

**FastExec**

**COLLABORATORS**

	<i>TITLE :</i> FastExec		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 9, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>FastExec</b>	<b>1</b>
1.1	FastExec.guide . . . . .	1
1.2	introduction . . . . .	1
1.3	requirements . . . . .	2
1.4	installation . . . . .	2
1.5	usage . . . . .	3
1.6	comments . . . . .	5
1.7	bugreports . . . . .	5
1.8	history . . . . .	6
1.9	credits . . . . .	7
1.10	author . . . . .	8

---

# Chapter 1

## FastExec

### 1.1 FastExec.guide

=====

June 10, 1997

Torbjörn Andersson  
presents  
FastExec 2.8

Public Domain

=====

#### 1. Contents

Contents	1
Introduction	2
Requirements	3
Installation	4
Usage	5
Comments	6
Bugreports	7
History	8
Credits	9
Author	10

### 1.2 introduction

#### 2. Introduction

-----

Most things automatically utilize the fast memory in my Amiga, but not `exec.library`, the heaviest used library of the AmigaOS, nor `expansion.library`. They remain in the slow

---

chip memory. Since that is a waste of both time and chip memory, I made FastExec.

FastExec will install a resident tag with the priority of `expansion.library` plus one. After `exec-` or `exec-` and `expansion.library` is initialized upon reboot, FastExec's tag will be run. First it will set the cache if it was specified on the command line. Then it patches `AddMemList()` in `exec.library`. If specified it will finally add non-autoconfig memory. Then we leave the control back to the system.

When someone calls `AddMemList()`, the patch uses the original function to add the memory to the system's list. Then it tries allocate fast memory and move `exec.library` to it. There are options to also move other things to fast memory.

If `expansion.library` cannot be moved to fast memory because it is added later, `AddLibrary()` will be patched to try to move it when it comes around.

Another feature with FastExec is that it can patch `exec's` interrupt routines, to speed up the system a little extra.

Under kickstart 1.2/1.3, `MakeLibrary()` is automatically patched so library bases will be longword aligned.

## 1.3 requirements

### 3. Requirements

---

FastExec requires true fast memory.

## 1.4 installation

### 4. Installation

---

Edit `S:Startup-Sequence` to start FastExec. I recommend putting it right after `SetPatch`, or maybe even first. The first time you try it, edit the first lines to something like this:

```
ask "Use FastExec?"
if warn
    FastExec REBOOT
endif
```

Then you can use your usual environment without depending on FastExec to work, there's no guarantee it will.

---

Don't redirect output to NIL:, FastExec will only give information if anything goes wrong.

## 1.5 usage

### 5. Usage

```
-----
Usage: FastExec [SYSINFO] [REBOOT] [NOEXEC] [FREEOLD] [PATCH]
             [FASTSSP] [FASTVBR] [FASTEXP] [FASTMEM] [FASTINT]
             [CACHE 0xhhhhhhhh] [ADDRESS 0xhhhhhhhh]
             [ADDMEM <base size attr pri> ...]
```

Any value can be written in both normal and hexadecimal form. Write "0x" in front for hex.

The options has no effect if FastExec already is resident (except for SYSINFO).

#### SYSINFO

-----

Shows various system information.

#### REBOOT

-----

Installs resident tag and reboots. Otherwise only the tag will be installed and no reboot will be performed.

#### NOEXEC

-----

Won't try to move exec.library to fast memory, in case it doesn't work and you would like to use the other options.

#### FREEOLD

-----

Will free the chip memory previously used by exec.library etc. when they are moved to fast memory (8264 bytes on my machine).

If you use this option you may get trouble to make more things than FastExec resident, since they can get a piece of that memory that gets overwritten at reboot.

#### PATCH

-----

Patches exec's interrupt routines.

Main feature is that ExecBase is embedded in lea-instructions instead of being read from location \$4, which can be useful if you have AGA chipset and/or use Enforcer.

#### FASTSSP

-----

Relocates the supervisor stack to fast memory.

---

FASTVBR  
-----

Relocates the vector base register to fast memory.

The 68000 processor has no VBR. (Output from SYSINFO will always say that it is zero).

FASTEXP  
-----

Relocates expansion.library to fast memory.

FASTMEM  
-----

Relocates memory list headers to fast memory.

Interesting side effect is that ShowConfig (version 39.1) says I have ~3970.0 meg CHIP...

FASTINT  
-----

Relocates memory for interrupt server lists to fast memory.

CACHE  
-----

Calls CacheControl() with the supplied parameter. Below are useful values as defined in exec.i. Logically "or" these together. Note that 68000 and 68010 has no cache. 68020 has instruction cache only.

```
CACRF_EnableI      = 0x00000001 (Enable instruction cache)
CACRF_IBE          = 0x00000010 (Enable instruction burst)
CACRF_EnableD      = 0x00000100 (Enable data cache)
CACRF_DBE          = 0x00001000 (Enable data burst)
CACRF_CopyBack     = 0x80000000
```

ADDRESS  
-----

Wait for memory with start at this address to be added before moving things.

If this memory has the highest priority (when it is added), exec.library etc. will be moved to it.

ADDMEM  
-----

Adds memory to the system's memory list. Can be useful if you have non-autoconfig memory. Use "5" for attributes if you don't know what else to use.

I'll take my memory as an example. This is the info I get about it from FastExec with the SYSINFO parameter:

MEMORY HEADERS:

Address	Name	Lower	Upper	Type	Pri
\$08000000	DKB1240_Memory	\$08000020	\$08800000	\$0005	10

If it was configured manually and I wanted FastExec to do it from now on, I would write:

```
FastExec ADDMEM 0x08000000 0x00800000 5 10
```

## 1.6 comments

### 6. Comments

-----

FastExec stays resident in memory, but after a reset it can happen that the fast memory is not located at its place. Then the system won't find exec.library with its information about resident things in memory. FastExec and for example the RAD-disk can't survive a reboot then. I guess this depends on the memory board you have.

I know there are other problems with FastExec, but I don't think I can make it work everywhere. Maybe it's possible, but I don't have much time or nice machines to test it on. In case you want to try, the source code is included in the archive.

## 1.7 bugreports

### 7. Bugreports

-----

If you report problems, please be as precise as possible. Simply telling me that FastExec "is buggy" or "doesn't work" is of course useless. Exactly what do you do, and what happens?

I want to know what option(s) that doesn't work. If you only use REBOOT and NOEXEC, there shouldn't be any trouble (as this basically does nothing except rebooting once). Tell me if it's removing NOEXEC that doesn't work, or the use of any of the extra options. Which one(s)?

How does FastExec seem to work if you use the original Startup-Sequence (except adding FastExec) and no extra utilities? Have you tried with SetPatch first and FastExec right after? If you use the bootmenu to turn off caches, does it make any difference?

How does FastExec work with the debugging tools Enforcer and MungWall?

If FastExec isn't compatible with some other program(s), which are they? Are they free from Enforcer hits?

Also, please give me the output from "FastExec SYSINFO", and

---



maybe information about (special) hardware you use.

## 1.8 history

### 8. History

---

#### 1.0 (26.11.95)

- First release.

#### 1.1 (14.1.96)

- lib\_NegSize under KS 1.3 is now set to correct value and not 0.
- Doesn't set the memory attribute flags MEMF\_LOCAL/MEMF\_24BITDMA under KS 1.3.
- The name string "expansion ram" is used for the fast memory under KS 1.3, instead of "Fast Memory".
- The priority that the fast memory has will not be changed, it was always set to 10 before.
- Could get wrong idea of where the kickstart was located, some checking together with alternative methods makes it safer now.
- If kickstart version is 2.0 or higher, FastExec will search through the kickstart for everything it needs from it. If FastExec will work doesn't depend on the exact kickstart version, but more on how it is built. If there aren't too big differences, FastExec should now work with any kickstart version.
- Changed some output from the SYSINFO argument, and documented that the argument exists. :)
- BOARDADDR handles more than one address.

#### 2.0 (21.4.96)

- Rewrote the main code, should work better out there.
- Can relocate a few extra things to fast memory, relocating expansion.library is optional.
- Added patches for much used functions in exec.
- Can set caches very early on bootup.
- Can add non-autoconfig memory.
- SYSINFO shows all libraries, and in what type of memory it is located.
- FastExec is now freeware.

#### 2.1 (25.8.96)

- Fixed SYSINFO output of FPU.
- When I changed mouse, FastExec always disabled the fast memory. I guess I could have fixed that, but I removed the option instead.
- Doesn't free the old chip memory header under kickstart 1.2/1.3, as it is misaligned (FASTMEM option caused FreeTwice-alert).
- Fixed bug with FASTINT under kickstart 1.2/1.3.
- Added NOFASTEXEC option.

#### 2.2 (29.8.96)

---

- Added LOCAL option.

#### 2.3 (11.9.96)

- Renamed NOFASTEXEC option NOEXEC.
- Added PCMCIA option.
- AddLibrary()-patch restores old function after expansion.library has been moved.

#### 2.4 (9.11.96)

- ~Gives some errors if installing fails.
- Restores old AddMemList()-function after the first memory of type MEMF\_FAST has been added.

#### 2.5 (18.1.97)

- Made FastExec public domain, and included the source code in the archive.

#### 2.6 (21.5.97)

- Fixed bug in Switch()-patch for machines with FPU.
- Sets CACRF\_WriteAllocate when CACHE option is used.
- Allocates memory for resident tag with MEMF\_REVERSE.
- Uses ReadArgs() (KS 2.0+).
- ~Doesn't free memory for old exec.library under KS 1.3 if eb\_ExecBase in expansion.library can't be replaced.
- Removed PCMCIA option.

#### 2.7 (28.5.97)

- Doesn't free memory used by things that are moved to fast memory, unless FREEOLD option is used.
- SYSINFO lists KickMem and KickTag.
- Removed LOCAL option.
- ~ADDMEM didn't accept negative priority.
- Doesn't restore old AddMemList()-function.
- Better check in AddMemList()-patch of memory to be added compared to whats already added.
- Tries to move expansion.library directly in AddMemList()-patch. Only if it hasn't been added, AddLibrary() is patched to do the job.
- Uses arp.library/GADS() under KS 1.3.
- Cleaned up interrupt patches. Doesn't patch interrupt 7. Works on KS 1.2/1.3.
- Removed patches for Exec-functions, they were far away from perfect.

#### 2.8 (10.6.97)

- Fixed bug in FASTVBR option.
- Added ADDRESS option.
- Allocates 8 extra bytes before the resident tag, which is necessary since they can get trashed at boot time.

## 1.9 credits

### 9. Credits

-----

---

Thanks to:

- All who registered when FastExec was shareware (or afterwards for that matter).
- All who have helped me in my more or less successful attempts in fixing problems.
- Amiga Shopper for politely asking before putting FastExec on their coverdisks and even sending me a copy of the magazine.

## 1.10 author

10. Author

-----  
FastExec executable and documentation are made by  
Torbjörn A. Andersson.

Email:  
d95ta@efd.lth.se

Home page:  
<http://www.efd.lth.se/~d95ta>

Home address:  
Torbjörn Andersson  
Knöppletorp 4379  
S-380 31 LÄCKEBY  
SWEDEN

I love you - AMIGA

---