIT THEN THE RESET KEY COMBINATION Ctrl-Commodore-Amiga if the reset logic is installed

or

o) HIT THE REPLACEMENT KEY SEQUENCE Tab-Commodore-Amiga if the short test failed.

- Watch what happens: If the computer immeditately starts booting, DiskSafe does not work - either because the reset logic is not present or some other patch broke DiskSafe. In the first case, try again with the additional RESETKEY argument and use Tab-Commodore-Amiga as reset key replacement.

If, however, the disk first continues writing and your amiga seems to ignore the reset, everything works fine.

A requester saying that the disk is write protected might appear before the computer starts booting because DiskSafe protects the disk by software. Ignore this requester!

IN ANY CASE: Remove the disk as soon as the busy light turns off. Do not mind if the computer warns you about that. The computer will now boot because the job of DiskSafe is finished, but do not mind about that - you might even have to remove the disk while booting.

- DO write protect the disk. (THIS PART OF THE TEST HAS CHANGED AGAIN!)

- Wait until the workbench comes up.

- Start a Shell.

- Insert the disk, and wait UNTIL THE BUSY LIGHT TURNS OFF. This MIGHT take a while. (AGAIN, THIS PART OF THE TEST HAS CHANGED!)

- Start the "CheckRoot" program of the "Extras" drawer like this to check the disk in the first internal drive:

CheckRoot df0:

Replace the "df0:" by the name of the DEVICE you inserted the test disk if necessary.

- Read the prompt of "CheckRoot". If it says "The root block is valid" then DiskSafe operated properly and the disk is valid. If the result is "The root block is invalid" then DiskSafe failed to work. If you get something like "Can't read the root block" then the disk drive is either not yet ready - wait a few seconds and try again - or is physically damaged. DiskSafe didn't work in this case, too.

Another test is is to check if DiskSafe can create a log file:

- Start DiskSafe again, with the same line as above:

DiskSafe df0: logfile=RAM:log chunksize=8192

Now look into the RAM: disk - a file "log" should have been appeared there. Use "type" or "more" to read it - it should contain the name of the destination file of the canceled copy operation above.

If you want to know more how DiskSafe works (or why it refuses to work), read the backgrounds .

1.7 Background of Operation

On every volume under control of the amiga filing system, a special data block called the "BitMap" block is kept. This "BitMap" stores the information which sections of the disk are free to use or already occupied by data - since you don't want to overwrite already existing files.

Whenever a file gets opened for writing, this "BitMap" is read into your computer's memory, to find out where the new incomming data can be stored - and it is not written back until the file gets closed, i.e. the disk operation is completed.

UNLESS, however, you press RESET during the file IO. In this case, only a part of the data gets written, but, even worse, the bitmap IS NOT WRITTEN BACK and the disk remains therefore invalid.

While booting, the filing system tries to repair the damaged BitMap, with more or less efford.

Now comes what DiskSafe does:

If you press reset, the reset is first captured by the keyboard device, which again informs DiskSafe and delays the reset, for a maximum of ten seconds (thus things must go fast). Alternatively, DiskSafe gets informed if you press the reset replacement sequence or call the ColdReboot() function.

However, this delaying of the reset signal does not work on all amigas, since some special hardware is required to do this. To keep production costs (and customer satisfaction) low, C= choose not to install this piece of hardware into every amiga on the market!

If, let us assume, the keyboard.device COULD postpone the reset, DiskSafe closes all files open for writing and flushes all disk buffers, thus writing the bitmap and leave the disk valid. If this operation completes, the keyboard.device is told to finally start the reset procedure, since the bitmap of all disks is now safe.

The logfile-creation is another "heavy-magic" operation: The list of open files is copied into a resident memory segment, to survive the reset. The actual log file is not written at reset time, since the disk might be quite busy, but by the next DiskSafe command locating the data left over, AFTER the reset. At this time the operating system is stable again, and the log file can be written safely.

REMARK: Experts might have noticed that I simplified the whole process how the disk validation and the filing process works, even how resets are delayed. I really know better, but I don't want to make things more complicate to understand, so have patience...

1.8 Installing DiskSafe

The installation process is quite simple:

Copy the "DiskSafe" program to your "C:" drawer, and the guide wherever you want.

After completing this, I recommand (!) you to test DiskSafe, read here for a short test.

If you found DiskSafe operates properly on your amiga, you might want to configure it.

1.9 Configure DiskSafe

After installing DiskSafe and testing it, you need to configure DiskSafe for your personal needs. A complete list of all shell arguments is available, too.

Edit the startup-sequence with an editor of your choice, and add above the "LoadWB" command the following line:

DiskSafe REBOOT drvs RESETKEY

The command line switch "REBOOT" is optional: Add it if you want protection against accidential software resets (by calling the ColdReboot() function), or leave it alone if you don't. I recommend adding a "REBOOT" - it does not cost more memory, it's just another patch DiskSafe adds to your system.

The switch "RESETKEY" is optional, too. It introduces a reset replacement sequence for all the folks that don't have a reset logic in their computer. Instead of pressing Ctrl-Commodore-Amiga, you should use the replacement sequence Tab-Commodore-Amiga. It resets your computer as well when this command line switch is present, and is always safe, for all Amiga models. Since it doesn't cost more memory, it's also recommended.

The drvs argument contains a list of all drives you want to save with DiskSafe. Consider the following rules when creating this argument list:

- Most important drives should go LAST, since they are saved FIRST.

- Slower drives should go FIRST, since they are saved LAST.

- If you install one partition of a drive, you should add all partitions. In particular, if you add one disk drive, add ALL.

The drive specifications must be given as DOS DEVICES. To put it in other words: VOLUMES or ASSIGNS WON'T WORK HERE!

A typical command line would look like this:

DiskSafe REBOOT df1: df0: dh1: dh0: RESETKEY

Please note the order!

WARNING: In order to work, DiskSafe patches some vectors of the dos.library plus the ColdReboot vector of exec.library, if you specified the REBOOT switch. It adds also an "inputhandler" if the RESETKEY option is given. Some virus checker programs might complain about this!

HINT: You may add a "chunk size" parameter to each device argument directly behind the colon, i.e. write "df0:11264" instead of "df0:".

More about this chunk size and more configuration options can be found in the complete list of all shell arguments.

1.10 Shell-Arguments

This is the complete list of all available shell arguments, as understood by DiskSafe:

SHOW

REBOOT

PATCHALERT

IGNORE

QUICKKEY

QUICKSEQ

LOGFILE

CHUNKSIZE

WAITVERIFY

VERIFYREQ

ENLARGEBUFFERS

RESETKEY

RESETSEQ

RESETPRI

DEVICES

including a description of the individual chunk sizes.

WARNING: There are more shell arguments than explained above. However, they are FOR INTERNAL USE ONLY. Don't call DiskSafe with them without good reason (i.e. without me asking you to do so)!

1.11 DEVICES

This argument defines the list of the devices that should be protected. To enable DiskSafe to really protect all devices, you should consider the following "rules of thumb":

- Most important drives should go LAST, since they are saved FIRST.

- Slower drives should go FIRST, since they are saved LAST.

- If you install one partition of a drive, you should add all partitions. In particular, if you add one disk drive, add ALL.

The drive specifications must be given as DOS DEVICES. To put it in other words: VOLUMES or ASSIGNS WON'T WORK HERE!

DiskSafe can be told to "cut" large I-O blocks in pieces, to allow quick abortion of large block transfers in case of an unexpected reset. This can be either done by the CHUNKSIZE option; or by appending the block size (or "chunk size") in bytes to the device

name, directly after the colon, i.e. specify "df0:11264" instead of "df0:". This limits the maximum block size of the first floppy drive to 11624 bytes. The explicit drive-specific limit will override the default value of the "CHUNKSIZE" option.

This option is most useful if you want to protect devices of quite different transfer speed, like floppies and hard drives. A small chunksize is appropriate for the floppy, but will slow down the HD; it's therefore better to give different limits for the floppy and the HD. A typical command line for this purpose will look like this:

DiskSafe df0:11624 dh0:1048576

BE WARNED! If you specify ONE or more individual chunk sizes, the Read() and Write() vectors of the dos.library will get patched. This will (should!) cause virus checkers to alert you about a possible virus. Due to this patch, the I/O transfer speed will go down somewhat, too. Since this patch is more complicated than the pure CHUNKSIZE patch, the slowdown will be even somewhat larger than with CHUNKSIZE only.

More informations about the I-O chunk size can be found in the CHUNKSIZE chapter.

1.12 REBOOT

You may tell DiskSafe to protect your drives by accidential software resets, too. Please add the "REBOOT" option in this case.

ATTENTION! Especially virus checkers alter usually the ColdReboot() system vector, to provide the installation of virii. However, exactly this vector must be patched by DiskSafe to make the software protection working. Thus, should you find this option non-working, please check your installed programs for this kind of applications.

1.13 PATCHALERT

If this option is present on the command line, DiskSafe will patch the system Alert()-vector to write back buffers before a guru will reset the system. However, this option is not without its quirks:

The good point is that DiskSafe will protect you against mild Gurus more reliable.

The bad points are that

- this doesn't work always. If the guru is caused by interrupt or supervisor code, DiskSafe is out of buisiness.

- if the system is just too much messed up, the DiskSafe patch might mess it up even more and might create just another guru. In this case, the DiskSafe reset handler might even delay the keyboard reset by at most ten seconds.

- the task that caused the guru and therefore entered the DiskSafe guru patch is not the task that will finally pass thru the guru to the system. While this is not a problem to the system, certain related patches might be confused by this. Especially, they *might* display "DiskSafe" itself as the task that caused the guru (which is technically correct because DiskSafe itself called the old Alert routine). These patches should be installed ON TOP of DiskSafe, i.e. run later in the startup-sequence. This goes for example for the "LastGuru" patch of the same author.

Conclusion: In most cases, this option is good. In others...

1.14 IGNORE

Add IGNORE to the command line to prevent DiskSafe from complaining about non-existing devices. This is useful if you boot with some of your HDs turned off.

The illegal device arguments are simply ignored in this case. The devices not valid at boot time, however, WILL NOT BE SAFED BY DISKSAFE, EVEN IF YOU MOUNT THEM LATER!

A better solution is therefore to add a mount list for these devices with the "mount" entry set to zero. These devices WILL be protected by DiskSafe as soon as they get mounted.

1.15 LOGFILE

You may request a logfile of the files that were open at reset time. For this, add "LOGFILE=file". The drawback of this log file generation is that it eats up more memory, cause all the file names must be kept in memory.

REMEMBER THAT THE LOG FILE WILL NOT BE WRITTEN WHEN THE ACTUAL RESET OCCURS. Instead, the next DiskSafe command with a LOGFILE argument will do this job.

To make this working, the so called "KickMemPtr" mechanism of the exec library is used. Again, some virus checkers might complain about this, or, even worse, might cancel the log file generation at all if they prevent programs from using these pointers.

1.16 QUICKKEY

You may ask DiskSafe for a quick reset, without saving data. This can be used to get a faster reset in case the SCSI or IDE bus broke down and DiskSafe can't operate anyhow. To enable the quick reset, add the command line switch QUICKKEY. The quicker reset is then obtained by pressing the left shift key first, and then, together with Shift held down, the usual reset combination.

You may also re-define the "QUICKKEY" by the QUICKSEQ option and use any other keyboard qualifier you prefer.

PRESSING SHIFT AFTER THE RESET COMBINATION YIELDS NOTHING since the keyboard is blocked by the pending reset signal at that time.

1.17 QUICKSEQ

DiskSafe 1.19 and up allows to specify the QUICKKEY used to specify a reset as "QUICK". This is the command line option:

QUICKSEQ=qualifier

The qualifier argument can be one or a combination of the following keyboard "qualifier" keys:

LShift the left "shift" key RShift the right "shift" key Ctrl the control key LAlt the left "alternate" key RAlt the right "alternate" key LAmiga the left "Amiga" or "Commodore" key RAmiga the right "Amiga" key

Combining these qualifiers is possible by placing a "+" sign between their names, i.e. "LShift+LAlt" specifies that you've to press both, the left shift and the left alt key to specify a reset as "quick".

Remember, these keys must be pressed TOGETHER with the actual reset combination, no matter whether this is the default Ctrl+Amiga+Amiga or any other replacement sequence.

This option defaults to "LShift".

1.18 RESETKEY

Some Amigas lack the required reset logic to block the keyboard reset for ten seconds. You may ask DiskSafe in this case for a replacement sequence you should use instead to reset the computer.

By adding the "RESETKEY" command line option, DiskSafe installs this replacement sequence which defaults to "Tab+Amiga+Amiga".

You may freely redefine this sequence by the RESETSEQ option.

1.19 RESETSEQ

This argument redefines the RESETKEY keyboard-reset replacement sequence, quite similar to the QUICKSEQ option.

The syntax for this command line option is as follows:

QUICKSEQ=qualifier+key

The qualifier argument can be one or a combination of the following keyboard "qualifier" keys:

LShift the left "shift" key RShift the right "shift" key Ctrl the control key LAlt the left "alternate" key RAlt the right "alternate" key LAmiga the left "Amiga" or "Commodore" key RAmiga the right "Amiga" key Num the key is a key of the numeric keypad

Combining these qualifiers is possible by placing a "+" sign between their names and must be pressed all together to generate the reset.

You may, additionally, add a regular key to the keyboard reset sequence:

Num a key on the numeric key pad Backspace the backspace key Tab the tab key Enter the enter key on the numeric keypad Return the standard return key Esc the escape key Del the delete key Help the "Help" key nearby. Up,Down Left,Right the cursor keys F1..F10 the function keys

or any other key by the character that is printed on top of it.

Thus, to give an example "LAmiga+RAmiga+Tab" is the default setting and requires three keys to be pressed, the "Tab" key with both Amiga keys.

Another example would be "RAmiga+RShift+Q" - reboots the computer if the right shift key is pressed together with the right Amiga key and the key "Q" on the keypad.

WARNING! Since DiskSafe must know where the "Q" key is on your keyboard - and since this might differ for national keyboards - DiskSafe should be run AFTER the default keyboard has been setup - or the american keyboard will be used to relate the character to its actual key on the keypad.

ANOTHER WARNING! Please note that you should not specify a qualifier that is already used up by the QUICKKEY option, or DiskSafe will generate a warning.

Qualifiers alone are possible as well: "LShift+RShift+LAlt+RAlt" reboots the computer with both shift and both alt keys pressed.

Finally, "LShift+Num+8" will cause a reboot if the "8" key on the numeric keypad is pressed together with the left shift key.

1.20 RESETPRI

This argument specifies the priority of the reset handler DiskSafe installs. It defaults to +16. Reset handlers with higher priority are run first, with lower priority later.

Due to system restrictions, the only acceptable values are +32,+16,0,-16 and -32.

1.21 WAITVERIFY

This option tells DiskSafe to hold the booting process if it finds a non validated disk. This may happen due to a crash that crashed the system so badly that DiskSafe wasn't able to write the file buffers back to disk.

If you specify WAITVERIFY, DiskSafe will wait until all drives given at its argument line are verified, so no disk-trashing occurs.

You may ask for a requester in this case, too, using the VERIFYREQ option.

1.22 VERIFYREQ

You may also ask for a requester if DiskSafe found a non validated disk during startup. To get it, specify the argument VERIFYREQ AND WAITVERIFY on the command line. DiskSafe will then, if one of the devices is not validated, pop up a requester telling you about what happend.

WARNING: You absolutely MUST specify WAITVERIFY as well, or you'll never see any requester.

Another WARNING: If more than one drive isn't verified, DiskSafe will only show a requester for the FIRST drive, not for all subsequent drives. The order which is used for checking is the same as for the saving process: LAST DRIVES GO FIRST! So the fastest drive gets checked first, and so on, up to the first device argument.

1.23 ENLARGEBUFFERS

If this option is given, DiskSafe tries to increase the number of disk buffers for filing systems validating broken disks. However, since the size of a disk buffer is undocumented, DiskSafe tries currently to estimate the disk buffer size by twice the size of the sector size of the device. DiskSafe will look for the largest continous block of memory available, and will make approximately half its size available as disk buffers for all validating devices.

However, not all filing systems implement the adjustment of the buffer size correctly, amongst them the FFS (Sigh!). DiskSafe *tries* currently to work around this problem, but if you encounter problems with this option, don't use it. It *should* work with the current V43 FFS, but I don't give any guarantee for third-party FFS patches ("V44 versions") or other filing systems.

1.24 CHUNKSIZE

It turned out that the workbench copies large files with one Read() or Write() call if enough memory is available. Such an IO operation can't be interrupted by DiskSafe, and if the device isn't fast enough to write the complete buffer, the disk might get damaged as well. To prevent these failures, you may ask DiskSafe to split large I/O operations in smaller blocks, allowing DiskSafe to abort the operation in time.

You have to specify for this extra the maximal acceptable chunk size that can be written at once. As a rule of thumb, the argument to CHUNKSIZE should be halve the number of bytes that can be written within the ten seconds of reboot delay time. A value of 11264 has been proven to work properly for the floppies.

BE WARNED! The CHUNKSIZE option may slow down the I/O throughput of your drive! Most modern HDs are fast enough so that you won't need this option. But IF you want to protect floppies or slower drives as well, make this option small enough so that the slowest drive is still safe! Try to estimate how many bytes can be written within five seconds, give this as argument. If the number is reasonably high, leave this option alone.

REMARK: With CHUNKSIZE set, DiskSafe patches the Read() and Write() vectors of your system as well. This might cause some virus checker programs to scream!

You may, additionally, specify individual chunk sizes for each device. More about this can be found in the DEVICES argument description.

1.25 SHOW

DiskSafe can print a list of all devices it added reset protection to. Call it from a shell like this:

DiskSafe SHOW

and you will either receive a note that DiskSafe is not installed, or a list of "safe" devices.

1.26 Troubleshooting

If things don't work....

Rule ONE: DON'T PANIC !

Rule of thumb: If the short test passed, IT IS VERY UNLIKELY that DiskSafe fails. Your computer has just proven that the reset logic IS present, so all necessary hardware is there. The only reasons left are software incompatibilities. Even without the reset logic, there's the workaround with the RESETKEY option and the Tab-Commodore-Amiga reset replacement sequence that should work always.

If the short test failed.... _____

Certain virus checker programs patch the reset handler mechanism of the keyboard device to make the installation of a virus impossible. This will also prevent DiskSafe from installing its reset handler. To test DiskSafe again:

- Turn off the computer.

- Reboot it with the startup sequence DISABLED. To do this, hold both mouse buttons while turning on the computer until the startup window shows up. Click now on the "Boot with no Startup-Sequence" gadget.

- Rerun the short test.

- If the short test works now: Remove all patches from the startup-sequence and from the WBStartup drawer. Re-install them again one by one to find out which patch causes the failure.

If the short test fails again, then the reset logic is broken. The failure might be caused by a custom keyboard that does not send the reset warning code, or by a defect, or non-installed reset logic. Try to contact a hardware technician if it is possible to add the reset logic.

As a last resort, you might want to use the RESETKEY command line option. It won't make the reset logic working, of course; it will, however, offer an replacement sequence Tab-Commodore-Amiga that resets the computer safely for all amiga models.

If the short test runs with success, but DiskSafe didn't operate as expected, or if DiskSafe doesn't work even WITH the RESETKEY option and you really used the replacement sequence...

Try the "ResetList" program in the "Extras" drawer. It lists the installed reset handlers, i.e. other programs that require the reset logic. Check the "Name" field of the output: The entries listed here should usually read "DiskSafe.Interrupt" and/or "trackdisk.device" - depending on your settings. If any other name appears here, and you find that DiskSafe doesn't work as expected, you should contact me.

Q: I get an error message like "xyz is not a valid DOS device" or "xyz is not a filing device" when installing DiskSafe.

A: You tried to install DiskSafe on a device that is either not a filing device, or you gave an assign or volume name as drive argument, or the device is currently not available.

How to find out?

Check the drive list as follows: Copy the "Devices" program of the "Extras" drawer to "C:" and run it with the same list of devices you gave to DiskSafe, e.g. like

devices df0: df1: dh0: dh1:

Read the output! You SHOULD get one big list of settings for each of the device arguments you provided - four in this example. Read the "Type" field of the output. Each of them should say "Device" and each device should have an item in the output list saying "ExecDevice", giving the name of the underlying device driver. Everything else won't work! No assigns, no volume names are valid here, no "non-filing" systems like PRT:, CON:. RAM: isn't possible as well (for obvious reasons, even though it can handle files).

If "devices" says "xyz not found", then either the provided device name is not valid (check for typos in this case!) or the device is not mounted, i.e. not accessible.

Removeable media SHOULD be mounted BEFORE you protect them with DiskSafe, even though no media is inserted!

If this is not possible for some reason, you MAY ask DiskSafe to ignore the drive in case it is turned off from time to time during startup. DiskSafe WON'T protect this drive UNLESS it is accessible during startup. Add the "IGNORE" option to the startup line, read the configuration section of the guide!

Q: I get the error message above with a device specification like "df0:df1:"

A: Insert blank spaces between the device names. They are needed as separator marks by the dos parsing routines!

Q: I get error messages with an argument line like

DiskSafe devices="df0: df1:"

A: Remove the double quotes. They group names with blank spaces together, hence asking DiskSafe to protect a device called "df0: df1:", in one word! Replace the line above by

---

DiskSafe devices=df0: df1:

without the quotes.

Q: DiskSafe failed to protect some drive I turned on later, after booting. I'm using the IGNORE option.

A: Sorry, I can't help here. All devices that should be protected MUST be accessible during invokation of DiskSafe. AT LEAST they should be mounted, even though they are turned off.

Q: I'm using a loud IDE drive that I'm parking from time to time to have a noisier computer. If this drive isn't accessible during the reset, i.e. parked, the computer will hang for ten seconds or starts unnecessarily this disk.

A: DiskSafe tries to safe your data to the drive by sending an CMD_UPDATE to each drive it should protect, and turns off the motor afterwards. If the drive isn't accessible for some reason, this system call might fail, hence causing the hang. I've currently no solution to this problem since I can't find out when an IDE drive is parked. The device driver opens as usual, without any error causing DiskSafe to assume that the drive is accessable.

The only work-around I can offer for now is the QUICKKEY option, read the configuration section of the guide. A special key causes DiskSafe in this case to pass the reset immediately, without protecting any drives.

Q: DiskSafe seems not to protect my floppies, even though I gave something like "df0: ..." as arguments to DiskSafe.

A: It is possible that your floppy drive(s) was simply too slow to complete the IO operation within the maximal ten seconds of reboot delay. If you want to protect your floppies, consider using the "CHUNKSIZE" option of DiskSafe - read on the configuration section of this guide.

A reasonable argument to "CHUNKSIZE" is 11264, for the usual Amiga floppies. BE WARNED: This WILL slow down all I/O operations a bit! Try to find out if this is acceptable for you, it's a speed/safety trade-off!

Q: I got a guru with some program I run. I reset the computer and got the reboot delay, but DiskSafe failed to protect my drive.

A: If the filing system gets damaged due to this failure, THERE IS NO WAY of saving the drive properly. I'm sorry, but I can't help you with this. It is a failure of the Amiga "OS" that it does not protect the filing system and its buffers by getting overwritten by some buggy software. Even DiskSafe can't help you in this situation. If the HD root block or the filing system gets lost, so are you! DiskSafe helps against accidential resets, not against buggy software!

Q: I get a considerably slow down when booting with DiskSafe installed.

A: Make sure all devices are mounted and ready for use when invoking DiskSafe. DiskSafe tries to access the devices, hence causing a mount operation if the drive isn't already acessable. This MIGHT be the reason of the slow down. If this does not help, contact me!

Q: Does DiskSafe work with the MultiFileSystem (MFS)?

A: Well, sort of. It protects only one filing system, namely the one active at invokation time of DiskSafe. This is usually the AmigaDOS OFS/FFS. If somebody really needs full protection, let me know!

Q: Does DiskSafe work with other filing systems as well?

A: I don't know as I haven't checked it. But it should, there's not much magic in the operation of DiskSafe. As long as your filing system supports the dos packet ACTION_FLUSH, everything should go fine. Try and ask the author of the filing system about it! Don't worry if you don't know what this means, she or he will know!

Q: What's the bug in the FFS you mentioned in the guide?

A: The ACTION_FLUSH packet does not operate as it should. The AmigaDos manual states that this packet "causes the file system to flush out all buffers to disk before returning this packet. If any writes are pending, they must be processed before responding to this packet. This packet allows an application to make sure that the data that is supposed to be on the disk is actually written to the disk instead of waiting in a buffer." (AmigaDos manual literally!)

THIS ISN'T TRUE! It returns immediately, without any error code. The data IS written back to disk, but some time AFTER the packet has been returned. This is a bug in the multithreading of the FFS, which hasn't been fixed, even in newer releases of the FFS than the 40.1 which came with the workbench 3.1.

Q: DiskSafe hangs if I try to protect partitions that have been turned into Linux partitions.

A: No idea what's going wrong here, sigh. Just don't try this.

Q: What to do if DiskSafe does still fails to operate?

A: Contact me per email or SnailMail. Please provide the following information:

- The version of DiskSafe you're using. Should be 1.14 or better!

- The version of the workbench. 2.1 should suffer, but please report anyways.

- The output of the "devices" program on your list of devices to be protected.

- Your computer hardware: Which model, which interfaces (SCSI/IDE?), which additional disk/hd/cd... whatever IO related hardware. Printer/monitor/mouse won't matter. Keyboard DOES matter! If known: Board revision of your computer.

- Which software are you working with that gets installed during startup:

Checklist: virus checkers, disk encryption programs, disk speeders,...

1.27 The DiskSafe API

DiskSafe provides now an API for external programs that might be interesting in its function, namely virus checkers and other reset handlers or programs that depend on its function.

Here is how it works:

Your program should check the public port list for a port named "DiskSafe.rendezvous":

port=(struct DiskSafePort *)FindPort("DiskSafe.rendezvous");

If this port is present, DiskSafe is installed. The port structure looks like this:

struct DiskSafePort { struct MsgPort dsp_Port; struct SignalSemaphore dsp_AccessSemaphore; UWORD dsp_Version; UWORD dsp_Revision; void * dsp_DeviceRoot; struct MinList dsp_ImmediateHandlers; struct MinList dsp_DeferedHandlers; void * dsp_OldOpen,dsp_NewOpen; void * dsp_OldClose,dsp_NewClose; void * dsp_OldRead,dsp_NewRead; void * dsp_OldWrite,dsp_New void * dsp_OldColdReboot,dsp_NewColdReboot; BOOL * dsp_ResetPending; ULONG * dsp_OpenCount; };

Make sure the dsp_Port.mp_SigTask field is NULL. This field is unrelated to the DiskSafe function itself. Provided DiskSafe is installed, it is NULL and will remain NULL in all situations. DO NOT scan anything in here in case it's not, just exit. DO NOT wait for this longword to become non-NULL, it NEVER WILL in older DiskSafe releases. You have been warned!

Version and Revision inform you about the current release of DiskSafe installed. Virus checkers might want to test whether this coincides with the version information of the DiskSafe program in the C: directory and should be alert if this is not so.

DeviceRoot points to an internal list of the patched devices. Do not parse it yourself or assume anything about how it looks like.

ImmediateHandlers and DeferedHandlers are lists that provide a mechanism of calling your code before and while DiskSafe is performing its job. Both lists contain hook structures (see utility/hooks.h) that will be called before DiskSafe starts closing files for the ImmediateHandlers, or after having closed the files, but before shutting down the handlers for the DeferedHandlers. To attach a hook to these lists, arbitrate the AccessSemaphore first with ObtainSemaphore(), then add the hook to the lists, then release the semaphore. Don't hold the semaphore for too long since this prevent DiskSafe from working.

All fields below contain the original and patched system entries installed by DiskSafe. A virus checker might make use of them.

The next two fields exist for DiskSafe 1.30 and up. You HAVE to check the version and revision before reading them. Write access is considered illegal:

dsp_ResetPending points to a word which is non-zero in case a reset is pending and DiskSafe is currently busy closing open files. Do not play with this flag!

dsp_OpenCount points to a long which is the number of files open for writing DiskSafe will be able to shut-down on a reset. However, even if this long word is null, this says effectively nothing. There's little reason to play with this word. First, even if all files are closed, a process might still start write access to the disk, either by still writing out pending disk buffers, or by any other write operation as deleting or renaming files. Furthermore, it is not guaranteed that this long word WILL stay valid, it may easely happen that another process opens a file for writing just immediately after you scanned it. Therefore, there's very little reason why you should read or depend on this long word at all. If you want to reset your system, just do it. That's what DiskSafe is good for... (-;

1.28 DiskSafe History

DiskSafe 1.03:

First AmiNet Release.

DiskSafe 1.04:

Bug fix! Found a horror bug in the FFS - ACTION_FLUSH does NOT update the disk like it should! Argh! Thank you, Gene, for reporting!

DiskSafe 1.05:

Added support for removable media. It is now possible to add external devices with no medium in it provided they are mounted. Add a mount icon in DEVS:DosDrivers for this purpose.

DiskSafe 1.06:

DiskSafe is now able to launch itself, RUN is no longer needed.

DiskSafe 1.07:

Minor bugfix of 1.06: Due to a typo in 1.00, the output of a warning message was broken. Again a "thank you" to Gene Heskett.

DiskSafe 1.10:

Added ColdReboot() patch and Shell arguments REBOOT and SHOW. The background code prints now warning messages, if DiskSafe cannot be launched.

DiskSafe 1.11:

Filled a tiny gap in the disksafe protection: Delete(), Rename(), Protect() and other calls that may write to the disk are now forbidden after a reset signal has been caught.

DiskSafe 1.12:

Added IGNORE,QUICKKEY and LOGFILE command line options. Especially the last one is very tricky. Thanks for the ides goes to Nils Goers (IGNORE option), Christoph Bielachowicz (QUICKKEY option) and Fabio Vitale (LOGFILE option).

DiskSafe 1.13:

The resident logfile creation failed to work with FastExec, since it got overwritten by the supervisor stack on machines with no autoconfig fast mem. This problem *should* be gone now! Thanks to Luca Longone for reporting and Harry Sintonen (FastExec) for his very useful remark about the bug.

DiskSafe 1.14:

Added the CHUNKSIZE option and two extra programs. Added a troubleshooting version to the guide.

DiskSafe 1.15:

Added the WAITVERIFY and VERIFYREQ command line arguments. Thanks to Steffen Clemenz for the idea.

DiskSafe 1.16:

The chunksize can now be given individually for each device. The SigBit of the DiskSafe.rendezvous port is now set to 0x00 instead of 0xff, as requested by Andreas Kleinert. (Might have caused problems for two of his programs.)

DiskSafe 1.17:

All previous versions could have failed to create a log file for KickStart versions V37 and V38 (so, 2.0 and 2.1). That should be fixed now.

DiskSafe 1.18:

Added the RESETKEY combination for systems with a broken reset logic. Thanks to Werner Mueller for the idea.

DiskSafe 1.19:

Added the RESETSEQ and QUICKSEQ arguments to allow adjustable reset and quick key combinations.

DiskSafe 1.20:

DiskSafe detects now a possible conflict between the QUICKSEQ and the RESETSEQ and prints a warning in this case.

DiskSafe 1.21:

The individual chunk size specifications conflicted with the WAITVERIFY option. Further more, I included the ResetList program.

DiskSafe 1.22:

Added even more sanity checks for the device drivers on startup, fixed a bug on startup.

DiskSafe 1.23:

Found that I could save as much as four bytes per open file (wow-thers!). Fixed the behaivor if allocating a new file handle for MODE_READWRITE failed due to lack of memory.

DiskSafe 1.24:

Found and removed a bug in the startup code, missed one bracket. Added an option to setup the priority of the reset handler.

DiskSafe 1.25:

Added an API for external programs like virus checkers and shutdown handlers.

DiskSafe 1.30:

Enhanced the API a bit. Added the PATCHALERT argument to capture gurus directly. Added the ENLARGEBUFFERS argument to increase the number of disk buffers for validating filing devices.

DiskSafe 1.31:

The version number in the API used to be incorrect. DiskSafe hash key calculation is now slightly more efficient. Added compatibility hack for ancient programs that don't initialize a6 correctly.

DiskSafe 1.32:

Urghl. The LOGFILE option of the previous releases caused ERROR_NO_FREE_STORE reports in case one of the directories on top was locked exclusively. This has been fixed now, but the logfile contains in that case only "(locked):" as device name because that's all DiskSafe can find out about the directory name in this situation. Sorry to all who reported that bug, and thanks to Burhard Breuer for giving me the hint.