

MemMinister

Marcus Ohlström

COLLABORATORS

	TITLE : MemMinister		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Marcus Ohlström	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MemMinister	1
1.1	MemMinister v1.2 (21.5.95) - AmigaGuide document	1
1.2	Index	1
1.3	An introduction to MemMinister	2
1.4	Copyrights concerning MemMinister	2
1.5	Disclaimer information	3
1.6	Distribution limitations	3
1.7	What is required to use MemMinister?	4
1.8	What is the main features of MemMinister?	4
1.9	Using MemMinister	5
1.10	MemMinister's main window	6
1.11	Using MemMinister from Shell/CLI	6
1.12	The Flush gadget	7
1.13	The Snap gadget	7
1.14	Tooltype settings	7
1.15	Tooltype LEFT	8
1.16	Tooltype TOP	8
1.17	Tooltype ZIPLEFT	8
1.18	Tooltype ZIPTOP	9
1.19	Tooltype ZIPWIDTH	9
1.20	Tooltype ZIPPED	9
1.21	Tooltype TITLEFORMAT	10
1.22	Tooltype DECIMALPOINT	10
1.23	Tooltype SAVEPOSITION	11
1.24	Tooltype FONTNAME	12
1.25	Tooltype FONTSIZE	12
1.26	Tooltype DELAYTIME	12
1.27	Tooltype WINDOWTOBACK	13
1.28	Tooltype TOOLPRI	13
1.29	System Tooltype: STARTPRI	13

1.30 System Tooltype: DONOTWAIT 14

1.31 History of MemMinister 14

1.32 Future plans for MemMinister 15

1.33 Bugs or undocumented features? 15

1.34 How to contact the author 15

1.35 Thanks to the following people 16

1.36 Introduction to Users Standards Group 16

Chapter 1

MemMinister

1.1 MemMinister v1.2 (21.5.95) - AmigaGuide document

MemMinister v1.2 (21.5.95)

Copyright ©1995, Marcus Ohlström
All rights reserved.

Introduction
Copyright
Disclaimer
Distribution
Requirements
Features
Usage
Tooltypes

History
Future

Bugs

Author
Thanks

MemMinister is endorsed by Users Standards Group .
AmigaGuide® document written by Peter Bornhall.

1.2 Index

Contents

Introduction
Copyright
Disclaimer
Distribution

Requirements

Features

Usage

 Main window

 Shell/CLI usage

 Flush gadget

 Snap gadget

Tooltypes

 LEFT

 TOP

 ZIPLEFT

 ZIPTOP

 ZIPWIDTH

 ZIPPED

 TITLEFORMAT

 DECIMALPOINT

 SAVEPOSITION

 FONTNAME

 FONTSIZE

 DELAYTIME

 WINDOWTOBACK

 TOOLPRI

 STARTPRI

 DONOTWAIT

History

Future

Bugs

Author

Thanks

1.3 An introduction to MemMinister

MemMinister is a memory watch utility developed on and for the Amiga computer. It is aimed first and foremost at the programmer, which can benefit from the special features of MemMinister.

MemMinister is small, both in size and memory requirements, and very efficient. It is user-friendly and configurable.

MemMinister features a FLUSH option and a SNAP feature, making it very easy for programmers to make sure their programs return all memory allocated.

See also: Features

1.4 Copyrights concerning MemMinister

MemMinister is Copyright ©1995 to Marcus Ohlström. Marcus Ohlström will be referred to as the AUTHOR throughout this document.

MemMinister is FreeWare, NOT Public Domain. This means that you are allowed to use and distribute it without any restraints, though you are not allowed to change, reverse engineer, patch or in any way modify the files distributed in the MemMinister package. Nor are you allowed to re-use all or parts of the sourcecode without written permission from the author.

You are NOT allowed to include MemMinister in any commercial products without the written permission from the author.

See also: Distribution

1.5 Disclaimer information

MemMinister 1.2 has been tested and evaluated on a number of Amiga models, including A500+, A1200 with/without accelerator, A3000, A4000/040 and CDTV with no problems whatsoever. This is a reasonable amount of testing to assume that it will work with all Amiga models.

However, the author does NOT give you any warranties, expressed or implied, that MemMinister will work on your specific hardware.

All use of this program is at your own risk, and the author can not be held responsible for any damage that might occur to your system, either direct or indirect.

See also: Requirements

1.6 Distribution limitations

You are allowed to freely copy and redistribute the MemMinister package as many times and to anyone you like, just as long as the conditions described below are followed.

- You may NOT add any files to the distributed package, except FILE_ID.DIZ, FILEDESC.TXT and FILEPATH.LST.
- You may NOT remove any files from the distributed package. The following files MUST be present and unchanged;

Filename	Size
MemMinister.info	628
MemMinister/MemMinister	7000
MemMinister/MemMinister.guide	31600
MemMinister/MemMinister.guide.info	468
MemMinister/MemMinister.info	604
MemMinister/Source.info	628
MemMinister/Source/Build.info	841

MemMinister/Source/MemMinister.c	18000
MemMinister/Source/MemMinister.c.info	486
MemMinister/Source/MemMinister.h	1128
MemMinister/Source/MemMinister.h.info	486
MemMinister/Source/MemMinister_rev.c	65
MemMinister/Source/MemMinister_rev.c.info	486
MemMinister/Source/Rev.config	3
MemMinister/Source/Rev.h	214
MemMinister/Source/SCOPTIONS	133
MemMinister/Source/SCOPTIONS.info	470
MemMinister/Source/SMakeFile	827
MemMinister/Source/SMakeFile.info	838

- You may NOT sell the MemMinister package. You MAY charge a small amount for copying and distribution costs though.
- You may NOT redistribute the MemMinister package if it doesn't conform to all of the conditions mentioned above.

1.7 What is required to use MemMinister?

MemMinister requires at least AmigaDOS v2.0 (v36) or higher to run. It will harmlessly exit on older DOS versions, telling you what went wrong.

MemMinister limits the amount of displayable memory to 999.999.999 bytes, but that will probably not be a problem for the foreseeable future.

1.8 What is the main features of MemMinister?

One night when I was finishing some functions for my new program, I wanted to find out if those functions returned all the memory they allocated. I started a memory watch program, and it promptly decided to crash! I began looking for the other similar programs I got, but they didn't work the way I wanted. None of them accepted my font settings, or in other words, they looked awful!

I decided to write my own memory watcher, and because I never was satisfied with any of the existing ones, I decided to do the ultimate memory watcher! So, I began looking for bugs and misfeatures in every memory watcher I got my hands on, and tried to improve on everything. And of course I also added some other features and functions. Thus, MemMinister was born!

MemMinister is a continuing project, and I will continue to develop it until both you and I are fully satisfied.

Here is a list of the main features in MemMinister

- MemMinister is font-sensitive, unlike most other memory watchers.
- MemMinister offers an easy way to flush memory from unused

resources, like libraries, devices, fonts and other flushable resources.

- MemMinister features an easy-to-use snap function, enabling the user to see exactly what changes in memory usage takes place.
- MemMinister is very small and efficient. Uses hardly any CPU time, usually less than 0.2% to 0.6% on an A1200. Depending on current mode of operation.
- MemMinister features a zipped window, not higher than the screen title, easy to position, showing you the amount of free memory.
- Configurable window title in zipped mode.

1.9 Using MemMinister

To start MemMinister, simply double-click its icon, or type MemMinister in a Shell/CLI-window. MemMinister doesn't require any other resources, so the only thing you need to do to use it, is to simply copy it to wherever you want it.

See also: Shell/CLI Usage

Alright, let's put the settings aside for a while. When you have started MemMinister, you will see the main window with two gadgets and a textfield consisting of three rows divided into four columns. The two gadgets are the flush and snap gadgets.

Anyway, the main intention for MemMinister is to be able to see whether a program releases all memory it has allocated or not. This might not be the case with all programs (see notes below), but most programs should release all the memory used. Now, to check if YOUR program returns all memory, follow these simple steps;

1. Start MemMinister, if it isn't already running.
2. Flush the memory.
3. Snap current amount of free memory.
4. Start your program and use it the way it's supposed to be used.
5. Quit your program.
6. Flush memory yet again.

Now, if all values in the rightmost column are zero, your program has returned all memory it allocated. If greater than zero, your program has NOT returned all of the memory it allocated, which means you will have to find the erroneous routine. Should the values be negative, your program freed more memory than it allocated. That should not happen, but you will never know (would be nice, though :).

Note: Some programs, for example programs using the audio.device, does NOT free all memory when run. The reason for this is because (in this example) audio.device can't be totally flushed out from memory. So a recommendation would be to run your program at least once BEFORE working through steps 1-6 above.

Note: If you launch your program from a shell, make sure you use the standard Commodore Shell/CON:. This is because not all shells and console devices allocate all buffers immediately (KingCON, for example, does not allocate buffers until they are needed). Otherwise, this will make MemMinister show incorrect values since some of the used memory is being allocated by your shell/CON: and not by your program.

Note: When checking your program, you shouldn't do anything else! Even re-arranging a window may (de)allocate some memory, and the values shown by MemMinister won't be valid anymore!

Now then, you might want to know more about how you can configure MemMinister to your personal preferences, so why not take a look at the tooltypes and find out?

1.10 MemMinister's main window

MemMinister's main window consists of two gadgets and three rows of text. This is the main window, as opposed to the zipped window, which does NOT have these features.

The two gadgets are the flush gadget, and the snap gadget. The flush gadget removes unnecessary resources from memory, and the snap gadget makes a "snapshot" of the current amount of free memory (chip, fast and total).

The interesting thing though, is the three rows divided into four columns. These are the main feature of MemMinister. The first column simply tells you the type of memory for that row, while the second column constantly shows the current amount of free memory. The third column is what is called "snapped" memory.

Snapped memory is simply the amount of free memory when MemMinister was started, or when you last pressed the snap gadget. This value does not change unless you press the snap gadget. This brings us to the fourth column.

The fourth column is the amount of memory (de)allocated since you started MemMinister (or manually snapped memory). This allows you to see exactly how much memory has been allocated or freed (negative values).

1.11 Using MemMinister from Shell/CLI

MemMinister has VERY limited usage from shell in the current version. Only two (one, if you really want to argue about it) options actually, these two;

```
1> MemMinister      ; starts MemMinister with default settings.
```

```
1> MemMinister -    ; MemMinister will try to load the settings
```

; from its icon tooltypes .

Nothing fancy as you can see, but there will probably be some changes to this part in an upcoming release. See Future for more info.

1.12 The Flush gadget

When you click the flush gadget, located in the top left corner of the main window (no, not the close gadget), MemMinister will attempt to flush memory used by loaded, but unused, libraries, fonts, devices, and other "flushable" resources. This behaviour is the same as if you would type Avail FLUSH in a shell-window.

The Amiga OS will automatically free this memory if necessary, so you don't HAVE to do it manually. But in some cases it is nice to be able to flush as easy as clicking a gadget, especially when you want to monitor the memory use of a certain program (see procedure in Usage). Still, the OS will not flush memory unless it really has to, which means that if you're out of fast memory, but you've got some chip memory left, the OS will use chip memory instead of flushing and using the fast memory. To prevent this, you have to flush the memory manually.

1.13 The Snap gadget

The snap gadget is the gadget with the text " Free Snapped Used " in it. When pressing this gadget, MemMinister will make a "snapshot" of the current amount of free memory, and use as a comparison until next time you press the snap gadget.

1.14 Tooltype settings

You configure MemMinister by setting the tooltypes of its icon. The tooltypes you can use are;

LEFT	Set left edge of main window.
TOP	Set top edge of main window.
ZIPLEFT	Set left edge of zipped window.
ZIPTOP	Set top edge of zipped window.
ZIPWIDTH	Set width of zipped window.
ZIPPED	Tell MemMinister to start in zipped mode.
TITLEFORMAT	Set windowtitle format in zipped mode.
DECIMALPOINT	Set decimalpoint used in zipped window.
SAVEPOSITION	Save window position on exit.
FONTNAME	Set font to use in main window.
FONTSIZE	Set size to use in main window.
DELAYTIME	Set update interval in 1/50th of a second.
WINDOWTOBACK	Open MemMinister behind other windows.
TOOLPRI	SYSTEM: Priority of program.
STARTPRI	SYSTEM: WBStartup priority.

DONOTWAIT SYSTEM: Wait for program to finish?

1.15 Tooltype LEFT

LEFT

This tooltype allows you to specify the left edge of MemMinister's main window. Values ranges from 0. Incorrect values will be adjusted to prevent failures.

Format: LEFT=integer

Example: LEFT=125
 LEFT=0
 LEFT=-1567

Default: LEFT=0

See also: TOP ZIPPED

1.16 Tooltype TOP

TOP

This tooltype allows you to specify the top edge of MemMinister's main window. Values ranges from 0. Incorrect values will be adjusted to prevent failures.

Format: TOP=integer

Example: TOP=125
 TOP=0
 TOP=-1567

Default: TOP=0

See also: LEFT ZIPPED

1.17 Tooltype ZIPLEFT

ZIPLEFT

This tooltype allows you to specify the left edge of MemMinister's zipped window. Values ranges from 0. Incorrect values will be adjusted to prevent failures.

Format: ZIPLEFT=integer

Example: ZIPLEFT=125
 ZIPLEFT=0

```
ZIPLEFT=-1567
```

```
Default:  ZIPLEFT=0
```

```
See also:  ZIPTOP  ZIPPED  ZIPWIDTH
```

1.18 Tooltype ZIPTOP

ZIPTOP

This tooltype allows you to specify the top edge of MemMinister's zipped window. Values ranges from 0. Incorrect values will be adjusted to prevent failures.

```
Format:  ZIPTOP=integer
```

```
Example:  ZIPTOP=125
          ZIPTOP=0
          ZIPTOP=-1567
```

```
Default:  ZIPTOP=0
```

```
See also:  ZIPLEFT  ZIPPED  ZIPWIDTH
```

1.19 Tooltype ZIPWIDTH

ZIPWIDTH

This tooltype allows you to specify the width of MemMinister's zipped window. Incorrect values will be adjusted to prevent failures.

```
Format:  ZIPWIDTH=integer
```

```
Example:  ZIPWIDTH=325
          ZIPWIDTH=50
          ZIPWIDTH=-100
```

```
Default:  ZIPWIDTH=200
```

```
See also:  ZIPLEFT  ZIPTOP  ZIPPED  TITLEFORMAT
```

1.20 Tooltype ZIPPED

ZIPPED

This tooltype allows you to specify that you want MemMinister to open a zipped window immediately instead of opening its main window.

```
Example:  ZIPPED
          (ZIPPED)
```

Default: (ZIPPED)

See also: ZIPLEFT ZIPTOP ZIPWIDTH

1.21 Tooltype TITLEFORMAT

TITLEFORMAT

This tooltype allows you to specify exactly what information the windowtitle should contain when MemMinister is in zipped mode. The title will be a copy of the string you enter here, with the exception that you may enter control codes for special information.

All control codes consists of %[modifier]code. Where code is one of the following;

c or C	= Total amount of free CHIP memory
f or F	= Total amount of free FAST memory
t or T	= Total amount of free memory

The modifiers are as follows;

b or B	= Bytes (default modifier)
k or K	= Kilobytes
m or M	= Megabytes

As you can see, codes and modifiers may be specified either in upper or lower case characters. %cm, %cM, %Cm and %CM yields the same result.

If you need to put a percent-sign (%) in the windowtitle, you'll have to put two % in the string, i.e. use %% for every % you want...

Format: TITLEFORMAT=string

Example: TITLEFORMAT=Total: %tmMB
TITLEFORMAT=%cmMB CHIP, %fmMB FAST
TITLEFORMAT=%%% %S%t%r%a%n%g%e

Default: TITLEFORMAT=Total: %t

See also: ZIPPED DECIMALPOINT

1.22 Tooltype DECIMALPOINT

DECIMALPOINT

This tooltype allows you to specify the decimalpoint used for kilobytes and megabytes in zipped mode. This allows you to replace the point with a comma, which is used in some countries.

For example, let's say you use "TITLEFORMAT=%cmMB CHIP" and you enter

"DECIMALPOINT=mega". That would result in a windowtitle with the following string (for example)

1mega65MB CHIP

This means that MemMinister places the entire string where the decimalpoint is supposed to be. There is probably no sensible reason to use more than one character, but the possibility is there.

"DECIMALPOINT" and "DECIMALPOINT=" results in the default char.

Format: DECIMALPOINT=string

Example: DECIMALPOINT=,
 DECIMALPOINT=k

Default: DECIMALPOINT=.

See also: ZIPPED TITLEFORMAT

Note: This does not affect the main window.

1.23 Tooltype SAVEPOSITION

SAVEPOSITION

This tooltype tells MemMinister to remember its window position and window mode when it is closed. This means that it will open up the same way that you left it last time you used it.

This behaviour is very useful when you want to find some nice values for the LEFT, TOP, ZIPLEFT and ZIPTOP tooltypes. Start MemMinister, position the window, and quit MemMinister. No tedious work finding the values you want.

For some strange reason though, there is no way to find the zipped window position without actually zipping the window, so you will have to perform the above process once for the main window and once for the zipped window. Don't blame me, blame the OS for this.

Format: SAVEPOSITION

Example: SAVEPOSITION
 (SAVEPOSITION)

Default: (SAVEPOSITION)

See also: LEFT TOP ZIPPED

Note: The position will of course not be saved if the computer is reset or if it should crash.

1.24 Tooltype FONTNAME

FONTNAME

This tooltype allows you to specify the font used in MemMinister's main window. MemMinister defaults to the default SCREEN font. If the font specified isn't found, MemMinister will revert to the default font.

Format: FONTNAME=string

Example: FONTNAME=courier.font
FONTNAME=helvetica.font
FONTNAME=CGTimes.font

Default: (FONTNAME=)

See also: FONTSIZE

Note: If you want to use FONTNAME, you must also set FONTSIZE!

1.25 Tooltype FONTSIZE

FONTSIZE

This tooltype allows you to specify the size of the font specified in FONTNAME. If the value specified here is invalid, MemMinister will revert to the default font. All values greater than 0 is permitted, but if the font is too big, the window will be clipped.

Format: FONTSIZE=integer

Example: FONTSIZE=8
FONTSIZE=13
FONTSIZE=100

Default: (FONTSIZE=)

See also: FONTNAME

Note: FONTSIZE is only used in conjunction with FONTNAME.

1.26 Tooltype DELAYTIME

DELAYTIME

This tooltype allows you to choose how often MemMinister should update the window. The value is measured in 1/50 of a second. Lowering this value will increase the strain on the CPU. You could set DELAYTIME=0 and use TOOLPRI=-128, which would cause MemMinister to update its data continously, but ONLY as long as no other program wants the CPU!

Format: DELAYTIME=integer

Example: DELAYTIME=7
 DELAYTIME=100
 DELAYTIME=0

Default: DELAYTIME=12

See also: TOOLPRI

1.27 Tooltype WINDOWTOBACK

WINDOWTOBACK

This tooltype tells MemMinister to open its window behind all other windows.

Format: WINDOWTOBACK

Example: WINDOWTOBACK
 (WINDOWTOBACK)

Default: (WINDOWTOBACK)

1.28 Tooltype TOOLPRI

TOOLPRI

This is a system tooltype, used to set the priority of a program. The higher the priority, the more CPU-time a program gets. Values ranges from -128 to 127, but you should NOT use priorities greater than 5!

Format: TOOLPRI=integer

Example: TOOLPRI=0
 TOOLPRI=-1
 TOOLPRI=5

Default: TOOLPRI=0

Note: This is a system tooltype, which means that it can be used with any program, not just MemMinister.

1.29 System Tooltype: STARTPRI

STARTPRI

This is a system tooltype, which tells Workbench in which order the programs in WBStartup should be started. Higher STARTPRI programs are started before the programs with lower values. Values ranges from -128 to 127, but I recommend you to use -128. That way MemMinister will

probably show memory usage as 0 when it starts.

Format: STARTPRI=integer

Example: STARTPRI=15
 STARTPRI=-128
 STARTPRI=0

Default: STARTPRI=0

Note: This is a system tooltype, which means that it can be used
 with any program, not just MemMinister.

1.30 System Tooltype: DONOTWAIT

DONOTWAIT

This is a system tooltype, telling Workbench not to wait for a program to exit. Using MemMinister, you should make sure that this tooltype is set, otherwise Workbench will pop up a requester asking you if you want to wait for MemMinister to finish (which you don't, you want it running).

Format: DONOTWAIT

Example: DONOTWAIT
 (DONOTWAIT)

Default: (DONOTWAIT)

Note: This is a system tooltype, which means that it can be used
 with any program, not just MemMinister.

1.31 History of MemMinister

v1.2 21-May-95 rev 36.308

- Limited Shell/CLI support.
- User-configurable window positions, with save option.
- MemMinister can now be started in zipped mode.
- Alternative font possible for main window.
- User-configurable update interval.
- Fully configurable window title in zipped mode.
- AmigaGuide document entirely rewritten.

v1.1 22-Aug-94 rev 37.185

- MemMinister now shows separate information about chip, fast and total amount of memory.
 - Added "Snapped" and "Used" fields.
 - Added snap gadget.
-

- Added flush gadget.
- Old v1.0 mode when window is zipped, i.e. when you click the zoom-gadget, you will get a small window showing the amount of total free memory in the windowtitle.
- Documentation rewritten in AmigaGuide format.

v1.0 15-Jul-94

- Simply opens a window showing the amount of free memory in the windowtitle.

1.32 Future plans for MemMinister

What are the plans for future releases? Well, this is a small list of things that might be incorporated soon, but if you have any suggestions to add, please let me know! Click [here](#) to see the author's address.

- Better Shell/CLI support.
- Extended support for configuring window title in zipped mode.
- Limited menu support for saving tooltype settings.
- Feature that unzips window when activated, and vice versa.

1.33 Bugs or undocumented features?

I'm currently not aware of any bugs in MemMinister, but there is always the possibility of bugs creeping in.

If YOU find any bugs, please contact me as soon as possible.

I would appreciate if you could include detailed information about your system, such as computer model, amount of memory, screenmode, fonts, peripherals, kickstart and workbench versions. And please, try to include a detailed description of how to recreate the problem. That would help a lot and make it easier to find the bug(s).

1.34 How to contact the author

If you want to contact the author, use one of the following addresses:

Snail: Marcus Ohlström
 Stopvägen 104
 S-16143 BROMMA
 SWEDEN

e-mail: marcus@karkis.canit.se

FidoNet: Marcus Ohlstrom @ 2:201/417.31 (or 2:201/411.99)
AmigaNet: Marcus Ohlstrom @ 39:164/104.31
USG-Net: Marcus Ohlstrom @ 8:102/104.31

I will try to reply, especially to electronic mail. If you want to be sure your letter reaches me, use snail-mail, but if you want a fast reply, use e-mail or FidoNet netmail.

1.35 Thanks to the following people

I would like to thank the following people, who have aided and supported me in getting MemMinister released (no particular order);

Peter Bornhall,
Stefan Pettersson,
Mattias Ahnberg
and Jim Franzén for beta-testing MemMinister

Peter Bornhall for all his ideas and writing this documentation

Tomas Kärki for being my Fido-boss, and supplying me with a
beta-testing echo

Terry Wood for being the first person sending me a letter
about MemMinister

1.36 Introduction to Users Standards Group

AN INTRODUCTION TO USERS STANDARDS GROUP

USG is a non-profit organisation of users and developers who are willing to support and aid in the development of high quality software and standards. As such, USG and USG-Net will help developers with ideas to make their product better, and to encourage them to actually release their products. USG will also help with suggestions, tips, technical details, and to provide betatesters for these products. Developers connected to USG-Net will also find it useful to set up their own echo, which will provide them with even more help in finding the "right way" for their product.

Joining USG is completely free of charge, and there is currently no reason to change this policy. As a member of USG, you are not required to do anything, so those who just feels that they want to support the idea should not be discouraged. Also, USG allows entire development teams to join. To join USG, fill in the following information;

Name	- First and last name, or name of development team
Address	- Withheld from the public eye on request
Country	- Your country
Born	- Date of birth, optional
E-mail	- Any E-mail and/or netaddresses you have
System	- Platform and system you use
Products	- Information on any products you are developing

Also, let us know if you are a developer, if you want to take active part in development, or if you are just supporting USG. If you are interested in becoming a betatester, enclose full details on your system setup and software. This information will then be sent to the betatester coordinator.

Send your membership info to;

Snail: Peter Bornhall
Back 5, Frammestad
S-46598 NOSSEBRO
SWEDEN

e-mail: bornhall@karkis.canit.se

FidoNet: Peter Bornhall @ 2:201/417.41
USG-Net: Peter Bornhall @ 8:102/104.41
AmigaNet: Peter Bornhall @ 39:164/104.41
