BootX

Peter Stuer

COLLABORATORS

	TITLE : BootX			
ACTION	*NAME*	*DATE*	*SIGNATURE*	
WRITTEN BY	Peter Stuer	February 6, 2023		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

BootX

1.1   BootX Help

Welcome to the BootX AmigaGuide®. Select any of the following

topics and follow the links to other subjects as you please:

1.2   About AmigaGuide

The Contents gadget will take you back to the last help menu

you visited.

Click Retrace to go back to the help screen you came from.

Browse forward and backward enable you to walk through the

help document.

1.3 Gadgets

This page provides you with help about all the gadgets that

can be found on the BootX screen.

1 Quit

2 Read Bootblock

3 Write Bootblock

4 The drives cycle gadget

5 Lock Drive

6 Check Files

7 Show installable bootblock

8 Built-in bootblock name

9 Install

1.4 Menus

Here are all the menus BootX provides.

1 Project

2 Check

3 Bootblocks

4 ToolBox

5 BBLib

6 Brain

7 Preferences

1.5 The Project Menu

The Project Menu provides access to some of the basic

actions during a BootX session.

1 Load Recog file

2 Show known

3 Print known

4 Iconify

5 Help

6 About

7 Quit

1.6   The Check Menu

The Check menu concentrates all the possible checks that BootX has
to offer.

1 Check blocks

2 Check memory

3 Check vectors

4 Reset vectors

5 Check files

6 Show report

7 Save report

8 Print report

9 Check disk

1.7   The ToolBox Menu

In this menu you will find some utilities that might come in handy
during your work.

1 Filter

1.8   Filter Utility

The Filter utility allows you to create a filter of path names
which BootX will ignore during a file check. The paths that are
part of a filter typically contain only data files like documents,
modules, IFF drawings, include files or other binary files.
Programmers might want to put their Include and Fd directories in
the filter. The Fonts directory is another good candidate.
Make sure you have checked the Filter checkbox in the Check Files
preference requester or the Filter will not be used during a file
check.
Don't forget to save your preferences after you have created or
modified a filter. BootX stores the names of the paths to be
filtered in the BootX.prefs file.
NOTE
The filter can be a mixed blessing. It will speed up
file checks since entire directories will be skipped.
However if you add a path to the filter that contains
executables and one of the executables is infected by

a virus it will not be found. Be carefull.

1 New

2 Add

3 Edit

4 Remove

5 OK

1.9   The Filter Utility New gadget

Click this gadget to start a new filter.

1.10   The Filter Utility Add gadget

When you click on the Add gadget BootX will present you with a
directory requester. Walk through your directory tree until you
are in the directory you wish to filter. Select OK when done. The
newly selected path will show up in the Filter listview.
If you have selected an entry from the Filter listview it will be
used as starting point of your tree travels.

1.11   The Filter Utility Edit gadget

This gadget can only be selected when you have clicked on one of
the Filter listview entries. BootX will present you with a
directory requester with the selected entry as starting point of
your tree walk. This allows you to edit existing path filters.

1.12   The Filter Utility Remove gadget

After you have selected an entry from the Filter listview, the
Remove gadget will become enabled. Click on it to remove the
current selected filter entry.

1.13   The Filter Utility OK gadget

Select the OK gadget to finish editing the path filter.

1.14   The Bootblocks Menu

Select one of the following subjects concerning the

Bootblocks menu:

1 Load/Save

2 Learn

3 Show again

1.15   The BBLib Menu

When BootX is started it will look for a ${"file" LINK GlossaryF} called BootX.BBLib .

This Bootblock library contains custom bootblocks that you can use

instead of the standard OldFileSystem or FastFileSystem

bootblocks.

With the Bootblock library system you can create several different

libraries that contain your collection games, demo or utility

bootblocks. The libraries are much easier to maintain than

several single files. If one of your autobooting game or demo

disks get infected by a bootvirus , you can always use the copy you

saved in your library to restore the disk.

You can install any bootblock in the library by selecting it using

the 'Show installable bootblock' gadget and then clicking on the

'Install' gadget.

1 New

2 Load

3 Save

4 Show

5 Merge

6 Add bootblock

7 Delete bootblock

8 Rename bootblock

1.16   The Brain Menu

At startup time BootX looks for a file called BootX.Brain . This

Brainfile contains all the bootblocks you have taught BootX using

the Bootblock/Learn menu item .

This Brainfile complements the knowledge base of BootX by letting

BootX recognize any new bootblock that are not yet recognized by

the BootX.Recog file.

NOTE

If you find new bootblocks that BootX does not yet

recognize please send them to me so that I can update

the Recog file. BootX will always give precedence to

the conclusion of the Recog file when it tries to

recognize a bootblock.

1 New

2 Load

3 Save

4 Show

5 Merge

6 Delete Brain cell

7 Rename Brain cell

1.17   The Preferences Menu

Want to change your preferences ? Look here for help:

1 Edit

2 Load

3 Save

1.18   Hints, tips and tricks

Here are some hints and tips on how to use BootX...

1 How to kill disk-validator viruses

2 Using BootX with little memory

3 Copying bootblocks with one drive

1.19   About BootX

BootX is a powerful viruskiller for the Commodore-Amiga computer.

It is fast (written completely in assembler) and tries to be as

user friendly as possible.

This program is freeware, this means that you can copy it

freely as long as you don't ask any more money for it than a

nominal fee for copying. If you want to distribute this program

you should keep this document with it. This program cannot be

used for commercial purposes without written permission from the

author.

Special permission is hereby granted to include BootX in

Public-Domain collections such as Fred Fish's Amiga Library.

BootX software and documentation are © 1989, 1990, 1991, 1992

by Peter Stuer. All rights reserved.

1.20   System requirements

BootX should run on any Amiga system with at least 512K RAM and

one disk drive. BootX requires KickStart v2.04 or higher to run. If

BootX is run with Workbench v2.1 or higher it will use the

locale.library to become language independent. Special care has been

taken in the program design to make the program work with 68020,

68030 and 68040 processors without sacrificing 68000

compatibility.

Should you have any trouble running BootX on your machine, please

write to me with the full specifications of your machine, that is

KickStart version, model, expansion boards etc...

BootX also requires the ReqTools.library v38 or higher. ReqTools

is copyrighted 1992 by Nico François.

If you want to save and load files that have been crunched with

PowerPacker you will also need the PowerPacker.library v35 or

higher. PowerPacker is copyrighted 1992 by Nico François.

To decrunch Library Imploded files during a file check you'll need

the Explode.library v6.0 or higher.

To use the help feature BootX requires the AmigaGuide.library to

display the BootX.guide. This library comes with WorkBench 3.0.

1.21   About the author and the program

Since so many people keep asking me about it... I graduated in

June 1991 as a programmer-analyst. BootX was created because at

the time (April 1989) I did not trust the available viruskillers.

In those days only the SCA and the Byte Bandit bootviruses

where a threat to our bootblocks .

If you have suggestions or remarks about this program, or if you

find any bugs , please let me know. If you find any new viruses or

linkviruses that BootX does not recognize yet, please send them on

a disk to me (my address follows this paragraph). I will return

the disk to you with an updated copy of BootX.

When sending in bug reports, please state exactly under what circumstances the bug occurred, what equipment was used and what happened. If possible also try to give me enough information to reproduce the bug. It is very difficult to find bugs when you don't know exactly what happened.

Write to the following address:

Peter Stuer

Kauwlei 21

B-2550 Kontich

Belgium - Europe

Fido: 2:292/603.7

1.22   Getting updates

If you send me a 3½" disk and a letter to get the most recent version of BootX, please don't forget to include enough stamps or money to send your disk back. BootX has been generating a lot of mail and most of the times I had to pay the postal fee myself to send the disk back. Remember, BootX itself is a FREEWARE so no money is expected for it (although I won't refuse it, grin...). Every letter with a disk AND money will be answered AS SOON AS POSSIBLE...

And please, state your name and address clearly in print because sometimes they are nearly impossible to decipher.

I will also try to put the latest available version of BootX on the following BBS (this may change without further notice):

Great Balls Of Fire BBS

Herman Stevens

Lubbeek

ADS Regional Coordinator Belgium

Tel. +32-(0)16-64.09.12

14400 BPS (N.8.1) v32bis v42bis

24-24 h.

Fido Node 2:292/603

NOTE

As of version 1.74 of the Recog file and BootX 5.23 a requester will appear when BootX thinks itself is older than 90 days or when the Recog file you are using is older than 60 days. This requester is just to warn you that it may be time to upgrade to a newer version.

1.23   Before you complain...

I just want to mention a couple of things here that will only
show up if you're snooping around in the system when BootX is
running.
Some of you seem to be running the Commodore development tool
Enforcer all the time. I have gotten some letters about the
Enforcer hits BootX causes. The only way I can check your
memory and vectors is to do some low memory reads. These will
typically show up as an Enforcer READ hit to addresses in the
region 0x64 to 0x74. This is normal. You can expect this to
happen at various points during program execution and
particularly during a vector check. Enforcer WRITE hits are
not normal. Please report these !
NOTE
If you haven't the faintest idea what I am talking
about, don't worry. I should only concern those people
who know what Enforcer does and how to use it.
BootX also launches a task called 'SnoopDos'. This is done to
prevent some old PowerPacker trojan horse to install itself in
memory. This task uses no CPU time. It is removed when BootX
quits running. Also, this task will only be launched when there
isn't yet a task called 'SnoopDos' in the system. This ghost task
won't prevent the real SnoopDos (by Eddy Carroll, Freely
distributable) from being run or functioning correct.

1.24   Acknowledgments

· Nico François - For his ReqTools and PowerPacker packages.
Also for answering my questions on programming in the new
KickStart v2.04 environment and for ßeta testing BootX.
· BootX ßeta testers - In no particular order:
Nico François (Belgium)
Johan Van Houtven (Belgium)
Bart Beeckmans (Belgium)
Steffen Salzmann (BRD)
James Knittle (USA)
Erik Løvendahl Sørensen (Denmark)
· Lieven Lema (Belgium) - For his translation into French of the

BootX locale file.

· Patrick Nordvall (Sweden) - For his translation into Swedish of
the BootX locale file.

· Steffen Salzmann (BRD) - For his translation into German of the
BootX locale file.

· Mikkel Bille Stegmann (Denmark) - For his translation into
Danish of the BootX locale file.

· Michele Giorato (Italy) - For his translation into Italian of the
BootX locale file.

· Tor W. Austerslått (Norway) - For his translation into
Norwegian of the BootX locale file.

· 'Big A' (my A3000) for assembling BootX fast and safe.

BootX has been created using the Macro68 Assembler, WShell 2.0
and CygnusEd Professional on an Amiga A3000 25/100 with 6MB
memory. The BootX documentation was formatted with my Roff
clone ARoff.

The assembler source code occupies more than 742.400 bytes and
consists of more than 34.500 lines.

1.25   Quit

When you click on the first gadget on the left side of the screen,
BootX quits running.
Before BootX quits it will check to see if you have made any
changes to the BBLib file or to the Brain file . If so, it will
ask you first if you want to save the changes you have made.
NOTE
Watch out that you don't leave any infected disks in
your drives when you quit BootX ! Especially disks
that are infected with disk-validator viruses.

1.26   Read Bootblock

This gadget explicitly instructs BootX to read the bootblock from
the current selected drive into the buffer. Normally BootX
detects when a new disk is inserted and automatically reads its
bootblock. But some old 5¼ drives don't support this automatic
diskchange. This gadget can also be used to read the bootblock
after another program has modified the bootblock.

You will also need to use this gadget when you have turned of the auto-diskchange detection or when you have locked a drive (see Lock Drive .

When BootX detects a virus in the bootblock it just read, a requester will be brought up asking you if you would like to kill the virus.

If you answer positively by clicking on the Yes gadget, BootX will try to install the bootblock current selected from the Bootblock library.

1.27   Write Bootblock

Here you instruct BootX to write the bootblock currently in the buffer to the selected drive. BootX maintains a buffer that holds the last read bootblock. This buffer is also visible in the large display window in semi-ASCII form (more about this later). The buffer allows you to read a bootblock from one disk and then copy it to another.

1.28   The drive cycle gadget

You can connect up to 4 disk drives to the Amiga. BootX checks how many drives are connected to the Amiga and installs their device names in the list of this cycle gadget .

Only the drives that are connected to the Amiga and are turned on (in case of external drives) will be reported by this gadget. To select a particular drive keep clicking on the gadget until its device name is shown.

1.29   Lock Drive

Normally when you insert a disk in the disk-drive, the Amiga tries to see if the checksum of the disk is OK. When it is not, the Amiga tries to validate it. Pre-2.0 KickStarts needed to load and execute a program called Disk-Validator from the L directory to check and/or repair the checksum.

When a disk-drive is 'locked', BootX will prevent AmigaDOS from validating your disk. This also means that Disk-Validator viruses have no chance of getting in. You can insert and remove disks as many times as you like.

Also the automatic diskchange detect will be temporarily disabled.

1.30   Check Files

This gadget is a short-cut for the Check files menu item.

1.31   Show installable bootblocks

When you click on this gadget, BootX will display the contents of
the current loaded bootblock library . You simply click on the
name of the bootblock to select it as current bootblock. Either
double-click on the name or click and select OK.
This action only selects the bootblock ! It is not immediately
written to the current disk nor is it put in the bootblock buffer.
You can see the name of the current selected bootblock in the
small window next to the gadget.
If you do not want to make a change select the Cancel gadget of
the requester .
NOTE
BootX already knows 2 bootblocks even when no
bootblock library has been loaded. They are the
standard AmigaDOS 2.0 bootblocks for OldFileSystem
and FastFileSystem disks. Their names have a
little dot completely to the right.

1.32   Built-in bootblock name

This window shows the name of the selected bootblock .

1.33   Install

Here you can write the selected bootblock to the bootsectors of
the disk in the current drive.
At startup BootX looks for a file called "BootX.BBLib" in the
directory from which BootX started. It contains several
alternative bootblocks. It is possible to load another bootblock
library using the Load menu item of the "BBLib" menu.

1.34 Load Recog file

As of BootX v4.30 all the recognition information BootX needs to

identify a virus or a bootblock is contained in a file called

BootX.Recog. With this menu item you can load different

BootX.Recog files than the one BootX loads at started called

"BootX.Recog".

NOTE

The BootX.Recog files are coded and protected by a

checksum to prevent anyone from modifying the

information it contains with a file editor.

Always try to get the latest version of the BootX.Recog file to

get maximum protection. You can find out the version of a

BootX.Recog file in several ways:

1. After BootX has successfully loaded a Recog file it will display

the version and the release date of the file.

2. The BootX version and the Recog file version are always shown

in the title bar of the screen.

3. Select the About menu item. The text will show the version and

the release date of the Recog file and how many bootblocks,

bootviruses and fileviruses it knows.

NOTE

These numbers may vary from the number reported by the

"Show known xxx" menu items. The numbers reported

here are the number of recog blocks that BootX can use

to identify a virus. The "Show known xxx" menu items

only report all the different names that BootX uses,

f.e. BootX can recognize several different mutants of

the same virus but the name will only show up once in

the "Show known bootviruses" list.

3. Use the AmigaDOS v2.x version command:

1> version BootX.Recog

1.35 Show known

This submenu allows you to view the contents of the current

loaded Recog file. Use the submenu items Bootblocks,

Bootviruses or Fileviruses to view the list you want. BootX

will then pop up a list requester. Use the scroll gadgets or

the scroll bar to move through the list. The window title of

the requester will tell you how many entries are being shown.

NOTE

The selection 'Show known fileviruses' also shows the

linkviruses and trojan horses that BootX knows.

Click on the window's close gadget to remove the requester.

1.36   Print known

Here you can request BootX to print the contents of the current

loaded Recog file to your printer.

NOTE

If you have to wait a long time (about 10 seconds) and

nothing happens, please check that your printer is

turned on and if so that it is on-line. BootX will

not be able to report an error until the default

time-out period is passed (usually 30 seconds) or

until the user put the printer on-line.

1.37   Iconify

Selecting Iconify causes BootX to close its screen and release as

much memory as possible. BootX will then try to find its icon in

the drawer from which it started and use it to create an AppIcon

on the Workbench. For more information about AppIcons I refer to

the Amiga User Interface Style Guide.

There are two ways to wake up BootX. You press the BootX hot key

combination (default Ctrl Left-Alt B) or you wake BootX up via its

AppIcon. The hot key combination can be changed via the menu item

Preferences/Edit .

When iconified BootX waits for two kinds of events. If you

double-click on the icon BootX reopens its screen and returns to

the state as it was before you iconified it with all the

information and preferences intact.

Another way to wake up BootX is to select or shift-select one or

more disk or drawer icons and drop them on the BootX AppIcon.

BootX will then wake up and present its Check Files Preference

requester. Set your preferences and when you click on

the Check Files gadget BootX will present you with the path

requester. BootX will have inserted the volume name or the drawer
name in the string gadget. You only have to click on OK or press
return to start the file check with that path.
If you drop a file icon on the AppIcon BootX will use that path of
that file as its starting point for the file check. This way you
can drop the icons you have 'left out' using the Leave Out
menu item of the Workbench Icon menu on the BootX AppIcon.
At present BootX uses no CPU time at all when it is in its
iconified state. No checking of memory or of newly inserted disks
is done. There is no need to continuously keep checking every
disk that is inserted. Just check them once to make sure they are
virus-free.
NOTE
If BootX can't find its icon it will use the default
tool icon instead. You can create your own default
tool icon using the Workbench 2.0 Icon Editor and save
it using the 'Save As Default Icon' menu item.
AppIcons are always positioned dynamically on the
Workbench screen, that is, whereever there is room.
You can not snapshot an AppIcon in a fixed place.

1.38   Help

As of BootX v5.00, the AmigaGuide library is used to implement the
help system. At the time of writing this library has only been
distributed by Commodore to registered developers. In the mean time
I might write a simply emulation for those users who do not have
AmigaGuide yet.

1.39   About

Here you can find out some details about BootX, among them the
version number, the Recog file version, the author of the language
translation and my address. Always try to get the latest version
available to have optimum virus protection.

1.40   Quit

Guess what this one does...

1.41 Check blocks

Some of the current viruses and possibly some to come alter or
damage the blocks on the surface of your disks. When you ask
BootX to check the blocks of the current drive it will scan the
complete surface of the disk that is currently in the drive,
looking for damaged or infected blocks.
When BootX encounters an infected block it will tell you together
with the name of the virus that infected that block. Most of the
time the information that was in an infected block is lost but in
some cases the block can be repaired and the information restored.
Currently BootX can repair blocks that are infected by the Saddam
disk-validator virus and by the Curse Of Little Sven virus.
It will tell you when it finds blocks infected by the Lamer
Exterminator virus.
To use the block check simply select the menu item. Make sure
there is no disk in the current drive. BootX will then lock the
drive and ask you to insert the disk you wish to check.
During the scan you'll get a progress report. You can leave your
disk write-protected. BootX will then issue a warning and it will
scan the disk without making any modifications.
NOTE
Be sure you insert your disk only after BootX asks for
it. This is necessary security and to make sure BootX
has a chance to recognize if your disk is a 880K or
1760K high-density disk.

1.42 Check memory

BootX checks the memory at strategic times during its actions.
With this memory action you can force BootX to recheck the memory.
When it finds a virus you will be alert with an Amiga DisplayAlert
and BootX will try to kill the virus in memory.
It is possible that the Amiga crashes after BootX has found a
virus. BootX will try to restore any changed information but
sometimes this is not possible. Therefore the safest solution has
been chosen of completely clearing the virus memory.

1.43   Check vectors

The flexibility of the Amiga operating system lies in the fact the
programmers have so many options of altering the way the Amiga
does specific things.
Regrettably virus programmers also know this and use various
vectors , as the entry points are called, to add the virus code
to the operating system. BootX can identify and kill virus if it
knows it. But new viruses are being programmed every day.
The vector check scans all the places in the Amiga memory that are
the most likely candidates for virus attacks. During this scan it
generate a vector check report. In this report is a lot of
technical information. After the scan BootX displays the report
in the main window. You can skip through the pages using the left
mouse button. The right mouse button aborts at any time. At the
end of the report BootX shows some information about your system.
If you need more accurate information use one of the many
dedicated tools.
When BootX finds a vector which points to abnormal address it will
try to identify the program that changed that vector. A lot of
programs 'patch' the operating system to fix bugs or to add
features. Commodore itself fixes small bugs this way using the
SetPatch command. I will try to keep BootX's knowledge base as up
to date as possible.
A normal vector will be displayed in black with the message 'OK'
next to it or the name of the program that changed that vector.
If BootX can not decide what code uses the vector it will display
the message 'Please check' in white next to the vector. This does
not necessarily mean that the vector is used by a virus. Don't
panic and contact a more experienced Amiga user (preferably a
programmer).
In case of doubt, save all your data and cold start the Amiga,
that is, turn of the power and wait a few seconds before turning
it on again.

1.44   Reset vectors

Here you are given to option to clear the first five vectors that
BootX shows in its vector check report. These vectors are used to
have foreign code survive a software reset (using the Ctrl - Left
Amiga - Right Amiga key combination).
This is not a fail safe way to kill a virus. Also if any other
software is using these vectors, it will not survive the next
reset. Also, some virus defend themselves against such action and
will not allow BootX to reset those vectors.

1.45   Check files

BootX was written in the days that bootviruses were the greatest
if not only treat to the Amiga, hence the name BootX. Over the
years this has changed. The biggest danger now comes from file
and linkviruses. This menu option provides you with the most
powerful tool in the Amiga world to fight and kill those viruses.
After you select the Check files menu item or click on the Check
files gadget BootX will present you with a Check Files Preferences
requester. BootX gives you a lot of options to customize your file
checks.
To remove the requester without checking your files click on the
window close gadget or the Cancel gadget.

1 Virus handling

2 Remove linkviruses

3 Handle viruses automatically

4 Show directories/files/checked files

5 Check crunched executables

6 Skip read-protected

7 Auto-remove read-protection

8 Generate report

9 Silent check

10 Filter

11 Examine directories

12 Checking files

1.46   Virus handling

In the top left corner you will find a cycle gadget that gives
the following options: Comment, Delete and Rename. It lets you
decide what BootX should do when it encounters a file virus.
The Comment option will let you add a filenote to the infected
file telling you the name of the virus. For example "BGS-9 (TTV1)
filevirus (detected by BootX)" tells you that this file contains
the BGS-9 filevirus and that it was detected by BootX. You can
see the filenote using most of the directory utilities of using
the AmigaDOS 'List' command in the CLI or the Shell.
The Delete option deletes the infected file.
The Rename option renames the infected file to a name describing
the virus. For example "002.BGS-9 virus" indicates that this file
contains the BGS-9 virus and it was the 2nd virus found during
this file check.

1.47   Remove linkviruses

When this check box is checked, BootX will try to remove a
linkvirus from an infected file. If the removal is successful the
file can be used again without further danger.
Be sure that enough memory is available during a file check as
BootX needs to load the infected file completely into memory in
order to repair it. Also there has to be enough disk spaces left.
When the virus has been removed, you will find two files on your
disk. One contains the original infected file. It will have a
new name reflecting the kind of linkvirus that is attached to the
file in the same way as the Rename option. The file
will not be executable anymore so there is no danger you
accidentally infect your memory. The virus code is still intact
though so don't panic if another viruskiller reports there still
is a virus on your disk.
The second file retains the original name and contains the cured
program.
NOTE
Some of the older linkviruses and some of the new
badly programmed linkviruses destroy the file in such
a way that repair becomes impossible. BootX handles
these cases gracefully and will tell you about it.

1.48   Handle viruses automatically

Normally BootX works completely interactively and asks the user's
permission every step of the way. When you check this option BootX
makes all decisions by itself and handles all viruses
automatically. This is very handy when you want to check that
400MB hard disk with 15000+ files on it.

1.49   Show directories/files/checked files

In the upper right hand corner you will find three options
controlling the display during the file check.
Check 'Show directories' if you want BootX to tell you when it
enters a new directory.
Check 'Show files' if you want BootX to show you the file it is
currently checking. When this option is on BootX also tries to
determine what kind of file it is looking at. Turn it off to
reduce the time needed to check a path.
Check 'Show checked files' if you want BootX to display only those
files that it really is checking.

1.50   Check crunched executables

BootX was the first viruskiller to offer you this powerful
feature. When it is checked BootX will try to decrunch all
crunched executables in memory and then check the decrunched
version of the file for any file- or linkviruses.
Currently BootX can decrunch the following crunchers:
· all PowerPacker versions
· all Imploder versions
· TNM Cruncher 1.1
· MegaCruncher
The Explode.library v6.0 or higher is required if you wish to
decrunch Library Imploded files.
If there is a cruncher you would BootX like to be able to
decrunch, then send me some files that have been crunched with it
together with the name of the cruncher and preferably the cruncher
program itself too. I will continue to add new decrunch code in
future versions of BootX.

NOTE

If BootX detects a linkvirus in a crunched file,

obviously it can not remove the virus since the file

is crunched. BootX will tell you to decrunch the file

first and ask you if you would like in the mean time

to handle the virus like a file virus, that is, to

comment, delete or rename the file (see Virus handling).

1.51   Skip read-protected

Sometimes you want to protect certain files on your disks from

reading by removing the read protection bit .

When this option is selected, BootX will ignore all read protected

files. This can be handy since BootX removes all protection bits

from infected files except for the Archive and the Delete bit.

NOTE

This option is mutually exclusive with the

'Auto-remove read-protection' option

1.52   Auto-remove read-protection

AmigaDOS 2.0 recognizes and honors the protection bits of a

file. That means that if the read protection is on, programs

including the Workbench and the Shell are not allowed to read that

file. When BootX encounters such a file it will ask you if you

want to remove the read protection of that file.

If you answer Yes BootX preserves all other protection bits and

removes the read protection. The file will then be rechecked.

If you wish to restore the read protection afterwards, use the

AmigaDOS command Protect like this:

1> protect <filename> -r

which will give the file its read protection back.

NOTE

This option is mutually exclusive with the 'Skip read

protected' option.

1.53   Generate report

Use this check box to turn on the BootX reporting feature. When
checked BootX will generate a file check report containing all the
information that was shown during the file check. Use this option
when doing an unattended file check of large hard disks. The
report will serve as a transcript of what happened during the file
check.

1.54   Silent check

When performing a file check in the background there is no need
for BootX continuously displaying all the files it checks. This
setting allows you to turn off this display while you can still
have BootX make a file check report.

1.55   Filter

To enable the path filter , check this checkbox gadget.

1.56   Examine directories

This cycle gadget allows you to decide how BootX handles
directories.
'All' is the most general option and will have BootX enter all
directories it encounters as many levels deep as required.
'Interactively' will have BootX ask you permission to check a
directory every time it encounters one. Check this option if you
want to make a quick scan of the first level of a directory for
example the root of a hard disk.
'None' will cause BootX to skip any directory it finds.

1.57   Checking files

You can start the file check by clicking on the 'OK' gadget. BootX
will pop up a path requester. This is kind of a stripped down
version of a file requester allowing you to comfortably navigate
your hard disk while looking for the path you wish to check.
The current path will always be displayed in the string gadget.
You can enter a path manually or you can use the parent and

directory buttons to look for the path.

Enter an empty string in the string gadget or click on Cancel to

call off the file check. For more information on how to use the

path requester I refer to the ReqTools documentation.

Double-click on a directory entry or select the OK gadget to begin

the file check. During the file check you can press the right

mouse button to pause the file check. Press the left mouse button to

stop the file check.

After the file check BootX will report how many directories,

files and crunched files it found. It will also tell you how many

files were checked and how many viruses were found.

If you asked BootX to generate a file check report you will be

given to option to look at it, print it, save it or ignore it.

More information about the print and save options can be found in

the following sections.

Viewing the file check report is the same as viewing a vector

check report: click the left mouse button the show the next page,

press the right mouse button to abort the viewing.

1.58   Show report

As mentioned in the section about Check files BootX is able to

generate a file check report while it's looking for viruses.

This menu item allows take another look at the last generated file

check report.

1.59   Save report

You can also save the last generated file check report to disk for

further reference. After you select this option BootX pops up the

ReqTools file requester (more information about this requester can

be found in the ReqTools documentation). As a default it suggests

the filename "LVCReport.doc" but you are free to use any name you

like.

The file check report is saved as a normal ASCII text file with

some standard ANSI codes to enhance the output. Most recent

document viewers such as PPMore can handle these ANSI codes.

1.60   Print report

You can print the last generated file check report from within
BootX using this menu item.
NOTE
If you have to wait a long time (about 10 seconds) and
nothing happens, please check that your printer is
turned on and if so that it is on-line. BootX will
not be able to report an error until the default
time-out period is passed (usually 30 seconds) or
until the user put the printer on-line.

1.61   Check disk

When you select this menu item, BootX will put up a requester
that allows you to specify which actions BootX should perform
when performing a complete disk check.
· Check bootblock: checks the bootblock
· Check blocks : checks all the disk blocks
· Check files : checks all the files
All these actions will be performed on the disk in the current
drive. Select OK to begin with the disk check. Click on the window
close gadget to cancel the disk check.

1.62   Load/Save

BootX can save the bootblock in the buffer as a file to any drive.
'Save' can be used to save the bootblocks of games, mega demos and
other programs to a separate data disk.
In case of a virus attack or a bad disk you can load the
previously saved bootblock back in the buffer and write it back to
the original disk.
'Load' will only allow you to load a file that contains a valid
bootblock. It checks if it is an OldFileSystem or FastFileSystem
bootblock and then recalculates the checksum of the bootblock.
When the bootblock checksum is wrong, BootX will ask you if you
want to repair it. If you select Yes, BootX repairs the bootblock
checksum of the loaded copy.
If the loading was successful BootX will try to identify the
bootblock. If the file you loaded did not contain a valid
bootblock, BootX will tell you.

1.63   Learn

This menu item is turned on when BootX does not recognize the
bootblock in its buffer. The bootblock can either be read from
disk (using the Read Bootblock gadget) or be loaded using the
Bootblocks/Load menu item.
If you want to teach an unrecognized bootblock to BootX use this
menu item. BootX will pop up a list requester showing the contents
of the current Brain file. Simply enter the name for the unknown
bootblock in the string gadget under the list. If the name you
entered already exists BootX will ask you if you'd like to use the
name anyway. This practise is not encouraged. Try to make your
names unique. It'll help you identify a bootblock much easier
later on.
After BootX has learned the bootblock it will display its name in
the message window in white. This way you know that it is the
Brain file that recognized the bootblock and not the Recog file.
NOTE
Only the Brain file can be updated by a user. The
Recog files will be periodically released by me. The
Recog file always takes precedence over the Brain
file. That is, if you teach BootX a bootblock as "New
Name" and some time later it is identified as "Some
new virus" you know that I have added the bootblock to
the Recog file. Please send me any bootblock that
BootX does not recognize yet. This way the Recog file
will always be up to date.

1.64   Show again

Use this menu item to refresh the display after it has been erased
or been used by some other BootX function.

1.65   New

When you select the BBLib/New menu item the current Bootblock
library in memory (if any) will be cleared so that you can started
a new one.
NOTE

BootX always adds four built-in bootblocks when you
start a new BootBlock library. These bootblock are
the standard OldFileSystem and the standard
FastFileSystem bootblock (normal and international
version) as they would be installed by the Install
command from the C directory. These two bootblocks
will not be saved with your Bootblock library. You
can tell which bootblocks are built-in by the little
dot that appears in the bootblock list. More
bootblocks will be added when the need arises.

1.66   Load

With the Load menu item you can load any Bootblock library. You
could for instance have a separate bootblock library for game,
demo or utility bootblocks.

1.67   Save

The Save menu item allows you to save the current Bootblock
library. This menu item will only be turned on when the Bootblock
library was actually changed.

1.68   Show

You can take a look at the contents of the current Bootblock
library by selecting the BBLib/Show menu item. BootX will pop up a
list requester showing you the names of the bootblocks.

1.69   Merge

You can share your Bootblock libraries with your friends. The
Merge menu item allows you to merge the contents of a Bootblock
library with the one in memory.
BootX will scan the complete library, ignoring duplicates and
adding any new bootblock not yet found in memory.

1.70   Add bootblock

Use this menu item to add the bootblock currently in the bootblock
buffer to the Bootblock library. BootX will pop up a list
requester with a string gadget. Enter the name of the bootblock
in the string gadget and click on OK (or press return).
BootX checks to see if that name already exists and if it does it
will ask you if you really want to add the bootblock.

1.71   Delete bootblock

You can use this menu item to delete a bootblock from the
Bootblock library.
NOTE
You can not delete a built-in bootblock. BootX will
warn you if you try to do so.

1.72   Rename bootblock

You can use this menu item to rename a bootblock in the
Bootblock library. Select the one you'd like to rename by clicking
on its name and then clicking on the OK gadget (or by double
clicking the name)
BootX will then ask you to enter the new name of the bootblock.
NOTE
You can not rename a built-in bootblock. BootX will
warn you if you try to do so.

1.73   New

When you select the Brain/New menu item the current Brainfile in
memory (if any) will be cleared so that you can started a new one.

1.74   Load

With the Load menu item you can load any Brainfile.

1.75   Save

The Save menu item allows you to save the current Brainfile. This
menu item will only be turned on when the Brainfile has been
changed during the run of the program.

1.76 Show

You can take a look at the contents of the current Brainfile by
selecting the Brain/Show menu item. BootX will pop up a list
requester showing you the names of the bootblocks BootX recognizes
in addition to the Recog file.

1.77 Merge

You can share your Brainfiles with your friends. The Merge menu
item allows you to merge the contents of a Brainfile with the one
in memory.
BootX will scan the complete Brainfile, ignoring duplicates and
adding any new bootblock not yet found in memory.

1.78 Delete Brain cell

You can use this menu item to delete a bootblock from the current
Brainfile.

1.79 Rename Brain cell

You can use this menu item to rename a bootblock in the current
Brainfile. Select the one you'd like to rename by clicking on its
name and then clicking on the OK gadget (or by double-clicking the
name)
BootX will then ask you to enter the new name of the bootblock.

1.80 Edit...

Use Preferences/Edit to change some settings of BootX. As of
version 5.10 all your settings are being kept in a file called
'BootX.prefs' which can be found in the directory 'BootX' of device
ENV: and ENVARC:.
The file is a plain ASCII file which can be read using any text
reader or modified using any text editor. Be carefull though
because the parsing routines are not very smart and presume a
couple of things. It is better to change your settings using BootX
and to save them.
1 Virus alarm

1.81   Virus alarm

BootX can grab your attention when it finds a virus by using one
of the following three possibilities:

(1) None: BootX will remain silent when it finds a virus.

(2) System beep: BootX will DisplayBeep the BootX screen. If you
have installed a sample using the Sound preferences editor this is
what you'll hear. Otherwise the colors of the BootX screen will
be flashed.

(3) Play sample: BootX will play the sample found in the file
BootX.Alarm. This file is loaded at startup time (when found).
The file must use the IFF 8SVX file format. Most if not all
sample creating programs support this file format.

1.82   Save icons

Check this settings if you'd like BootX to create an icon for any
of the files it saves.

1.83   Detect diskchange

If this option is turned on, BootX will know when you insert a
disk in any of the available disk drives. That drive will be made
current and BootX will try to read the bootblock of the disk you
just inserted.
To make BootX read a bootblock of a disk when this settings is
turned off, make the drive current that contains the disk you'd
like to check and click on the Read Bootblock gadget.

1.84   Check disk-validator

The disk-validator used to be a part of AmigaDOS 1.2/1.3 that
could be found as a file called Disk-Validator in the L drawer of
a boot disk.
There are some viruses that make use of the feature to infect your
computer and disks. The disk-validator would be loaded
automatically when AmigaDOS detected a non validated disk. The
disk-validator would then be put to work to repair the disk.
When this option is turned on BootX will check if such a file
exists on your disk after it has tested the bootblock. BootX will
check if it has been infected before a possible virus gets a
chance to get in your system. BootX does this by not using
AmigaDOS but directly manipulating the disk.
To have the most benefit of this feature, also use the Lock Drive
gadget. Read also How to kill disk-validator viruses... .
NOTE
KickStart 2.0 does not need the disk-validator anymore
since the repair code has been built in. This also
means that Amigas running KickStart 2.0 or later can
not be infected by inserting a disk that contains one
of those 'disk-validator viruses'.

1.85   Crunch

BootX crunch the Bootblock library and the Brain file before
saving them. It uses the powerpacker.library (© Nico François) to
crunch the files.
The 'Never' setting will never crunch any files.
The 'Ask' setting will cause BootX to ask you every time you save
the Bootblock library or the Brain file if you'd like BootX to
crunch them.
The 'Always' setting will ask no questions and will always crunch
before saving.
NOTE
Sometimes BootX will not be able to crunch your files.
BootX will typically report this as a buffer overflow.
This is because your data files have to much entropy.
This is neither a bug in BootX nor a bug in the
powerpacker.library. In this case you will have to
turn off crunching and save the file uncrunched.

1.86   Iconify at startup

When this setting is on (checked) BootX will start running in an
iconified state. See Iconify for more details.

1.87   Check memory at startup

If this setting is checked BootX will check the memory of the
Amiga for viruses every time it gets started or after it returns
from an iconified state. It is a good idea to have this setting
turned on.
You can always force BootX to recheck your memory using the
Check/Check memory menu item.

1.88   Check vectors at startup

This option works the same as the 'Check memory at startup'
option. Turn it on if you'd like BootX perform a complete vector
check every time you start BootX.
NOTE
In order to let you read any error message that are
shown at startup time (for example if something goes
wrong when trying to load the Recog file) BootX will
look at your settings right before starting the vector
check. If you have turned on the Pause after page
setting, BootX will ask you to click the left mouse
button before it shows you the vector check report.

1.89   Pause after page

Turning this setting on causes BootX to pause the screen output
every time a full screen page has been shown. See also the note in
Check vectors at startup .

1.90   Wrap words

Sometimes the output of BootX is longer than will fit in the
window. With this option you control wether BootX cuts off the
output or wraps the remaining part on the next line.

1.91   Display Mode

BootX can use any screen mode that the Monitors database offers to
you. If you click this gadget BootX will put up the ReqTools
display requester allowing you to choose which screen mode will be
used.
If you select a new display mode your choice will become active
immediately.

1.92   Palette

Clicking this gadget makes BootX put up the ReqTools palette
requester. You can use it to customize the BootX screen colors.

1.93   Load/Save

Use these menu items to load or save the current settings from or
to the BootX.prefs file.
After a succesfull load BootX will close and reopen its screen
even if you did not change the display settings.

1.94   How to kill disk-validator viruses...

First make sure that your memory is clear of any virus. You can
do this by turning of your Amiga for a few seconds and then
booting from a NON-INFECTED disk. Always make sure all the disks
you insert are WRITE-PROTECTED unless it contains a disk-validator
virus you want to kill.
Start BootX and lock the current drive by clicking on the Lock
gadget next to the drives cycle gadget. This prevents AmigaDOS
and the Workbench from accessing the disks you are about to
insert.
Insert the infected disk into the current drive and click on the
Read gadget. BootX now reads the bootblock and checks the file
called 'disk-validator'. This file can be found in the L drawer
of the disk.
If BootX does not check the disk-validator, make sure that the
'Check Disk-Validator' check box is checked in the Preferences
requester.

When BootX finds a disk-validator virus, it will ask you if it
should kill the virus. If you select Yes, BootX destroys the
complete virus code and makes a harmless executable program of it
that does nothing than set the dos error code to -1.
Next you are given the advice to repair the disk using the 'Check
blocks' menu item.
As a last step you need to copy an UNINFECTED disk-validator file
into the L drawer of the repaired disk. This step could not be
automated because the Disk-Validator file is copyrighted by
Commodore Amiga.

1.95   Using BootX with little memory

BootX has become quite a large program over the years and together
with its support files it can occupy over 200K of RAM.
Don't generate a file check report when checking your hard disk.
The file check report for hard disks with 5000+ files can grow
easily larger than 200K.
Clear the Brain and BBLib files using the 'BBLib New' and 'Brain
New' menu items respectively.
Delete or rename the file 'BootX.Alarm'. It contains the sample
BootX uses to alert the user if it finds a virus. Use the System
Beep option of the Preference requester instead.
Use the lowest possible screen mode the Display Mode requester has
to offer. Specify the screen dimension as low as possible.

1.96   Copying bootblocks with one drive

Turn off the Detect Diskchange setting in the Preference
requester.
Insert your disk in the disk drive.
Click on the Read Bootblock gadget. The bootblock will now be read
into the BootX buffer and will be shown in the main window.
Remove the disk from your disk drive and insert the disk you want
the bootblock copied to.
Click on the Write Bootblock gadget. BootX will now write its
buffer to the newly inserted disk.

1.97   About Safe Hex International

Our motto: "Safe Hex..."

BootX can be kept up to date thanks to the energy and work put
into a global anti-virus information bank called Safe Hex
International (SHI) founded by Erik Løvendahl Sørensen from
Denmark.

This group has over 250 international members now, among them
some of the programmers of well-known anti-virus programs like
Steve Tibbet and Jonathan Potter. Among the activities of this
group are:

- Spreading information to anti-virus programmers as fast as
possible.

- Trying to get names and proof against virus programmers and
giving the information to the justice department of his/her
country to press charges.

- Writing articles in over 8 popular magazines to inform new Amiga
users about viruses and how to protect themselves.

- Collecting viruses and virus information as well as the programs
to kill the viruses.

- Helping people to get their money back if it has been lost due
to virus infection.

- Releasing the newest and best viruskillers found.

- We have set up over 20 Regional Virus Centers worldwide. These
virus centers provide free help against viruses through a
hotline and they translate the newest viruskillers and their
documentation in their local language.

Also the organization is prepared to pay $ 1000 US for the
name and the address of a virus programmer. All information
will be delt with confidentially and will be handed over to the
proper authorities of the virus programmer's country. Data
crimes are punished severely most of the time with up to 5 years
imprisonment.

All this is volunteer work. If you want some more information
about this organization or you want to sponsor our work, contact
Erik at the following address:

SHI Safe Hex International

Erik Løvendahl Sørensen

Snaphanevej 10

4720 Præstø

Denmark - Europe

Phone: 00 45 55 99 25 12

Fax : 00 45 55 99 34 98

Fido : (Soon)

1.98   Addresses

Peter Stuer

Kauwlei 21

B-2550 Kontich

Belgium - Europe

Fido : 2:292/603.7

Great Balls Of Fire BBS

Herman Stevens

Lubbeek

ADS Regional Coordinator Belgium

Phone: +32-(0)16-64.09.12

14400 BPS (N.8.1) v32bis v42bis

24-24 h.

Fido : 2:292/603

SHI Safe Hex International

Erik Løvendahl Sørensen

Snaphanevej 10

4720 Præstø

Denmark - Europe

Phone: 00 45 55 99 25 12

Fax : 00 45 55 99 34 98

Fido : (Soon)

1.99   Using BootX in a different language

If you are using Workbench v2.1 BootX will try to open the

locale.library to make use of its language support. If BootX

finds a catalog in the language currently used in the system BootX

will use it.

Normally you set your preferred language using the Locale

preference editor of the Workbench. If BootX finds a tooltype in

its icon called LANGUAGE it will use this language to override the

system language. For example if you run your Workbench in English

but you want to use BootX in Dutch then add the following tooltype

to the BootX icon:

LANGUAGE=nederlands

NOTE

Make sure BootX can find the appropriate language

catalog either in the directory

'Locale:Catalogs/<language name>' or in the directory

'Catalog/<language name>' from which BootX started.

1.100   Glossary

Glossary

The glossary will help inexperienced users understand some

of the more technical terms found in this document.

A B C D E F G H I J K L M

N O P Q R S T U V W X Y Z

1.101   Glossary Part A

A

AmigaDOS

The disk operating system (DOS) used by Amiga computers.

ANSI

American National Standards Institute. A set of codes to perform

special functions like printing text in bold.

appicon

An icon on the Workbench that allows the user to pass arguments to

an application. BootX puts an AppIcon on the Workbench when you

select the Project/Iconify menu item.

application gadget

A graphics image that appears within a program which can be

manipulated with the mouse to perform a certain function.

ASCII

American Standard Code for Information Interchange. The code set

that the Amiga uses to represent letter and figures.

assembler

A programming language

attributes

A series of flags stored with every file. Attributes indicate file

type and control the file operations (read, write, delete, etc.)

Also protection bits .

1.102   Glossary Part B

B

BBLib

See 'bootblock library'.

boot

The startup process for a computer. It comes from the expression

"pulling yourself up by the bootstraps."

bootblock

Part of a disk or hard disk that is read first by the Amiga to

initialize or recognize the disk. Usually contains a small

program (disks) or parameters (hard disk).

bootblock library

A custom file of BootX that contain several bootblocks that the

user can install on his disks.

bootvirus

Virus that hides in the bootblock of a disk. Only gets activated

when you boot from the infected disk.

Brain file

Supplementary recognition information that the user has taught

BootX using the Bootblocks/Learn menu item.

bug

A mistake in a program

1.103   Glossary Part C

C

check box

A gadget used to let a user turn an option on or off. When a check

mark appears in the box, the selection is considered to be "on."

Chip RAM

The area of RAM accessible to the Amiga's custom chip set used for

graphics and sound data. Also called graphics memory.

CLI (Command Line Interface)

A means of communicating with the Amiga by issuing commands from

the keyboard.

click

To press and release a mouse button

close gadget

A system gadget used to close windows. It appears in the upper

left corner of the window.

cold reboot

To reset the Amiga by turning the power off, waiting 20 seconds,

and then turning it on again.

CPU

Central Processing Unit. The micro processor that executes the

instruction that make up a program. The Amiga either has a

MC68000, MC68020, MC68030 or MC68040 CPU.

Crunch

Transforming data in such a way that the result uses less memory

or disk space. BootX optionally uses the PowerPacker library to

crunch or decrunch the data it uses.

current drive

The drive that will be used by default for any BootX operation

unless the user specifies another.

cycle gadget

An application gadget that allows the user to select one of

several options. One option is displayed at a time and, as the

gadget is selected, the other options become visible. The

displayed option is the selected option.

1.104   Glossary Part D

D

default

Initial setting, what will be used if nothing else is supplied.

delete

To erase or discard a file, buffer, or other stored item.

device

A physical mechanism, such as a disk drive, used as a source or

destination for information

directory

A subdivision in a computer's filing system. Directories are

represented on the Workbench as drawer icons.

disk

A medium for storage of computer data.

disk-validator

An extension of the pre-2.0 AmigaDOS. It was loaded from disk

whenever the user inserted a bad or corrupted disk. The

disk-validator tried to repair the disk.

double-click

To press and release the mouse's selection button twice.

drawer

See 'directory'

drive name

The name assigned to a floppy disk drive or hard disk such as

Work:, WB2.x: or DF0:. It is the same as the device name.

1.105   Glossary Part E

E

executable

A type of file that contains program instructions that the Amiga

can use the execute a program. Executables are the prime target

for linkviruses.

1.106   Glossary Part F

F

FastFileSystem (FFS)

An enhanced Amiga file system usable with both floppy and hard

disks. A volume is formatted as either FFS or OldFileSystem (OFS).

The advantages of FFS are increased storage capacity and faster

access of files.

Fast RAM

General memory used by programs and data, as opposed to Chip RAM .

file

A collection of data

filename

The name by which a program or a data file is known to the Amiga

when you want to access it on disk or hard disk.

FPU

Floating Point Unit. Special purpose micro processor to speed up

calculations using real numbers. The FPU's available for the Amiga

are called MC68881 and MC68882.

1.107   Glossary Part G

G

gadget

Any graphics image that can be found in a window or on a screen

that can be manipulated with the mouse to perform a certain

function.

graphics memory

See Chip RAM

GUI

Graphical User Interface

1.108   Glossary Part H

H

hot key

A special key or key combination that triggers a certain action.

BootX uses a hot key to uniconify or to bring its screen to the

front.

1.109   Glossary Part I

I

icon

An image appearing on the screen to represent a disk, drawer,

project or tool. Icons can be moved and selected with the mouse to

allow you to work with the items they represent.

IFF 8SVX

A file format following the Interchange File Format (IFF)

specifications. It's purpose is to allow easy interchange of 8

bit sound samples.

1.110   Glossary Part K

K

K

Kilobyte. 1024 bytes.

KickStart

Software that is read from disk and used to boot the Amiga. Also

refers to the portion of the OS that is in ROM. BootX requires at

least version 2.04 of the KickStart.

1.111   Glossary Part L

L

Locale.library

A library available with Workbench 2.1 that allows programmers to

write language independent programs.

library

A related set of functions that can be shared by various programs.

BootX uses for example the PowerPacker.library to load and save

crunched data.

linkvirus

A virus that spreads itself by linking its own code to other

programs. When you run an infected program the virus hides itself

in memory before allowing the program to execute. There it waits

to infect other programs.

1.112   Glossary Part M

M

MB

Megabyte. 1,048,576 bytes

MMU

Memory Management Unit. Special purpose micro processor that can

efficiently control the memory of the Amiga. The MMU available for

the Amiga is the MC68851.

menu

A list of on-screen options, displayed by using the menu button,

from which users can choose commands that control a program.

menu button

The right mouse button

menu item

An option that appears in a menu

1.113   Glossary Part O

O

OldFileSystem (OFS)

The file system used by pre-2.0 AmigaDOS.

1.114   Glossary Part P

P

path

The series of device, directory and subdirectory names that

defines the location of a file.

PowerPacker.library

A shared library written by Nico François. It allows BootX to read

and write crunched files using the PowerPacker crunching

algorithm.

processor

See CPU

protection bits

See attributes

1.115   Glossary Part R

R

RAM

Random Access Memory. Part of the Amiga's internal memory that

can be used for data storage and is directly accessible by the

CPU. Data in RAM is lost when the Amiga is rebooted or powered

off.

read only

If a disk status or a file attribute is Read Only, the user can

only look at the contents of the disk or the file without

modifying it.

reboot

To reset the Amiga by pressing the keys Ctrl, Left-Amiga and

Right-Amiga simultaneously. Also called a warm boot.

Recog file

The file that contains all the information that BootX needs to

recognize and kill boot- and linkviruses.

ReqTools.library

A shared library written by Nico François. It has special

functions to create the several kinds of requesters that BootX

uses.

requester

A window that allows the user to control options, access files or

confirm actions.

ROM

Read Only Memory.

root directory

The main directory of a volume. All other directories are created

in the root. The word comes from the analogy between the structure

of the file system and the shape of a tree.

1.116   Glossary Part S

S

SCSI

Small Computer System Interface. A standard interface protocol for

connecting peripherals, usually mass storage devices like hard

disks, to computers.

selection button

The left mouse button.

subdirectory

A directory within another directory

1.117   Glossary Part T

T

Tool Types

A method for passing arguments from the Workbench to a program.

BootX uses the tool types to store the user preferences.

1.118   Glossary Part V

V

vector

A memory address used by the operating system to access parts and

functions of the libraries and devices. Most viruses change one or

more viruses to modify the behavior of the Amiga.

volume

A floppy disk or a hard disk partition

volume name

The name given to a disk or partition

1.119   Glossary Part W

W

Workbench

The Amiga's icon-based GUI .

write-enable

To allow information to be written onto a disk.

write-protect

To prevent information from being written onto a disk.

1.120   Glossary Error

No entries found for this letter.

1.121   Index

T