

# **MemClear Release 1.8 documentation**

**COLLABORATORS**

	<i>TITLE :</i> MemClear Release 1.8 documentation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 28, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>MemClear Release 1.8 documentation</b>	<b>1</b>
1.1	MemClear Release 1.8 documentation	1
1.2	Disclaimer	1
1.3	Purpose	2
1.4	Documentation	3
1.5	System Requirements	3
1.6	Shell Usage	3
1.7	Examples	5
1.8	Error Codes	6
1.9	Memory Checker	6
1.10	Author	7
1.11	ThanX	7
1.12	Speed	7
1.13	History	8

# Chapter 1

## MemClear Release 1.8 documentation

### 1.1 MemClear Release 1.8 documentation

```

===== <->
===== MemClear V 1.8 =====
=====

```

MemClear © Copyright 1991-1994 by Ralf Thanner

```

Ralf Thanner
Ellerstr. 83
40227 Düsseldorf
GERMANY

```

Disclaimer

Purpose

Documentation

Author

ThanX

Speed

History

### 1.2 Disclaimer

```

===== <->
===== Disclaimer & Distribution =====
=====

```

The executable and non-executable parts of this software package may NOT be altered by any means (this includes editing, crunching, and reverse-engineering), except archiving. The

author

is in NO way

liable for any changes made to any part of the package, or consequences thereof as he is in NO way liable for damages or loss of data directly or indirectly caused by this software.

Distribution

-----

Neither fees may be charged nor profits may be made by distributing this piece of software. Only a nominal fee for cost of magnetic media may be acceptable. Outside a single machine environment, you are NOT allowed to reproduce only some parts of the package, but you have to copy it completely. See this list of contents for verification:

- MemClear (dir)
- MemClear.doc
- MemClear.doc.info
- MemClear.guide
- MemClear.guide.info
- MemClear
- MemClear.info

If any parts were already missing when you received this package, look out for another source to get your software in future.

### 1.3 Purpose

===== <->

===== Purpose =====  
=====

MemClear clears, fills the unused memory chunks. After clearing memory, it prints the amount of cleared bytes in CHIP- and FAST MEMORY and the total amount of free bytes. Keep in mind that MemClear now clears and fills with blazing speed.

Due to the fact that MemClear will show you the amount of free memory, how much is in-use and the largest available memory blocks you can also use it instead of the Avail command. It can also flush all currently unused devices, libraries and fonts from memory. MemClear is pure, too, and for those reasons a good replacement for Avail.

One of the most powerfull routines included in MemClear is the automatic

Memory Checker



===== <->

===== Usage =====  
 =====

MemClear cannot be started from Workbench, it is a Shell tool!

The following commands are supported:

"AVAIL" - Displays the free memory without clearing. Avail does NOT count the real free bytes. It is just an avail clone.

NOTE: if you have 'DynamiCache' in the background the amount of free bytes shown may differ from the amount of memory you have inbuilt.

"FLUSH" - Flushing all currently unused devices, libraries and fonts from memory. After flushing memory MemClear will print how much more memory is free now.

"NoClear" - This option is useful if you just want your memory flushed without clearing. Especially if you're using DynamiCache, this opt will be very useful in case you just want to flush unused devices, fonts etc.

"NoWarn" - This option switches off the obligatory Memory Checker and should only be used if you are either running OS3.x (performs also a memory sanity check) or if you really know what you are doing.

"QUIET" - All actions will be performed silent.

Following options are allowed:

"F" - Fills only free fast-memory.

"C" - Fills only free chip-memory.

If you don't want the memory cleared, but rather be filled use a 'FILL VALUE' which is defined this way:

---

- "-\$FE" - Fills memory with \$fe instead of zeros. All hexadecimals are allowed. One single hex value, e.g. '\$b' will be expanded to '\$bb'.
- "-A" - Fills memory with chosen character. This option IS case sensitive.

Please refer to the

Examples

section for further explanation.

If MemClear is invoked without any option it will simply fill the free memory with zeros and print the (real) free amount of memory. Real free memory means that this is not necessarily the amount of free memory the system has stored in its memheader structure, but rather the amount of bytes encountered in the memory chunks.

## 1.7 Examples

===== <->

===== Examples =====

MemClear F

clears the fast-mem area with zeros and won't touch a single byte in chip memory.

MemClear FLUSH C -\$ff

first flushs unused stuff from memory and then fills all free chip-mem with the hex-value '\$ff' leaving the fast-mem area untouched.

MemClear AVAIL

displays complete memory statistics, comparable to Commodore's Avail command without filling, clearing or flushing.

MemClear FLUSH NOCLEAR

flushs unused system resources, such as fonts, libraries, and devices from memory without filling or clearing. You may configure a "MemFlush" button or menu inside DirectoryOpus using this template. That's very useful if you're running out of memory. MemClear returns

Error Codes  
of 5 (WARN) if no unused memory could be freed.



## 1.8 Error Codes

```

=====
===== Error Codes =====
=====

```

MemClear will return an error code if something went wrong. This is very useful for 'FAILAT' and 'IF' commands in batch skripts.

```

RETURN WARN (05) for zero bytes gained with FLUSH.
RETURN ERROR (10) for all internal errors (e.g. no arp.lib).
RETURN FAIL (20) for an error that was encountered by the

```

Memory Checker

.

Example:

```
-----
```

```

FAILAT 5
  MemClear FLUSH QUIET           ; will flush memory without
                                ; printing any text...
IF WARN
  TYPE "flush_brachte_nichts.txt" ; "flush_wasnt_successfull"-)
ELSE

```

## 1.9 Memory Checker

```

=====
===== Memory Checker =====
=====

```

MemClear performs two different structure tests. The first scans through all memory nodes checking them against correct values and boundaries. When something went wrong the following message will be printed " Warning! Memory List is corrupted at: '\$address' ". If you read this SAVE EVERYTHING IMMEDIATELY, because sooner or later your Amiga will crash! There are some bad written programs which destroy parts of the internal memory lists. These programs, like an old version of decrunch.library did, WILL cause a crash if memory is flushed or cleared, even if you have already 'quitted' them! There is no guarantee that the printed address is the location where the memory list is corrupted, but you have a good chance for correctness.

The second test just checks if the amount of 'real' free bytes, counted by MemClear while scanning through all memory nodes, is the same as stored in Exeabase. When this error message appears there is also something wrong with your system but normally it won't crash in the next seconds.

## 1.10 Author

```
=====
===== Author =====
=====
```

MemClear V1.8 was written by Ralf Thanner using the Devpac assembler, the original MemClear idea was suggested from MemClear by John Hodgson which had some substantial disadvantages.

The main disadvantage was that it did not clear anything. If you don't believe me test it, use a monitor to load a file directly into memory, e.g. at location \$50000. Now start MemClear by John Hodgson. After MemClear has finished use a monitor to control your previously choosed location. Nothing has changed, no zeros... funny, isn't it?

## 1.11 ThanX

```
=====
===== ThanX =====
=====
```

Timm Müller - For help with documentation/guidefile and some guide on the way to good readable and structured sources.

Olaf Barthel - For supporting me with the original version of MemClear done by John Hodgson.

Martin Berndt - For telling me that MemClear is really useful because it helped programming and debugging the PICASSO II graphics card software. Sometimes it is really nice to get such a good response. Also thanx for finding the DynamiCache 'bug'!-)

## 1.12 Speed

```
=====
===== Speed comparison table =====
=====
```

complete memory clearing / filling using MemClear v1.8

---

```

A500, MC68000 7Mhz, 512kb Chip + 2MB Ranger Mem : 3,0 MB/sec
A1000, MC68010 7MHz, 512kb Chip + 4MB Fast Mem : 3,1 MB/sec
A3000, MC68030 25MHz, 2MB Chip + 4MB Fast Mem : 8,5 MB/sec

```

all tests performed under realistic circumstances, that means after booting fully-featured Workbench from HD.

## 1.13 History

```

=====
===== History =====
=====
----- V1.8 -----
- clear and fill algorithm now fits perfectly into the 680x0
  cache, which leads to an remarkable
    speed
    gain.
- the
    Memory Checker
    is much better now and should be able to
  handle all kind of structure errors.
- MemClear will now return 'Error Codes' for usage in batch
  skripts.

----- V1.7 -----
- added the check for 'real' free bytes and the value stored
  in Exeabase. With this routine implemented earlier I would
  have found the reason for mysterious crashes after using
  'decrunch.library' much earlier;-)
- if you use 'FLUSH' all available memory will be printed.
- after flushing the memory gain will be shown.
- MemClear is now using official 'cli-readargs' routines,
  instead of my own home brewn interface.
- included fill-pattern routine in normal clear routine.
- rewrote clear; clears/fills now about 3.1MB/sec on MC68010!
- command 'avail' shows now REALLY everything Avail does;-)

----- V1.6 -----
- implemented 'FLUSH' option.
- fixed small (and silly) bug in the 'byte counting' routine.

----- V1.5 -----
- fixed small bug in the cli-interface.

```

- MemClear is now PURE.

---

#### ----- V1.4 -----

- added choice which mem-area should be cleared.
- improved the cli-interface (e.g. 'f -a' and '-a f' are now both recognized).
- fixed some troubles with MC68040.

---

#### ----- V1.3 -----

- code completely rewritten.
- new clear algo which fits now perfect into the 68020/30 cache and also takes advantage of the 68010 prefetch!-) Clear needs now 1 second for 2.4MB on 68010 (average value, may differ).
- Blitter won't be used to clear chip-mem anymore, due to the fact that the blitter in the A3000 or A4000 is slower than the processor (MC68040 speed rulez!).
- added different fill-patterns. This one uses a real simple and 'slow' clear algo but who cares!?)

---

#### ----- V1.2 -----

- Added the 'free mem' text in instructions.
- Fixed the 'print-txt' routine.

---

#### ----- V1.1 -----

- Removed the FAST OPTION, since it's fast enough without this nasty method.
- Clears now \$C00000 mem on A500.
- Added the 'total free bytes' for a complete overview of free memory, MemClear shows now almost everything AVAIL does.

---

#### ----- V1.0 -----

- Improved clear algorithm, needs now less than two seconds for 2.5 MB.
- added special 'FAST OPTION' which is a pure hardware hack. (only usefull when you have more than 5 MB; disables all.)

---

#### ----- V0.9 -----

- Thought, that this is the final version... No bug reports.
  - Works fine, but clears slow (about 4 seconds for 2.5 MB).
-