

doc/NerveEX

Torsten Ebeling

COLLABORATORS

	<i>TITLE :</i> doc/NerveEX		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Torsten Ebeling	February 24, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	doc/NerveEX	1
1.1	doc/NerveEX.guide	1
1.2	doc/NerveEX.guide/COPYRIGHT	1
1.3	doc/NerveEX.guide/INTRODUCTION	2
1.4	doc/NerveEX.guide/REALLY_QUIT	2
1.5	doc/NerveEX.guide/REALLY_QUIT_2	2
1.6	doc/NerveEX.guide/INSTALLATION	3
1.7	doc/NerveEX.guide/NERVEEXPREFS	3
1.8	doc/NerveEX.guide/LIBRARY	4
1.9	doc/NerveEX.guide/FUNCTION	4
1.10	doc/NerveEX.guide/TASK	4
1.11	doc/NerveEX.guide/STRUCTURE	5
1.12	doc/NerveEX.guide/MEMBER	5
1.13	doc/NerveEX.guide/IDENTIFICATION	6
1.14	doc/NerveEX.guide/GD_NEW_LEVEL	6
1.15	doc/NerveEX.guide/GD_ADD	6
1.16	doc/NerveEX.guide/GD_DEL	6
1.17	doc/NerveEX.guide/BOX	6
1.18	doc/NerveEX.guide/MN_COPY	7
1.19	doc/NerveEX.guide/MN_PASTE	7
1.20	doc/NerveEX.guide/MN_FONT	7
1.21	doc/NerveEX.guide/MN_TRACE	7
1.22	doc/NerveEX.guide/DEBUG	7
1.23	doc/NerveEX.guide/NERVEEX	8
1.24	doc/NerveEX.guide/TROUBLESHOOTING	8
1.25	doc/NerveEX.guide/QUESTIONS	8
1.26	doc/NerveEX.guide/TECHNICAL_DES	9
1.27	doc/NerveEX.guide/KNOWN_BUGS	10
1.28	doc/NerveEX.guide/FUTURE_PLANS	11
1.29	doc/NerveEX.guide/ACKNOWLEDGMENTS	11

Chapter 1

doc/NerveEX

1.1 doc/NerveEX.guide

```
          NerveEX
(Hide bothersome requesters and other stuff)
          User's Guide
          Version 1.0
          $Date: 95/02/06 17:02:22 $
          written by Torsten Ebeling
          email:  t_ebelin@informatik.uni-kl.de
```

CONTENTS

- 0. Copyright
- 1. Introduction
- 2. Installation
- 3. NerveEXPrefs
- 3a. Debug
- 4. NerveEX
- 5. Troubleshooting
- 6. Questions and Answers
- 7. Technical description
- 8. Known bugs
- 9. Future plans
- 10. Acknowledgments

1.2 doc/NerveEX.guide/COPYRIGHT

0. COPYRIGHT

IMPORTANT NOTICE: This program is copyrighted by Torsten Ebeling, but

can be freely distributed, provided that the following rules are respected.

- No change is made to the program nor to the accompanying documentation
- The package is always distributed in its complete form.
- Every form of distribution is allowed and encouraged, but no fee can be charged for this program except for, possibly, the cost of magnetic media and/or disk duplication and shipping.
- Inclusion in PD software libraries such as Fish Disks is allowed, provided the fees charged for these disks are comparable with those charged by Fred Fish.
- The program cannot be distributed in any commercial product without the written consent of the author.
- The author cannot be held liable for the suitability or accuracy of or any damage directly or indirectly caused by the use or misuse of this program. It is the sole responsibility of the user her/him self.

By copying, distributing and/or using the program you indicate your acceptance of the above rules.

1.3 doc/NerveEX.guide/INTRODUCTION

1. INTRODUCTION

NerveEX is a small program to keep selected tasks from executing particular library functions.

My incentive for writing NerveEX was to allow hiding requesters like these:

```
Do you really want to quit this program?
```

I tried to make the GUI as simple to handle as possible, but I doubt it's truly novice user safe.

```
IF YOU DON'T KNOW ANYTHING ABOUT JUMP-VECTORS IN LIBRARIES OR  
PROGRAMMING YOUR AMIGA STOP HERE AND DELETE ME.
```

There is a very short description of libraries in the technical description part of this manual.

1.4 doc/NerveEX.guide/REALLY_QUIT

```
Are you sure you want to quit this program?
```

1.5 doc/NerveEX.guide/REALLY_QUIT_2

etc. etc. etc. ...?

1.6 doc/NerveEX.guide/INSTALLATION

2. INSTALLATION

NerveEX requires at least OS2.04 to run. NerveEX is very simple to install.

Use the installer script or execute the following steps to install NerveEX:

- Copy NerveEX and NerveEXPrefs to your preferred path for utilities.
- Copy NerveEX.prefs to ENVARC: or generate your own preferences file.

Now change the preferences file at will and enjoy programs without being bothered by annoying requesters.

1.7 doc/NerveEX.guide/NERVEEXPREFS

3. NERVEEXPREFS

In order to enter all the settings that are needed for NerveEX to work, there's a GUI provided with the program. NEVER modify the configuration file generated by NerveEXPrefs by hand. If this file is not what it's supposed to be, NerveEX may crash or not work properly.

Here's a description of the gadgets and what they do:

```

_____ ...
| |
| |-- Library
| | |
| | |-- Function
| | | |
| | | -- Task
| | | |
| | | -- Structure
| | | |
| | | -- Member
| | | |
| | | -- Identification
| | | |
| | | ____
| | | --|____| Function
| | | ____
| | |--|____| Library
| | |
| . .
| etc.
|_____ ...
_____

```

```
|_____| New Level Add Del
|_____Stringgadget1_____| |_____Stringgadget2_____| |Cyclegadget|
```

Here's a description of the advanced menu items and what they do:

- Font
- Trace calls

All the other menu items are standard menu items and need not be explained further.

1.8 doc/NerveEX.guide/LIBRARY

3.1.1 Library

This entry let's you define the library in which the following functions will be patched. This entry MUST end with ".library".

Example: "intuition.library"

See technical description for more details.

1.9 doc/NerveEX.guide/FUNCTION

3.1.2 Function

This entry let's you define the function which you want to patch in the library entered above. To do so you must enter the offset for this function. The offset will be automatically scanned by NerveEXPrefs if you have installed fd-files on your harddisk and made an "assign FD:" to this directory. The offset has to be entered in the "Stringgadget2".

Example: "EasyRequestArgs" Stringgadget2(Offset): "0x024C"

See technical description for more details.

1.10 doc/NerveEX.guide/TASK

3.1.3 Task

This entry defines the task for which the following entries are valid.

Example: "#?" for all tasks or "#?MyName#?" for a special task.

See technical description for more details.

1.11 doc/NerveEX.guide/STRUCTURE

3.1.4 Structure

This entry defines the structure for which the following entries are valid.

Example: "EasyRequest"

This entry is only for clarity. NerveEX doesn't use it. But you have to enter the register, which contains the base address of this structure. This register has to be entered in the Stringgadget2. In contrast to the structure name this entry is mandatory.

Example: "A1"

See technical description for more details.

1.12 doc/NerveEX.guide/MEMBER

3.1.5 Member

This entry defines the member of the above entered structure.

TEMPLATE: [<name>] {(<integer-value>)} [[<name>] {(<integer-value>)} ...]

Example:

The register points to (struct Window *) and we want to reference the title text of the screen:

```
"Window(46)->WScreen.Title(22)"
or
"(46)(22)"
or
"Enterprise(46)NC1701D(22)Warp"
```

1. Window(46) or (46) points to "struct Screen *"
2. ->Screen.Title(22) or (22) points to "UBYTE *Title"

The name between the closing and opening parentheses is not necessary. Incorrect entries are not accepted. The entry cannot be enabled until it is entered correctly.

The "Cycle gadget" defines what type of pointer the entry represents:

- '*String' : Pointer to a string.
- '*Hex' : Pointer to hexadecimal value.

- 'Hex' : The register contains the value itself.

See technical description for more details.

1.13 doc/NerveEX.guide/IDENTIFICATION

3.1.6 Identification

This entry defines the name or hexadecimal value which the above member points to.

See technical description for more details.

1.14 doc/NerveEX.guide/GD_NEW_LEVEL

3.1.7 New Level

Clicking this button creates a new entry. This entry will be inserted as child of the current one.

1.15 doc/NerveEX.guide/GD_ADD

3.1.8 Add

Clicking this button creates a new entry. This entry will be inserted behind the current entry on the same level.

1.16 doc/NerveEX.guide/GD_DEL

3.1.9 Del

The "Del" button deletes the current entry and all its children.

1.17 doc/NerveEX.guide/BOX

3.1.10 Clickbox

Double-clicking the checkbox expands or shrinks all children of this entry. Double-clicking the text enables or disables the entry.

1.18 doc/NerveEX.guide/MN_COPY

3.2.1 Copy

1.19 doc/NerveEX.guide/MN_PASTE

3.2.2 Paste

1.20 doc/NerveEX.guide/MN_FONT

3.2.1 Font

This menu item lets you define the font used for the tree structure.

1.21 doc/NerveEX.guide/MN_TRACE

3.2.2 Trace calls

This menu item opens a new window where you can snoop some debugging information.

See Debug for more details.

1.22 doc/NerveEX.guide/DEBUG

3a. DEBUG

The debugging window shows which tasks call `Open-`, `CloseLibrary` or one of the functions you patched with NerveEX.

There are 3 Check marks you can select/deselect:

- Trace `Open-` and `CloseLibrary` calls
With this check mark enabled you can trace all, oh wonder, `Open-` and `CloseLibrary` calls made by each task
- Trace registers
With this check mark enabled you can watch the register contents during every function call which is patched by NerveEX.
- Trace management stuff
This check mark is not used, but looks good.
- Trace memory
Now you will ask where you can find this check mark?
Search and yea shall find.

All messages are hopefully self-explanatory.

1.23 doc/NerveEX.guide/NERVEEX

4. NERVEEX

NerveEX is the main program which does all the work.

It stays in memory and detects all Open- and CloseLibrary calls made by other tasks. If a library is opened for the first time NerveEX checks if it has to patch functions in this library. If a library is closed for the last time NerveEX removes all patches from this library.

NerveEXPrefs is not required to hide requesters, it is only used to generate the preferences file read by NerveEX, so you only need it if you want to do so.

That's all!

See technical description for more details.

1.24 doc/NerveEX.guide/TROUBLESHOOTING

5. TROUBLESHOOTING

The following functions are NOT allowed to be patched with NerveEX.

```
AllocVec ();
FindTask ();
AllocSignal ();
PutMsg ();
Wait ();
FreeSignal ();
FreeVec ();
```

If you patch these functions, NerveEX will hang (and you will blame me for writing such a bad and useless program).

1.25 doc/NerveEX.guide/QUESTIONS

6. QUESTIONS AND ANSWERS

Question: NerveEX doesn't work on my Amiga xxxx. Why?

Answer: I don't know!
Send me an email and I will answer you.

Question: NerveEX doesn't hide my requester. Why?

Answer: Check if all entries are entered correctly !!!
If you try to hide requesters for a special task, check if the task doesn't change its name. Did you enter the correct

register or structure reference? ...

1.26 doc/NerveEX.guide/TECHNICAL_DES

7. TECHNICAL DESCRIPTION

What are libraries?

Simply speaking libraries are collections of functions, which are accessed through pointers (jump-vectors) in a so-called jump-vector-table. A jump-vector can be made to point to a function other than the original library function. Through this it is possible to change what happens when a library function is called. It is also possible to selectively call the original routine from the new one, which is one of the things NerveEX does.

What is NerveEX's job?

If a task executes a function patched with NerveEX, control is passed to a NerveEX supplied function instead. This function sends a message to NerveEX which contains the "name" of the original function, the current task name, and the contents of the involved registers. NerveEX receives this message and decides, depending on these parameters, if the original function has to be called. After this it signals the calling task if it has to call the original function. That's all, folks.

For more clarity I will give you an example:

Task "A" wants to call the "EasyRequestArgs" function in order to pop up a requester like the following when you leave the program:

```

|_Title_____||_
|
|
| Do you really want to quit? |
|
|      _____      |
|  |__OK__|      |__CANCEL__| |
|_____||_____||

```

Now say you don't want to see this requester anymore when you want to quit.

- 1.) First you have to figure out which library contains this function. In this case it is "intuition.library". Enter this in NerveEXPrefs.

See Library

- 2.) Enter the function name ("EasyRequestArgs") in NerveEXPrefs as
-

child of the library name. The corresponding jump-vector will be automatically scanned by NerveEXPrefs if you have "fd-files" installed. If not and you don't know anything about jump-vectors, libraries or programming on the Amiga stop here and delete me.

See Function

- 3.) OK, now I assume that you know what you are doing. The third thing you have to do is to enter the task name. In this case an "A".

See Task

- 4.) After this you have to find out which register points to the structure where to find the requester text. Here it is register A1 and the structure is EasyStruct. Enter this in NerveEXPrefs as child of the task name.

See Structure

- 5.) The fifth step is to find the structure member which points to the string. Here it is "es_TextFormat" with offset 12. See "Member" to find out how to enter this in NerveEXPrefs.

See Member

- 6.) The last thing to do is to enter the string.

See Identification

From now on with NerveEX started this requester will be banned from your screen, if you didn't make any mistakes.

What if you don't understand this manual:

- I explained it really badly
- You don't understand English
- My English is very bad
- or you don't have the slightest notion about the Amiga

1.27 doc/NerveEX.guide/KNOWN_BUGS

8. KNOWN BUGS

This is the first release and there are no known bugs in this version. This doesn't mean there aren't any unknown bugs. If you find any bugs please mail them to me.

1.28 doc/NerveEX.guide/FUTURE_PLANS

9. FUTURE PLANS

If I get positive response on NerveEX there are several features I want to include in future versions of NerveEX such as:

- Commodity support
- TagList items as ident-text
- Changing the contents of registers before calling the function
- Warp propulsion system
- Holographic environment simulator
- Auto-destruct system
- Replicator system
- Deflector shields
- ...

1.29 doc/NerveEX.guide/ACKNOWLEDGMENTS

10. ACKNOWLEDGMENTS

I would like to thank Martin Apel for providing me with ideas and helping me with problems and Eric Sean Conner(y) for proof-reading the documentation.
