

\$VER: HappyENV 1.0 (05.06.97)

COLLABORATORS

	TITLE : \$VER: HappyENV 1.0 (05.06.97)		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		January 19, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	\$VER: HappyENV 1.0 (05.06.97)	1
1.1	HappyENV V1.0 Documentation	1
1.2	Introduction	1
1.3	About its Name	2
1.4	Copyright & Disclaimer	3
1.5	Installation	4
1.6	Startup Options	5
1.7	Technical Notes	6
1.8	Limitations	8
1.9	Programming Guidelines	8
1.10	History	10
1.11	Future	10
1.12	Credits	11
1.13	Reporting Bugs	12
1.14	Author	12

Chapter 1

\$VER: HappyENV 1.0 (05.06.97)

1.1 HappyENV V1.0 Documentation

HappyENV V1.0 Documentation
June 5th 1997

Introduction

About its Name

Copyright & Disclaimer

Installation

Startup Options

Technical Notes

Limitations

Programming Guidelines

History

Future

Credits

Author

HappyENV is CARDWARE

Please send a picture postcard to me if
you keep it installed on your system !

1.2 Introduction

This is a kind of RAM disk handler specialized for environment variables that are usually found in RAM:ENV. Because environment variables (short: ENV vars) are usually quite small and there are lots of them, HappyENV is optimized for lots of small files. This brings you the following advantages:

- * Smaller memory usage
- * Faster accessing of ENV vars
- * You can do "delete RAM:#? all" without hurting ENV vars
- * Conceptionally much cleaner than the RAM: concept

This works by providing you an AmigaDOS(TM) handler for a ENV: device.

And there is an additional feature:

Every time a ENV var is accessed and cannot be found in ENV:, HappyENV tries to copy it automatically from ENVARC:. This makes it possible to leave out copying the whole ENVARC: directory to ENV: during boottime. To avoid many disk accesses HappyENV remembers files that are not in ENVARC:. So the additional advantages are:

- * Faster booting
- * Even smaller memory usage
- * You do not have to clean up ENVARC:, because unused files in it will not be copied to ENV:

HappyENV should run on every Amiga(TM), please tell me when it does not work correctly.

1.3 About its Name

Everything calls itself "magic":

MagicWB, MagicLink, MagicED, MagicFile, MagicUserInterface, MagicShit, MagicLocker, MagicSelector, MagicC64, DesktopMagic, BlizzMagic, MagicWord, MagicTrash, MagicFileRequester, MagicColors, MagicCX, MagicMenu, MagicPtr, MagicClip, MaxonMagic, MagicClock, MagicEye, MagicFrames, MagicColors, ...

just to mention some of them, isn't this really annoying ???

And what is even disgusting: None of them is really magic and most of them are highly trivial !

If something names itself "magic", I wouldn't expect a miracle, but at least something which was generally thought of to be impossible to implement. If I can imagine how it is done by simply reading a short description of it, there can be no magic in it at all.

Now check the list above again and you will see that nothing is really magic, except maybe "MagicLocker". :-)

What are the reasons to call all programs "magic" ? I can imagine two possibilities:

- 1.) Just naming a program after what it does is annoying, too. There needs to be added some redundancy to make it "unique". ;-)
 - 2.) (I better do not say that here.)
-

So what has this to do with HappyENV ?

Well, I do not want to name it simply "ENV-Handler", so I thought about alternatives. The first idea was "MagicENV", but I forgot about it faster than it came, because of the above mentioned reasons.

The second idea was "HappyENV".

Surely this ENV-handler cannot be happy and your computer cannot be happy, too, but maybe it makes you happy. So there is more truth in the name than the "magic" programs have. And it isn't that annoying.

PS: If you are a author of a "magic" program, then please send your flames to NIL:. And do not take this too serious.

1.4 Copyright & Disclaimer

Copyright:

HappyENV is CARDWARE. This means that if you keep it installed on your system you are obliged to send me a picture postcard (NOT an EMail, but you can additionally send me EMail).

By sending me a postcard, you get a legal user of HappyENV and all future versions of it. Using HappyENV without sending me a postcard is both immoral and illegal.

To those internet freaks: A postcard gets sent via snail mail. :-)

Please send me a picture postcard ! (Yes, I am talking to YOU!)
I could have made HappyENV a crippled piece of Shareware, but I decided to distribute it this way. It is more comfortable and cheaper for you; and I do not have to spend time dealing with keyfiles.

The source code is included, but only for tutorial purposes. Do not include it or parts of it in other programs without my permission.
I (Martin Gierich) am the copyright owner.

MountENV and MountENV.asm are Public Domain, do with them what you want.

Distribution:

You are allowed to copy HappyENV as long as you do not charge more than the price for the medium (and maybe a REALLY SMALL copying fee).

You are allowed to put it on a collection of Public Domain/Freeware/Shareware programs, if you send me a free copy of it. (If you put it on a cover disk of a disk magazine, then please send me the corresponding issue of this magazine, too.)

Following files belong to the distribution:

HappyENV-Handler	- main program, to be copied to L:
HappyENV.guide	- documentation
MountENV	- starter, to be used in the startup-sequence
Source/HappyENV-Handler.asm	- assembler source code of main program
Source/HappyENV-Handler.debug	- for debugging purposes
Source/MountENV.asm	- assembler source code of starter

Source/debug.i	- include file used in source codes
Source/DosPackets	- overview of all known Dos-Packets
Source/Null.asm	- example source code of a simple handler

Disclaimer:

Use it at your own risk ! I can give you no warranty.
The source code is included, so you can watch out for bugs for yourself.
I am not responsible for anything that happens, especially if it is something bad. :-)

1.5 Installation

It is not possible to modify the startup-sequence directly from within an installer script so you have to install it manually. However I have added some "ACTION" buttons for your convenience.

If you want to check out how much memory/speed you gain:

Get yourself a stopwatch. Reboot your computer and measure the time booting takes. Write down the amount of free memory available directly after booting. Now install HappyENV (see below), do the rebooting procedure again and compare the results.

First copy HappyENV-Handler to L: (ACTION: copy HappyENV-Handler L:) and MountENV to C: (ACTION: copy MountENV C:) using a directory utility or shell. Then you need to modify your startup-sequence using a text editor. It probably looks something like this:

```
-----  
Setpatch QUIET  
Version >NIL:  
Failat 21  
MakeDir RAM:T RAM:Clipboards RAM:ENV RAM:ENV/Sys  
Copy ENVARC: RAM:ENV ALL QUIET NOREQ  
Resident C:Assign  
Assign ENV: RAM:ENV  
Assign T: RAM:T  
Assign CLIPS: RAM:Clipboards  
etc. etc.  
-----
```

For safety reasons backup your startup-sequence first

(ACTION: copy S:Startup-Sequence S:Startup-Sequence.old) .
Now make use of the text editor (ACTION: ed S:Startup-Sequence) and delete the "RAM:ENV" and the "RAM:ENV/Sys" part from the "Makedir ..." line. Then delete the "Copy ENVARC: ..." and the "Assign ENV: ..." line and insert "C:MountENV" where the "Assign ENV: ..." line was. After that your startup-sequence should look like this:

```
-----  
Setpatch QUIET  
Version >NIL:  
Failat 21  
MakeDir RAM:T RAM:Clipboards
```

```
Resident C:Assign
C:MountENV
Assign T: RAM:T
Assign CLIPS: RAM:Clipboards
etc. etc.
-----
```

If you really want to, you may add some startup options to MountENV. After booting again, HappyENV hopefully works fine.

Deinstallation can be performed by changing the startup-sequence back to its state before (ACTION: copy S:Startup-Sequence.old S:Startup-Sequence) and deleting "L:HappyENV-Handler" (ACTION: delete L:HappyENV-Handler) and "C:MountENV" (ACTION: delete C:MountENV) .

There is no need of a usage description, because it simply gets used by the programs you use.

1.6 Startup Options

MountENV allows the use of some startup options. Its template is:

```
DEVNAME/K,VOLNAME/K,ICON/S,NOREQ/S,NOCOPY/S
```

or some more common usage description:

```
MountENV [DEVNAME <name>] [VOLNAME <name>] [ICON] [NOREQ] [NOCOPY]
```

DEVNAME must be the first (!) option, since the others get passed to the HappyENV-Handler.

DEVNAME is the name of the AmigaDOS(TM) device to mount. Do not append ":". Default is "ENV".

VOLNAME is the name of the volume inserted into the above device. Do not append ":". Default is "Env" and if ICON is specified default is "Environment".

ICON switches off disk icon hiding on Workbench(TM).

NOREQ switches off displaying the unsupported packet warning requester.

NOCOPY switches off to copy files from ENVARC: automatically.

If ICON is not specified, device and volume name should be the same.

Do not use underscores in device and volume names.

Some notes on these topics can be found in Technical Notes .

Examples:

```
MountENV volname MyEnv icon
```

```
-> This will put a disk icon on your WB labelled "MyEnv"
```

```
MountENV devname ram2 volname "Ram Disk 2" icon nocopy
```

```
-> This will give you a second ram-disk (but optimized for small files)
```


1.7 Technical Notes

How HappyENV works:

HappyENV-Handler works quite similar to the ram-handler that can be found on OS1.3 Workbench(TM) disks or in the Kickstart ROM of the newer OS versions. They both have an an DOS packet interface to dos.library (the part of the operating system that provides an interface for application programs to all disk oriented activities) which is the same as the hard and floppy disk filesystem has.

HappyENV and Ram-Disk store their files in memory instead on disk, so they get lost if you switch your computer off. If you decide to keep Environment variables (short: ENV vars) in your Ram-Disk, they have to be copied from ENVARC: to there during booting. Assigning ENV: to ENVARC: is no great idea, because:

- 1.) Accessing there is slow and some ENV vars get accessed quite often.
- 2.) Some programs (example MUI) create lots of temporary ENV vars.
- 3.) The "Save" and "Use" buttons in preferences programs result to have the same effect.
- 4.) You get the last preferences settings instead of your preferred settings after booting.

As an solution HappyENV offers to copy files from ENVARC: on demand. If you check for the existence of an file or want to read the file, HappyENV searches ENVARC: for a file with the same name and relative path. On success it will be copied to ENV: and it is now a real file in ENV:, otherwise a hidden file of size zero will be created in ENV: that avoids searching ENVARC: for the same file again. To allow specifying paths (especially deep ones) HappyENV creates directories on the fly. So if you cannot delete a diretory that seems to be empty, it has some hidden files in it.

Ram-Disk has a block oriented design which is smart when having big files, but ineffective (slow and memory hungry) when using lots of small files. HappyENV however allocates only one block of memory for each file which is only 16 bytes plus length of filename plus size of its contents big. To avoid memory fragmentation it uses memory pools under OS3.0 or better with a puddle size of 5000 bytes.

In opposite to Ram-Disk notification support of HappyENV REALLY works, even with directories, and HappyENV can get write protected using the command "Lock". Since it is annoying to see an extra disk icon for ENV: on the Workbench(TM) I have made a hack that surpresses it: The volume node, though present, does not get linked into the DOS device list. Using startup option ICON you can switch this off.

Protection bits:

If a file was copied from ENVARC: and not changed meanwhile it has the A (archive) protection bit set. The P (pure) protection bit indicates that an application watches this file for changes using notification. R (read), W (write) and D (delete) are always set and H (hide/reserved), S (script) and E (executable) are always unset.

Alternative ways of mounting the HappyENV-Handler (usual way is using MountENV) :

* MCP (written by Stefan Sommerfeld)

The current beta version has the possibility to mount HappyENV. It checks for presence of an ENV: assign and for HappyENV-Handler residing in your L: directory and mounts it if possible. So all you have to do after installing HappyENV is to delete the "MountENV" line from your startup-sequence. Since I have not seen this version yet, I have not tested it.

* The good old Mount command

MountENV is equivalent to this Mountlist entry:

```
-----
ENV:
    Handler = L:HappyENV-Handler
    Stacksize = 2000
    Priority = 5
    GlobVec = -1
#
-----
```

If given, the startup options, except DEVNAME, are mapped to the "Startup" field. Since the Mount command dislikes spaces in the "Startup" field, you should replace them by underscores. HappyENV will complain if it does not like your options. So for example

```
MountENV DEVNAME Test VOLNAME "Ram Disk 2" ICON NOCOPY
```

is equivalent to:

```
-----
TEST:
    Handler = L:HappyENV-Handler
    Stacksize = 2000
    Priority = 5
    GlobVec = -1
    Startup = VOLNAME_"Ram_Disk_2"_ICON_NOCOPY
#
-----
```

Btw. the OS3.0+ Mount command pops up an annoying requester when trying to mount a device named ENV:, because Mount itself needs ENV: to be present.

The debug version:

There is a debug version for those who want to play around with it. It is named HappyENV-Handler.debug and is located in the Source directory so you can rename it to HappyENV-Handler or use the Mount command to make use of it. You need a debugging terminal or Sushi

(the program, not the fishy one :-) to see debugging output.
See `Reporting Bugs` for more.

1.8 Limitations

- * `Ixemul.library` does strange things. It scans `ENV:` to see what is in there. So it does not find `ENV` vars that are not already in there.
Solution: If this is a problem, then insert the line
`"C:copy ENVARC: ENV: NOREQ QUIET"` into the startup-sequence where the same line except for the `"ALL"` switch was.
- * `ASL` and `ReqTools` `filerequesters` do not like to display `ENV:` any more. This should not be really a problem, because you can enter `ENV:` manually in the directory string gadget. Regard it as a feature. Anyway, solution: Add the line `"Assign Environment: ENV:"` to the user-startup or use the `ICON` startup option or see below.
- * I saw a `CDROM` which does an `"assign env: mycd:env add"`. This obviously does not work. It even will not work with most applications.
Solution: Use `"MountENV DEVNAME Environment VOLNAME Environment"` instead of the `"MountENV"` line in your startup-sequence and add the line `"Assign ENV: Environment:"` below it. This even lets `ENV:` appear in `filerequesters`.
- * All files/dirs have the same date (date of start of HappyENV). Adding date stamps would swallow 12 bytes per file/dir. I cannot see any need for date support for `ENV` vars (except that directory utilities use it to update their directory list). Determining whether a file has been changed should take place using Notification. Since `IPrefs` relies on Notification support by `ENV:`, it is no problem, if your program relies on it, too.
- * Protection bits are unsupported, too. Again, I cannot see any use for them. There should be no executables or scripts in `ENV:` !
Since they eat up only one byte each file/dir it is quite probable that I add them (but maybe I force `S`, `P` and `E` to be always unset).
- * When trying `"DIR ENV: ALL"` you get a `"circular entry found"`. It seems like `DIR` interprets something into `fib_DiskKey`. This is crap !
Is `DIR` a directory lister or a disk checker ???
- * `SnoopDOS` does not like one handler to access another handler. This might cause deadlocks.
Solution: Turn off `"Monitor packets"` and `"Packet debugger"` in `SnoopDos` or use option `"NOCOPY"` with `MountENV`.

1.9 Programming Guidelines

General things:

Debugging your programs always should take place with `MungWall`, `Enforcer` (if you have a `MMU`) and `Poolwatch` (if you use `OS3.0+` mempools) running

in background. It seems that some people think that Enforcer substitutes MungWall. This is not true ! Without MungWall some Enforcer hits will not appear and MungWall hits are much worse than Enforcer hits ! These tools together with the needed Sushi and Segtracker can be found on Aminet, on Meeting Pearls CDROMs, on the Amiga Developer CDROM and on the NDUK disk set.

My debug script that gets executed early in the startup-sequence looks like this:

```
-----
SYS:Asm/Bug/SegTracker
run >NIL: SYS:Asm/Bug/enforcer RAWIO SHOWPC STACKCHECK LED=0 BUFFERSIZE=10000
run >NIL: SYS:Asm/Bug/mungwall NAMETAG SHOWFAIL SHOWSTACK SHOWPC SHOWHUNK
run >NIL: SYS:Asm/Bug/poolwatch
run >NIL: SYS:Asm/Bug/sushi <>CON:0/30/640/170/Sushi/AUTO/CLOSE/WAIT/INACTIVE ON ←
      BUFK=64 NOPROMPT ASKEXIT ASKSAVE
-----
```

Files in ENV: :

In the root dir of ENV: there should only be real global environment variables. They contain one line of text and they can luckily be viewed using "GetEnv" from CLI.

Prefs/Configuration files can be stored in ENV:, too, if you follow these guidelines (in my opinion):

- * The user needs to have a choice between "Save", "Use" and "Cancel". "Save" saves the pref/config file to ENVARC: and to ENV:, "Use" saves it only to ENV:. If your program allows only "Save" and "Cancel", PROGDIR: is a much better place to put your prefs/config files to.
- * They should not be too big, especially not bigger than 5KB.
- * They want to be stored in a separate subdirectory named after their application. This is to avoid a mess.
- * It is a good idea to append .prefs or .config to their name. So they look something like: ENV:AppName/AppName.prefs
- * If possible use an IFF compatible format and use iffparse.library to read/write them.
- * Additionally searching in PROGDIR: for them or allowing to set the path using ToolTypes would be nice.

Documentation:

Make sure that all (or at least nearly all) lines in your AmigaGuide(TM) documentation are less than 78 characters long (This means up to 77.) ! Otherwise users who use the default screenwidth of 640 pixels will have to scroll left and right once per line and will angrily give up after reading one page. Anyway having about 70 characters per line increases readability of the whole text.

1.10 History

V0.6 (13-Dec-95): First working version.

It worked on my A500 OS2.1, but it crashed under OS3.0. When I got my 68030 with MMU I tried to find this bug using Enforcer. It has not helped me, but I found this bug by chance: It was Workbench crashing (unbelievable ?) because ACTION_INHIBIT did nothing and returned DOSTRUE.

V0.7 (18-Jun-96): First release.

V0.8 (31-Aug-96): Lots of small bugs fixed.

- * ACTION_RENAME crashed. This caused MagicWB to crash.
- * ACTION_FH_FROM_LOCK for ACCESS_READ added.
- * Access to a dir now creates a dir, if this dir is present in ENVARC:
- * Copies files from ENVARC: instead of SYS:Prefs/Env-Archive
- * Read/Write returns Result2=0 on success (for compatibility)
- * id_InUse, fib_NumBlocks, id_NumSoftErrs are filled correctly
- * Multiple ENVARC: assign (ASSIGN ADD) supported
- * Improved this guide a lot
- * Fixed two bugs in subdirectory creation

V0.9 (27-Oct-96): Notification handling rearranged

- * No more annoying requesters with OS2.04
- * A bit faster and uses less memory
- * Protection bits reflect file status
- * Path name expansion added to source code only (pretty useless)
- * Major changes in this guide:
History, Future, Credits, Reporting Bugs

V1.0 (5-Jun-97, one year after first release): Much better packet support, now HappyENV is really worth the version number 1.0 .

Unbelievably that it took me more than half an year for this update ! This is mainly due I had to do a lot for my studies, especially preparing exams, and I was abroad for three months.

- * Mixed read/write with one filehandle possible (fixes many problems)
- * SetFileSize() now supported. Fixes this nasty bug when trying to set a variable shorter than it was before, using SETENV.
- * Warning requester for unsupported DOS packets
- * Protection bits reflect the state of a file/dir, see Technical Notes
- * Startup options and new MountENV
- * Other major changes in this guide:
Installation, Startup, Technical and Credits .

1.11 Future

Ideas for future releases:

- * Multiuser support, will also bring date, protection bits and filenotes.
 - * Workaround for the ixemul problem: Check directory structure of ENVARC:
 - * Let it survive resets
 - * Implement these few missing DOS-packets
 - * German documentation
 - * And all things I forgot to mention ...
-

Planned for next release:

- * Bugfixes, if there are some
- * Multiuser, date, protbits and filenotes support.

Probably there will come a "HappyRAM" handler some day. This will be HappyENV minus its autocopy (ENVARC->ENV) feature and plus a block oriented design to allow handling of big files and to speed it up.

1.12 Credits

Many thanks for postcards (and similar things) go to:

Dave Dustin, Tim Nugent, Ken de Veaux, Alex Kazik, Pete Dalling, Fredrik Hallenberg, Heikki Pora, Andre Wyrwa, Andreas Baum, Roland Bengtsson, Kevin Fairhurst, 8ight/nowhere, Carsten Dumke, Malcom Harnden, Thomas Wenzel, Piotr Pawlow, Petri Rätty, Steffen Köhler, Lothar Dall'O, Norman Shearer, Fergus o'Hea, Michael Schulz, Stuart Johnson, Angelo, Gianomo Mules, Mario Cattaneo, Ercl Ismael, Ian Grant, Gernod Schomberg, Stephane Payet, Magnus Holmgren, Hansjörg Schwarz, Wil, P. P. Martin, Andy, Petri Nieminen, Janne Salmijärvi, Brad Rogers, Juan Jesus Garcia de Saria, Steve Bellinguere, Graham J. Chilman, Nicola Smolenski, Colin Hinks, Douglas Brebner, Esquenet Alexandre, Andreas Schröder, Grzegorz Calkowski, Jürgen Weippert, Dieter Vogt, Bill Eaves, Kai Weber, Hans Ponsen, Faroul, Sven Thönissen, Andy Thomson, Joris Weidener, Mirko Ortolina, Dean Minett, Thomas Reinhard, Andreas Bombe, Warrik, Davide Precone, Tziastas Nikos, Hans-Joachim Reinsch, Michele Berionne, Nicholas M. Green, Stefan, Ji Yong Dijkhuis, Benoit Broc, G. C. Shaw, Rolf Rotvel, Danuel Colter, Shane Cracknell, Brian Jones, Robert Crapard, Dude, Barbara Dehn, Gunnar Andersson and Christoph Hornsteiner.

I love your postcards, they have made me happy !
Thank you !

I was surprised, how many people have sent me one. Feedback on my other programs was much smaller.

Sorry for my late answers, but I was very busy preparing exams.

Further thanks go to:

- * All those who have reported bugs and send suggestions
 - * Those who did some beta testing
 - * Erik for lending me his A1200
 - * Frank for using my programs (I persuaded him :-)
 - * Christian for writing lots of mails/postcards to me
 - * Amiga Shopper, Amiga Computing and Amiga User for free magazines.
-

1.13 Reporting Bugs

- * Very important: Please tell me your OS version and your CPU type.
A more detailed system description does not hurt, but is not necessary in many cases.
- * Also important: Please give me the version number of HappyENV you are using ! Without this I cannot track down the bug in many cases.
- * If you encounter the "Task XY uses packet Z" requester, please tell it to me, except when Disksalv causes it. You can then get rid of this requester using startup option NOREQ. It is quite harmless, especially if the number is 18.
- * Maybe it is helpful to tell me about commodities (like ToolManager), system replacements (like FastIPrefs), patches (like MCP) or other programs accessing ENV variables.
- * The MagicWB-demon works quite crappy. Tell me if you use it or, much better, use TragicWB or RealIcons (both on Aminet) instead.
- * Maybe you have strange things in your startup-sequence or your user-startup. You can watch execution having "set ECHO ON" as the first line in your startup-sequence.
- * Insert the line "C:copy ENVARC: ENV: ALL QUIET NOREQ" after the "MountENV" line back to your startup-sequence and reboot. If then the bug dissappears it is a problem with the autoload feature.
- * SnoopDos or DosTrace logfiles can be helpful.
- * The debugging version of HappyENV produces a nice logfile:
 - * delete l:HappyENV-Handler
 - * copy HappyENV-Handler.debug to L:HappyENV-Handler
 - * Get yourself a copy of SUSHI (Aminet/dev/debug), then insert this line in your startup-sequence before the "MountENV" line:
run >NIL: sushi <>CON:0/30/640/170/Sushi/AUTO/CLOSE/WAIT/INACTIVE ON BUFK=128 ↵
NOPROMPT ASKEXIT ASKSAVE
 - * You should see lots of output on the sushi window when booting again
 - * Save this output to T:sushi.out by pressing CTRL-F in the sushi window.

Please do not send me large logfiles (about >10KB) or other binary data without contacting me first. The reason for this is not a limit of my mailbox (it can handle more than 1MB), but to avoid that lots of people sending logfiles for the same bug. A good idea is to LHA-archive and uuencode the logfiles before sending.

1.14 Author

My snail mail address (fine for writing postcards :-)) :

Martin Gierich
Narzissenweg 7

76297 Stutensee
GERMANY

My EMail address:

gierich@gmx.de
or Martin.Gierich@stud.uni-karlsruhe.de

My homepage (beta versions are available here):

<http://www.home4u.de/gierich/dindex.html>

Please have a look at Limitations and Reporting Bugs before sending a bug report to me.

I am usually quite slow when answering EMail, because I have to do a lot for my studies. But I will make sure that everybody gets an answer. Be warned that I am extremely slow at answering letters (snail mail).

I am sorry, that I cannot answer to everybody who writes a postcard to me. I would get poor and I would have to sell my Amiga. :-)
If you want your letter to be answered, then please include return postage (or some kind of a gift). If you write your EMail address on your postcard/letter, I will surely answer using EMail, even if it takes a while.

My system config is:

AT A1200 (OS3.1), Blizzard 1230-IV (68030/50MHz with MMU, 68882/50MHz FPU) with SCSI-Kit, SVGA monitor, 2MB chip 16MB fast RAM.

Some other programs written by me can be found on Aminet:

- * util/shell/ZShell (A CLI replacement)
- * text/hyper/MiniXpkGuide (AmigaGuide/Multi-viewer)
- * gfx/misc/PromoteView (See anims etc. on VGA monitors)
- * biz/dbase/UnitedManager (For a game called United/Oberfoul)
- * util/cli/DMAP (Very old universal utility)

STAY HAPPY !