

MUIMenu

Giuseppe Chiesa

Copyright © CopyrightÂ©1994 Giuseppe Chiesa

COLLABORATORS

	<i>TITLE :</i> MUIMenu	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY	Giuseppe Chiesa	March 1, 2022
<i>SIGNATURE</i>		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MUIMenu	1
1.1	Introduction	1
1.2	Description	1
1.3	Requirements	2
1.4	Copyrights and Contact	2
1.5	Thanks	3
1.6	Install	3
1.7	Usage	4
1.8	Text Gadget	5
1.9	Edit Buttons	5
1.10	Other Buttons	5
1.11	Edit Window	6
1.12	Sync VS ASync	6
1.13	Configuration File	7
1.14	Source	7
1.15	Known Bugs and Bug Report	8
1.16	Future	8
1.17	About MUI - MagicUserInterface	8

Chapter 1

MUIMenu

1.1 Introduction

Introduction to MUI-Menu The Pop Up Menu Tool Manager

Welcome to MUIMenu 1.00

[About MUI - MagicUserInterface](#)

[Description](#)

[Requirements](#)

[Copyrights and Contact](#)

[Thanks](#)

[Installing](#)

[Using MUIMenu](#)

[Source](#)

[Known Bugs and Bug Report](#)

[Future](#)

1.2 Description

Description

MUIMenu creates a configurable Pop-Up Menu with some labelled buttons and lets you add, delete or change any of these objects to create your own custom menu.

After that you have just to press a button to execute the corresponding program.

MUIMenu lets you save the created menu and reloads the [Configuration file](#) every time it is loaded.

When you create a new menu button you have only to specify button label, the program to be executed, its parameters and if you want to execute it

synchronously or asynchronously.

(see [Sync VS Async](#))

You can build a menu with max 48 ITEMS : 12 rows x 4 cols

1.3 Requirements

Requirements

All you need is a wonderful machine called Amiga and WB2.0+

Of course you need MUIMaster.library (2.0+) and MUI Classes

MUI - MagicUserInterface is available through Aminet or in any PD BBS.

(see [MUI - MagicUserInterface](#))

1.4 Copyrights and Contact

Copyrights

Idea, Design and Code by Giuseppe Chiesa 1994

Based on the original program by Claudio Fuse' aka DBL : DBLMenu

This program is GiftWare, you can use it with no restrictions but if you like it I suggest you to send me a little gift...

the more appreciated are postcards, bug reports and programming suggestion (new ideas and critics are always welcome)...

and if you have nothing to do and by accident you paint a new icon for this program feel free to send me your masterpiece.

If you want to spread this program you're free to do it with the only restriction of spreading the entire MUIMenu.lha archive leaving it as it is and without altering its contents.

NOTE:

There's no warranty that there are no more bugs present in this program. Please remember that you use this product at your own risk and that the author cannot be liable responsible of any damage,direct or indirect,that the use of this program may produce.

You are not allowed to sell this program or to change anything in it and in its package. You are allowed to use part of the source included in the archive but you do it at your own risk.

Contact

If you want to contact me for any question or critic please feel free to

reach me at ...

E-Mail:

chieg@ghost.sm.dsi.unimi.it

or (for postcards and any other gift)

Address:

Giuseppe Chiesa

via Orombelli, 18

20131 - Milano

ITALY

1.5 Thanks

Thanks

I want to thank stefan Stuntz for his wonderful MUI,

I've never seen anything so easy and so powerful at the same time.

Thanks to Claudio for his support and his idea.

Thanks to my sweet love Francesca because she puts up with me,
to my parents for everything they do for me and I shall never be able to
repay enough.

To my aunt Ada to whom I dedicate this little work...

...We all miss you...Thank you for being my aunt

And last I want say hello to Furio, I still repeat the same old thing...

...try using MUI...try using MUI...try using MUI...try using MUI...

[to be continued]

Thanks also to everyone who gives a try to this program and to whom
contacts me even just to tell me what he thinks about MUIMenu or say
hello... bye !!

1.6 Install

Installing MUIMenu

If you have MUI already installed you have just to copy MUIMenu and the
AmigaGuide documentation in the same drawer and then you're ready to
start.

I suggest you to configure MUIMenu with MUI-Prefs so to control it with a
hot-key, to create your menu and to put it in the WBStartup drawer.

1.7 Usage

Usage

When you start MUIMenu for the first time you should see the main window with the only editing buttons and without any executable program.

Now you can add an ITEM to the menu using **ADD** , this will open the **Edit Window** and will let you insert all necessary data.

You have to insert item name, program to be executed when button is pressed and all the necessary parameters for the specified program; you can also decide if the program must be executed Asynchronously or Synchronously.

(See **Sync VS ASync**).

N.B.

ITEM max chars.

Name 16

Program 32 (with path)

Parameters 48

Press **OK** to accept or **CANCEL** to cancel ITEM insertion.

If you press **OK** you'll get a new ITEM in the main window and if you press it the corresponding program will be executed followed by the specified parameters.

To obtain the same results you can press the key near the label button.

It is a keyboard shortcuts for that ITEM, please note that letters start from 'b' and not from 'a' because it can be more useful to keep 'a' for **ADD** use.

Proceeding this way you can add new elements and create your own menu.

Pressing **SAVE** you can save your menu in the **Configuration File**

called "MUIMenu.data" in S: drawer so that when you reload MUIMenu it can create your menu by reading its **Configuration File** .

If you want you can also **CHANGE** an ITEM without having to **DELETE** it and then **ADD** it again.

Press **QUIT** or Close Window to quit program.

There is also a **Text Gadget** in the main window, refer to it for some useful information.

I suggest you to configure your application so that it can be controlled with a hot-key using MUI-Prefs program, this will let you have it under your control everytime you need to execute a program.

NOTE:

Closing the window doesn't quit the program but iconifies MUIMenu.

1.8 Text Gadget

Text Gadget

There is a text gadget in the main window, it gives you information about what's going on.

It is mainly useful when something goes wrong during **Configuration File** loading.

In this case look, with **CHANGE** button, to the last item loaded and look if everything's ok.

This problem doesn't exist when you use **SAVE** button to save **Configuration File** , it may happen only if you make an error in the config file editing it manually

...but why use an editor when the program itself lets you automatically create your own menus ?

1.9 Edit Buttons

Edit Buttons

ADD

This Button opens the **Edit Window** and lets you add a new item to the menu.

DELETE

This button lets you choose the element to delete and then kills it.

Press **ABOUT** button to Cancel action.

CHANGE

This button lets you choose the element to change and the open the **Edit Window** to let you change the ITEM attributes.

Press **ABOUT** button to Cancel action.

1.10 Other Buttons

Other Buttons

QUIT

This button lets you quit the application.

MUIMenu tells you when you're quitting without saving all changes.

I suggest you to use a hot-key to make the use more comfortable.

SAVE

This button saves the menu to the **Configuration File** "S:MUIMenu.data"

ABOUT

This button opens a MUI-Requester containing some useful information.

This button is also used to cancel **DELETE** or **CHANGE** operation.

Keyboard ShortCuts

On the right side of each button there's a label button containing the keyboard shortcut for that ITEM.

1.11 Edit Window

Edit Window

This window lets you introduce ITEM name, program to be executed, its parameters and execution type (See **Sync VS ASync**).

Key Shortcuts are available and a popup file requester is provided for program selection.

Buttons

OK

--

Accept ITEM and adds or changes it in the menu.

CANCEL

Cancel adding or changing operation.

1.12 Sync VS ASync

Synchronous VS Asynchronous execution

When you execute a program (child) from inside another one (father) the father usually waits for its child to finish, this is Synchronous execution.

This means that if the child program doesn't detach itself from MUIMenu, who is of course the father of all its children(the menu ITEMS) you can't press another button and execute another program until the first one is ended.

So you can choose ASync in the **Edit Window** to

automatically detach the program from MUIMenu and let it execute another one when a button is pressed.

Actually, MUIMenu doesn't even let you press **Edit Buttons** or **Other Buttons** if the last program is executed

Synchronously and it doesn't detach from its father.

N.B.

Remember that MUIMenu memorize each button pression and replies after the last program has finished execution if it can't do it before.

1.13 Configuration File

Configuration File

The Configuration File is an ASCII file containing a title :

MUIMenu Data File

an end :

END MUIMenu

and, in the middle, each menu ITEM data preceded by ITEM keyword :

ITEM <-- ITEM keyword

Cray Emulator <-- Button Label

Dh1:Emulators/CrayII <-- Program Name with complete path

-nottoofast <-- Parameters

1 <-- Asynchronous execution (0 for Sync)

N.B.

Indention is just for clarity

1.14 Source

Source

In the archive you can also find the source code...

Well I know it is not the best code you've ever seen but I have no time to rewrite some functions.

So look at it if you're interested in MUI application generation and dynamic creation of gadget.

Feel free to report any problem or critic!!

1.15 Known Bugs and Bug Report

Known Bugs and Bug Report

Well...I haven't found any bug yet so please tell me if you find something wrong.

Bug-Report to

E-Mail:

chieg@ghost.sm.dsi.unimi.it

1.16 Future

Future

- Maybe a different positioning for new buttons when a column is finished

- ... tell me

E-Mail:

chieg@ghost.sm.dsi.unimi.it

1.17 About MUI - MagicUserInterface

This application uses

MUI - MagicUserInterface

(c) Copyright 1993 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXX.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 20.- or US\$ 15.-

to

Stefan Stuntz

Eduard-Spranger-Straße 7

80935 München

GERMANY

NOTE:

Try programming using MUI...and you'll never give it up.

Thanks Stefan.
