

mui23Edev.readme

COLLABORATORS

	<i>TITLE :</i> mui23Edev.readme		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 8, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	mui23Edev.readme	1
1.1	mui23Edev.README	1
1.2	Introduction	1
1.3	Changes	2
1.4	Filename extensions	2
1.5	The Files	2
1.6	Files: ReadMe.guide	3
1.7	Files: Modules/muimaster.m	3
1.8	Files: Modules/libraries/mui.m	3
1.9	Files: Modules/libraries/Ev31_mui.m	4
1.10	Files: Modules/libraries/Ev31_mui.e	4
1.11	Files: Modules/libraries/muip.m	4
1.12	Files: Modules/libraries/Ev31_muip.m	5
1.13	Files: MacroFiles/mui.ma	5
1.14	Files: MacroFiles/mui.pma	6
1.15	Files: Sources/doMethod.e	6
1.16	Files: Sources/installhook.e	6
1.17	Files: Sources/MUI-Demo.e	7
1.18	Files: Sources/PopUp.e	7
1.19	Problems with MUI and E	7
1.20	Problems: TRUE=1 or TRUE=-1	7
1.21	Problems: Problem with SetAttrsA()	8
1.22	Problems: TAG_IGNORE,0	9
1.23	Copyrights and authors	9
1.24	My address	9
1.25	Future	9
1.26	Bugs	10
1.27	Disclaimer	10

Chapter 1

mui23Edev.readme

1.1 mui23Edev.README

MUI2.3 DEVELOPER FILES FOR AMIGA E

~~~~~

|              |                               |
|--------------|-------------------------------|
| Introduction | What is mui23Edev.lha ?       |
| Changes      | What's new ?                  |
| Files        | Description of the files.     |
| Problems     | Problems with MUI and AmigaE. |
| Future       | New MUI -> new files ?        |
| Bugs         |                               |
| Copyrights   | and authors.                  |
| Disclaimer   | No warranty!                  |

I'm wondering if there is anybody using this files. If you use this files, please think about sending me an e-mail (or maybe a postcard). You don't have to write a long letter, only something like: "I use your mui23Edev-files". Thankyou!

Jan Hendrik Schulz

### 1.2 Introduction

INTRODUCTION

~~~~~

The original files for Amiga-E programmer that came with the developer archive of MUI until now are not very usefull.

The muimaster.m file, that should define all the functions of the muimaster.library, is not up-to-date, and the mui.e file is only a very bad replacement for the mui.h file that C programmer could use!

Thats why I created my own MUI-AmigaE-interface. This archive (as a replacement for the original Amiga-E drawer of the MUI developer archive) contains nearly all that's needed to write MUI-programs in E and some additional informations in this guide!

1.3 Changes

CHANGES

~~~~~

Changes from "mui22Edev.lha" to "mui23Edev.lha"

- New modules for MUI 2.3 and AmigaE v3.1
- Some files removed

Changes from "mui22Edev.lha" to "mui22Edev2.lha"

- Readme-file is now an amigaguide dokument.
- Now ALL struct's of the original mui.h file as OBJECT's in modules available. (see muip.m )
- Special module- and macro-files for AmigaE v3
- The names of the functions in muimaster.m are changed from MuI\_... to Mui\_...!
- Some files removed:
  - "installhook2.e" and "installhook2.m", because it's possible to use the normal installhook instead.
  - "domethod.m", because it's available as "tools/Boopsi.m" with the AmigaE v3 distribution.
- MUIMASTER\_VMIN in mui.m changed from 8 to 9 (In the original mui.h file of mui22dev.lha it's 8, but that's probably wrong!?)
- New set() and nnset() macro definitions -> No setAttrsA variable needed. (Different from the Mac2E v4 definition, see: new set() )

Changes from "mui21Edev.lha" to "mui22Edev.lha"

- New modules for MUI 2.2 (Only "modules/libraries/mui.m" is new)
- New filename-extensions to give a better indication of the filetype
- Some new files
- Sourcecodes etc. changed to use tabs instead of spaces where it's possible, to decrease filesize. (Use an editor or textviewer with tabstops at every 8th column to get correct textformatting!)

## 1.4 Filename extensions

### FILENAME-EXTENSIONS

~~~~~

- .e - An E-sourcecode that could be compiled directly (with AmigaE v3.1).
- .m - An E-module file.
- .ma - This is a macrofile for "Mac2E"
- .pma - And this is a pre-analysed macrofile for "Mac2E" v4.

1.5 The Files

THE FILES

~~~~~

ReadMe.guide

```

Modules/          muimaster.m

Modules/libraries/  mui.m          Ev31_mui.m
                   Ev31_mui.e
                   muip.m          Ev31_muip.m

MacroFiles/        mui.ma          mui.pma

Sources/           domethod.e      installhook.e
                   MUI-Demo.e      PopUp.e

Information about the  Filename-Extensions

```

## 1.6 Files: ReadMe.guide

```

"ReadMe.guide"
~~~~~

```

This file. Please read the entire file before mailing me something like:  
 "This files are creating only errors, what make I wrong?" There are some  
 special problems with MUI and Amiga-E which can cause big trouble :-(

## 1.7 Files: Modules/muimaster.m

```

"Modules/muimaster.m"
~~~~~

```

This file defines the functions of the muimaster.library. Copy it to  
 EMODULES: and to use it write in your source:

```
MODULE 'muimaster'
```

ATTENTION: With "mui22Edev2.lha" I changed the names of the functions from  
 MuI\_... to Mui\_... (Lionel Vintenat (Mac2E) told me that Mui\_... looks  
 nicer and I think he's right.) Maybe you have to change your old sources  
 to use this file.

## 1.8 Files: Modules/libraries/mui.m

```

"Modules/libraries/mui.m"
~~~~~

```

You need this file only if you still use AmigaE v2.1. If you use the  
 new AmigaE v3.1 then use the file Modules/libraries/Ev31\_mui.m !

This file contains some of the #define's and struct's of the mui.h file  
 that came for C programmer with the developer archive of MUI 2.3. Use the  
 C mui.h file with its comments to get more informations about the #define's  
 and struct's.

---

In this E-module-file all CONSTs and the structs (OBJECTs) which are of interest for E programmer, are defined. To see which structs are in this module, use the ShowModule-tool that came with the E compiler. Therefore all OBJECTs are at the beginning of the module.

The CONSTs are written exactly the same way as in the mui.h file. The identifiers of the OBJECTs are written totally lowercase!!

Copy this file to "EMODULES:libraries/" and to use it write in your source:

```
MODULE 'libraries/mui'
```

See also: Modules/libraries/muip.m and Modules/libraries/Ev31\_mui.m .

## 1.9 Files: Modules/libraries/Ev31\_mui.m

```
"Modules/libraries/Ev31_mui.m"
~~~~~
```

This is the AmigaE v3.1 version of mui.m . There are three differences with this file:

- The OBJECTs are using the new features ("PTR TO xyz" is defined as "PTR TO xyz" and not as "LONG" etc.)
- The \_\_dummyxfc2\_\_ OBJECT is included.
- All macros of MacroFiles/mui.ma and some more (see Ev31\_mui.e ) are included ("MacroFiles/mui.ma" and "Mac2E" are now obsolete!)

If you have the new AmigaE v3.1, and if you want to use this file, then copy it to "EMODULES:libraries/" and rename it to "mui.m". Don't forget to add "OPT PREPROCESS" to the source, if you want to use the macros!

## 1.10 Files: Modules/libraries/Ev31\_mui.e

```
"Modules/libraries/Ev31_mui.e"
~~~~~
```

This file is the source of Modules/libraries/Ev31\_mui.m . I included this file, because ShowModule don't shows the bodys of macros :-(

So, if you want to use the macros in "Modules/libraries/Ev31\_mui.m" with AmigaE v3.1, you could look at this file to see, which macros are available. (The macros are all at the end of the file.)

The contents of this file is nearly identical with the original C mui.h file, but without the comments.

## 1.11 Files: Modules/libraries/muip.m

---

```
"Modules/libraries/muip.m"
```

```
~~~~~
```

This e-module contains all the MUIP\_...-structs of the original C mui.h file as OBJECT's. I think, this OBJECT's are not very usefull in E, that's why I didn't include them in former releases. But Lionel Vintenat (Mac2E) told me, that one reason for including his own MUI-files with Mac2E v4 was my incomplete mui.m file.

OK, now ALL struct's are available as OBJECT's, but not all in one file! The (I think) very useless MUIP\_... struct's are in this extra module. The mui.m module is big enough, even without this OBJECT's! If you really need them, write

```
MODULE 'libraries/muip.m'
```

in your source. (And please send me a mail to tell me the reason! I know absolutly no situation where this OBJECT's are usefull in E!!)

See also: Modules/libraries/Ev31\_muip.m

## 1.12 Files: Modules/libraries/Ev31\_muip.m

```
"Modules/libraries/Ev31_muip.m"
```

```
~~~~~
```

This is a special version of muip.m for the new AmigaE v3.1. There is only one difference:

- The OBJECTs are using the new features ("PTR TO xyz" is defined as "PTR TO xyz" and not as "LONG" etc.)

If you have the new AmigaE v3.1, and if you want to use this file, then copy it to "EMODULES:libraries/" and rename it to "muip.m".

## 1.13 Files: MacroFiles/mui.ma

```
"MacroFiles/mui.ma"
```

```
~~~~~
```

You need this file only, if you still use AmigaE v2.1 and Mac2E. If you use the new AmigaE v3.1 then use Modules/libraries/Ev31\_mui.m .

This file contains the most macro definitions of the C mui.h file which are not simple CONSTs and because of that not in the mui.m module. Only "the most", because some of them couldn't be translated to E v2.1b, but they are not very important.

To use this file and its macrodefinitions you need the program Mac2E , and for more informations about the macros, have a look at the comments of the mui.h file that came with MUI.



The macronames are written the same way as the C originals in mui.h - with one exception: The macro "String()" is renamed to "StringMUI()" to avoid conflicts with the E-Function String()

Copy this file to ... where you want it - for example in a MacroFiles-directory in your Amiga-E-directory.

See also: MacroFiles/mui.pma

## 1.14 Files: MacroFiles/mui.pma

```
"MacroFiles/mui.pma"
~~~~~
```

This is the pre-analysed version of the macrofile MacroFiles/mui.ma for direct use with Mac2E v4.0.

## 1.15 Files: Sources/doMethod.e

```
"Sources/doMethod.e"
~~~~~
```

To write MUI programs, you need a DoMethod function. Because this function is part of the amiga.lib, which we don't have as E programmer, you have to use your own one, or better use this one written by Wouter van Oortmerssen.

Add it into your source to use it and have a look at the demo-source Sources/MUI\_Demo.e to see, how to use it.

If you use AmigaE v3.1, you can find this function in the emodule "EMODULES:tools/Boopsi.m" which comes with E v3.1, too.

## 1.16 Files: Sources/installhook.e

```
"Sources/installhook.e"
~~~~~
```

Sometimes - for example if you use PopUps - you need hooks. This source gives you a PROC that makes it very easy to install the needed hook-structures. Look at this source (the comments) and the demo-source Sources/PopUp.e to see, how to use it.

This source (without the comments) and an E-module of it can also be found in the AmigaE v3.1 distribution. It's only still included for users of AmigaE v2.1b.

NOTE: Normaly a hookfunktion has three arguments. But the first one, a ptr to the hookstructure, is (in most cases) not very interesting. I'd never needed this pointer and so I changed the installhook-source. This changed

---

source was included in the mui22Edev.lha as installhook2. But now I noticed that it's possible to use the original installhook-source with only two argumented hookfunctions, too. Why? Installhook uses an assembler-interface to call the hookfunction. One thing this interface does, is putting the 3 arguments on the stack \*and\*remove\*them\* after the hookfunction-call. It's no problem if a hookfunction like "PROC hf(m,o)" only uses the last two arguments on the stack (without the ptr to the hookstructure) and not all three. (Have a look at the comments in installhook.e)

## 1.17 Files: Sources/MUI-Demo.e

"Sources/MUI-Demo.e"  
~~~~~

This is a translation of the C demo-source "MUI-Demo.c" that came with MUI v2.1. Use it to see, how to write MUI-programms in E and where are the differences between C and E. This source contains macros: Mac2E or the new AmigaE v3.1 is needed to compile this source (See the comments in the source).

See also: Problems

## 1.18 Files: Sources/PopUp.e

"Sources/PopUp.e"  
~~~~~

This is a translation of the demo-source "PopUp.c" that came with MUI2.1. Use it to see, how to use the installhook function.

## 1.19 Problems with MUI and E

PROBLEMS WITH MUI AND E  
~~~~~

|             |                                     |
|-------------|-------------------------------------|
| MUI_TRUE    | Why a CONST MUI_TRUE?               |
| SetAttrsA() | Problems with set()                 |
| TAG_IGNORE  | Why "TAG_IGNORE,0," in some macros? |

## 1.20 Problems: TRUE=1 or TRUE=-1

TRUE=1 or TRUE=-1  
~~~~~

In C TRUE has the value 1 but in E TRUE has the value -1. That can cause problems, therefore a CONST "MUI\_TRUE" with value 1 is defined in Modules/libraries/mui.m . Use this one instead of TRUE whenever you want to give TRUE to MUI. See "Sources/MUI-Demo.e" and "Sources/PopUp.e".

---

## 1.21 Problems: Problem with SetAttrsA()

Problem with SetAttrsA()  
 ~~~~~

Whenever you try to set an attribut to the value it allready has, MUI overwrites this attribut in the taglist with TAG\_IGNORE to make shure that notify-class don't react on it. This is nessesary to prevent endless-notification-loops. As long as you are a C programmer and use SetAttr() that is no problem, because MUI then changes only a *\*copy\** of the original datas of the SetAttrs()-call. But if you use SetAttrsA(), as we must do because SetAttrs() is a function of amiga.lib, it is a problem! You only give a PTR to the original-datas with SetAttrsA() and now MUI changes this *\*original\** datas. If you then uses this datas again at the next execution of the same SetAttrsA()-call, the datas are still changed and nothing (TAG\_IGNORE) will happen :-(

One possible way to avoid this problem is, to make the attribut a non-constant data. Instead of

```
SetAttrsA(obj, [MUIA_..., value, ..., TAG_DONE])
```

write

```
SetAttrsA(obj, [Eval('(MUIA_...)), value, ..., TAG_DONE])
~~~~~
```

Now everytime this SetAttrsA()-call is executed, the attribute is again "evaluated" and stored in the list.

BUT: x:=[Eval('(MUIA\_...)),...]; SetAttrsA(obj,x); SetAttrsA(obj,x) has *\*NOT\** the wished result, because the non-constand datas of the LIST are only evaluated when x is assigned to to the LIST but not everytime this PTR to the LIST in x is used later!

To avoid the SetAttrsA()-problem the macros set() (and nnset()) in the files MacroFiles/mui.ma and Modules/libraries/Ev31\_mui.m are defined like this:

```
#define set(obj,attr,value) SetAttrsA(obj,[Eval('(attr)),value,TAG_DONE])
```

The global variable "setAttrsA" that was needed with the set() and nnset() of former releases is now obsolete.

INFO: You should use THIS set() and not the one that came with Mac2E v4, because that one can cause problems if you write something like:

```
set( func_getobj(), MUIA_xyz, value ) or
set( obj_array[]++, MUIA_xyz, value ) or
set( x:=func(), MUIA_xyz, value )
```

## 1.22 Problems: TAG\_IGNORE,0

TAG\_IGNORE,0  
~~~~~

Some of the macros in MacroFiles/mui.ma and Modules/libraries/Ev31_mui.m are ending with "[TAG_IGNORE,0,". That seems to be superfluous but it is needed, because in AmigaE v2.1b if you want to split a statement over several lines, you can't end a line with "[" but with a comma. And with AmigaE v3.1 it's needed to allow the same use of the macros.

1.23 Copyrights and authors

COPYRIGHTS AND AUTHORS
~~~~~

Jan Hendrik Schulz (that's me): - this ReadMe guide  
 ( my address ) - all files in Modules/ and MacroFiles/  
                   (created from the original C mui.h file)  
 - translations of MUI-Demo and PopUp to E

Stefan Stuntz: - MUI with mui.h and the original C sources  
                   of MUI-Demo.e and PopUp.e

Lionel Vintenat: - Mac2E

Wouter van Oortmerssen: - AmigaE  
                   - doMethod.e  
                   - installhook.e (comments added by me)

## 1.24 My address

MY ADDRESS  
~~~~~

Snail mail: Jan Hendrik Schulz
 Reinhard-Raffalt-Str. 41
 94036 Passau
 Germany

Internet: schulzj@fmi.uni-passau.de

1.25 Future

FUTURE
~~~~~

With future versions of MUI this files will be included, I hope!  
 And if Stefan forget me again, there will be an archive like this!

---

If you have some ideas to make things better, please inform me about them!

## 1.26 Bugs

BUGS  
~~~~

The PupUp-example hangs if you try to close it while there are still some ASL-PopUps open. I don't know why! If anybody sees the bug, please send me a mail.

If you find a bug in the macrofiles, the modules or this textfile, please send me a mail to give me the chance to remove it.

1.27 Disclaimer

DISCLAIMER
~~~~~

This files are provided "AS IS" without warrenty of any kind, expressed or implied! I'm NOT liable to you for damages or problems, including any general, special, incidental or consequential damages or problems arising out of the use or inhability to use of the files. Including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of a program build with this files.

(I hope you know what I want to say! My english is not good enough to write such legal stuff!)

---