

AppMenu

COLLABORATORS

	<i>TITLE :</i> AppMenu		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		November 24, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AppMenu	1
1.1	AppMenu Guide	1
1.2	Copyright	1
1.3	Introduction	2
1.4	Installing and running	2
1.5	Using	3
1.6	Gadget: Menu Items	4
1.7	Gadget: Command	4
1.8	Gadget: '>' - Choose command	4
1.9	Gadget: Arg	4
1.10	Gadget: New	5
1.11	Gadget: Del	5
1.12	Gadget: Save	5
1.13	Gadget: Use	5
1.14	Menu: Project/Quit; Gadget: Cancel	5
1.15	Menu: Project/Open...	5
1.16	Menu: Project/Save As...	5
1.17	Menu: Edit/Cut	6
1.18	Menu: Edit/Copy	6
1.19	Menu: Edit/Paste	6
1.20	Menu: Edit/Move Up, Edit/Move Down	6
1.21	Menu: Edit/Last saved	6
1.22	Menu: Edit/Restore	6
1.23	Menu: Options/Create icons?	6
1.24	Error messages	6
1.25	Author and programs	7
1.26	History and what's new	8

Chapter 1

AppMenu

1.1 AppMenu Guide

AppMenu - Version 2.3 (3.4.95)

Copyright © 1993-1995 Marius Chwalczyk.
All Rights Reserved.

Copyright
Introduction
Installing and running
Using
Error messages
Author and programs
History and what's new

1.2 Copyright

The package "AppMenu - Version 2.3" is Copyright © 1993-1995 by Marius Chwalczyk. All Rights Reserved.

It's NOT ALLOWED to sell this package. It can be freely distributable as long as:

1. only a moderate charge (copying, data medium price) is allowed.
2. all of the files listed above must be included in their original form without modifications of any kind.

AppMenu	11628 - preferences command (WB 3.x)
AppMenuCx	4828 - program (WB 3.x)
Copyright-ReadMe	1607 - this file
Copyright-LiesMich	1856 - this file in german
AppMenu.guide	17305 - english documentation
AppMenu_D.guide	22090 - german documentation
Workbench_2.x	(dir) -
AppMenu	12352 - > versions for Workbench 2.x

AppMenuCx	5480	-	
AppMenu.info	1052	-	
AppMenuCx.info	860	-	
Copyright-ReadMe.info	835	-	
Copyright-LiesMich.info	835	-	> icon files
AppMenu.guide.info	840	-	
AppMenu_D.guide.info	840	-	
Workbench_2.x.info	628	-	
AppMenu.info	1052	-	
AppMenuCx.info	860	-	

3. the package can be packed (e.g. with LhA or Lharc), but no crunching of executable files is allowed.

If this package will be included in PD series, please inform me (the author) about it. Thanks for a free copy of the distribution!

This program is "as-is", all use is at your own risk.

1.3 Introduction

The package AppMenu have two programs. Both run only under Release 3 (V39) and later versions of the AmigaOS. There are versions for Release 2 (V37/38) in "Workbench_2.x" drawer too.

The one AppMenuCx is a commodity that adds menu items to the Workbench's Tools menu. When a menu item is activated, AppMenuCx starts a command. The command is started as separate, asynchronous processes. You can also choose its arguments through icons or file requester.

All menu items and commands can be free defined with the help of the second program AppMenu . This program works like the others preferences programmes.

1.4 Installing and running

Copy both programs AppMenuCx and AppMenu in directories you choosed. Best place first in WBStartup drawer, second in Prefs drawer.

Two preferences files will be used too (like by system prefs): "ENV:AppMenu/AppMenu.prefs" and "ENVARC:AppMenu/AppMenu.prefs". You must create both directories with "MakeDir ENV:AppMenu" and "MakeDir ENVARC:AppMenu".

The programs using the language lcalization from release 2.1 of AmigaOS. If one of the included catalog files is for the selected language, it can be coped in one of the directories:

"LOCALE:Catalogs/<name of language>/" or
 "<dir of the program>/Catalogs/<name of language>/".

Both programs can be started from Workbench or from Shell.

AppMenuCx is a commodity and uses only one Shell argument or one icon's tool type: CX_PRIORITY=<number>. Template is CX_PRIORITY/K/N. Default number is 0. See Workbench handbook about commodities.

AppMenu have following Shell arguments: FROM=<filename>, EDIT, USE, SAVE and CLIPUNIT=<clip unit num>. Template is FROM,EDIT/S,USE/S,SAVE/S,CLIPUNIT/K/N. '<filename>' is the preferences file (default "ENVARC:AppMenu/AppMenu.prefs"). Default '<clip unit num>' is 0 (primary clipboard). Use this arguments like in Workbench's preferences programs.

AppMenu uses following tool types too:

CLIPUNIT=<clip unit num> see above.

SAVEICONS=<NO|YES> is used for state of the Create Icons? menu item. Default is YES. By Shell start the menu shows NO.

SCREENFONT=<NO|YES> says if AppMenu should use the preferences Screen Font instead of "Topaz/8". Default is YES. It's always used, also by Shell start.

1.5 Using

AppMenu creates a file with description of the menu items. AppMenuCx needs it to make items in the Tools menu and to starts the commands.

AppMenu's window presents a list gadget with its scroller and a string field below, a separate string field, a cycle gadget and six buttons. It have a menu too.

There are following gadgets:

```

Menu Items:      ' list      '
                  ' of        '
                  'menu items'

New      Del      'menu item'

Command:          Arg:  'arguments'

>                'command'

Save           Use           Cancel

```

And following menu items:

```

Project          Edit          Options
Open...  'O'      Cut          'X'      Create icons? 'I'
Save as... 'A'    Copy        'C'
About...  '?'    Paste        'V'
~~~~~
Quit      'Q'    Move Up      'U'
                  Move Down  'D'

```

```
~~~~~
Last saved 'L'
Restore    'R'
```

AppMenuCx has no window, but you can control it like others commodities with the "Exchange" program. AppMenuCx uses always the newest "ENV:AppMenu/AppMenu.prefs" file.

1.6 Gadget: Menu Items

This listview gadgets show the item list. To change a item just select it. Now you can update the Command text. Press Return or Tab key or change Arg gadget to activate the string field below the listview gadget. Here you can modify the item text and press Return to take the item back to the list.

1.7 Gadget: Command

Type the command and its arguments you want to execute. Note that you must specify the full path of the command when you start AppMenu from Workbench. It knows "C:" path only in this case. You can solve this problem when you add this line in "S:User-startup": "ASSIGN C: <path_of_the_cmd> <...> ADD".

The arguments will be added at the end of the command. If it needs them on the other place, mark it with ' [] ' like by Alias command.

It's possible to type several files when you separate them with <Ctrl LCommand J>. All programs will be started synchronous. It's made internal through the DOS function System() so you can use it by 'Execute Command ...' or in Shell too.

1.8 Gadget: '>' - Choose command

Show file requester. You can choose new text for Command .

1.9 Gadget: Arg

Select if and from where you want arguments to a command. If one of the string gadgets is active you can press <RCommand G> and <RCommand Shift G> to select it. The possibilities are:

None	Execute command without extra arguments.
Requester	If you activate the item a file requester appears and you can choose a file name as an argument to the command. You can cancel the execution too.
Icons	The names of selected icons in Workbench (if any) will be

arguments.

Icons/Request Same as above. If no icon selected as 'Requester'.

1.10 Gadget: New

Add a new item to the list. Just type a Command text, choose Arg and write a menu item text. To quick add a program just drop its icons in the Menu Items gadget. To add a new item before an existing item in the list simple select this item first.

The number of items you can add is limited only by available memory, the system however creates current only 63 selectable menu items.

1.11 Gadget: Del

Remove the selected item from the list.

1.12 Gadget: Save

Save the item list to "ENVARC:AppMenuCx/AppMenu.prefs". Further as Use .

1.13 Gadget: Use

Save the item list to "ENV:AppMenuCx/AppMenu.prefs" and quit AppMenu.

1.14 Menu: Project/Quit; Gadget: Cancel

Quit AppMenu. If the changed items are not saved a requester appears.

1.15 Menu: Project/Open...

Load new items from a selected file. If the changed items are not saved a requester appears. You can also drop a file icons in the window (except in Menu Items gadget).

1.16 Menu: Project/Save As...

Save the item list to a selected preferences file.

1.17 Menu: Edit/Cut

Copy the text of a item to clip like with `Copy` and delete the item from the list.

1.18 Menu: Edit/Copy

Copy the text of a item to clip. It will be all saved in the form: `'<command-text>;<arguments-code><item-text>'`. So you can delete this item and add it back to the list at other place with `New` and `Paste`. You can use it in Shell (note `'`) too.

1.19 Menu: Edit/Paste

Change current command with a text from clip. If the text has the form like in `Copy` all will be changed.

1.20 Menu: Edit/Move Up, Edit/Move Down

Move the selected item in the list by one up or down.

1.21 Menu: Edit/Last saved

Load the at last with `Save` saved items.

1.22 Menu: Edit/Restore

Undo all changes in the list since the programm starts.

1.23 Menu: Options/Create icons?

Says if create an icon for file from `Save As ...`.

1.24 Error messages

The programs can make following error messages.

AppMenu:

- Could not activate appwindow
No Paste with icon placed in window. Continuation.
- Could not open file requester
Could not open library: asl.library (V37) or not enough memory for file requester. Chosed function breaks. Continuation.
- Could not open window
Not enough memory for window. Quit.
- Error coping to Clip
- Error pasting from Clip
Could not open Clipboard or an error by read or write. Copy or Paste breaks. Continuation.
- Error loading IFF file <name>
Error loading file or not enough memory for new preferences. Cont.
- Error saving IFF file <name>
Error saving file. Continuation.
- Invalid resident library
Version for WB 3.x was started on 2.x or could not open one of the libraries: intuition.library (V37), gadtools.library (V37), workbench.library (V37), icon.library (V37), iffparse.library (V37). Quit.
- Not enough memory available
Not enough memory for program or choosed function. In first case program quits, in second the funktion breaks.

AppMenuCx:

- Command line too long
Chosed arguments makes the command too long. System set max lenght at 512. The command will not be executed.
- Could not create all menu items
Not enough memory to create all menu items. Continuation.
- Could not create CxObject
Could not make the program to be Commodity. Quit.
- Could not open file requester
Could not open library: asl.library (V37) or not enough memory for file requester. Chosed function breaks. Continuation.
- Could not setup notification on file "ENV:AppMenu/AppMenu.prefs"
Not enough memory or problems with the file. Quit.
- Error loading IFF file "ENV:AppMenu/AppMenu.prefs"
Error loading file or not enough memory for new preferences. Quit.
- Invalid resident library
Version for WB 3.x was started on 2.x or could not open one of the libraries: intuition.library (V37), commodities.library (V37), workbench.library (V37), icon.library (V37), iffparse.library (V37). Quit.
- Not enough memory available
Not enough memory for program or choosed function. In first case program quits, in second the funktion breaks.

1.25 Author and programs

If you have some comments, additional ideas, bug reports, or you want to

reward my work feel free to send me a letter or to call me. (English isn't my native language, but I do my best.)

You can reach me via

Mail: Marius Chwalczyk
Zigelstr. 59/165
D-67655 Kaiserslautern
Germany

Phone: 0631-10944

AppMenu and AppMenuCx are written in C and compiled with SAS/C 6.55. They are tested on an Amiga 3000 (2.0 and 3.1) with "Enforcer" and "Mungwall".

If you want to create catalogs file of your language use the included catalog description files. Please take care that the Texts with the standard font "Topaz/8" in all gadgets fit. I'm ready to include your translation in the next release.

The german catalog file was create with help of the program "KitCat 1.1" by Albert Weinert.

1.26 History and what's new

NEW = something new added, CHG = something changed, FIX = bug fixed

2.3 (3.4.95)

- NEW: two versions of programmes, for WB 2.x and 3.x
- CHG: using memory pool functions: V37 amiga.lib, V39 exec
- CHG: german catalogs

AppMenu:

- CHG: appwindow behaviour changed: to add new item drop icon at listview gadget, to load prefs drop its icon elsewhere
- CHG: GUI full font sensitive now
- CHG: menu shortcuts and texts like by OS
- CHG: "Create Icons?" defaults to YES if started from WB, and to NO if from Shell
- NEW: reject icons in file requester
- NEW: "Edit/Cut" menu item and CLIPUNIT/K/N argument
- NEW: CTRL-F is window "wake up" key
- FIX: NewLookMenu problem (tag was in CreateMenus(), now correct in LayoutMenus())

2.2 (31.8.94)

- nothing new, but ...

Here some technical explanations: I'm rather sure that both programs version 2.1 can crash the 68000 machines. I have an '030 in my A3000 so I can test it. There can be a 80000003 Guru ("Illegal address access (ie: odd)"). The programs can make word (UWORD) access of an odd address. On my computer it's OK but on the 68000 it's illegal!

When I made localization I used the first two bytes (one word) of the strings for index number. Like this: "ID = *(UWORD*)STR_ID; STR = STR_ID + 2;". I find out now that the compiler (SAS/C 6.51) places the strings like

"static const UBYTE str[];" on odd addresses too. For a string (byte array) it's OK but with a word there are problems. Now I use only the first byte for the index.

I had the idea with word index from a program "Cat2H" by Niko François. It was good but not complete :-). So be warned if you're using this program!

2.1 (18.7.94)

- NEW: both programs localized
- NEW: german catalog file included
- CHG: small changes in texts and size of window and gadgets

2.0 (14.5.94)

- Programm splitted in AppMenuCx (Tools menu) and AppMenu (prefs)

AppMenuCx:

- NEW: place holder ' [] ' to mark where to place arguments
- NEW: multiple select and pattern matching options in file requester

AppMenu:

- FIX: gadgets ('New') and menus working correct now.
- CHG: both CREATEICONS and SCREENFONT are default YES and only as tool types, but used if started from Shell too.
- CHG: 'Copy' uses all texts; 'Paste' doesn't create new items, but changes the current one (use 'New' first).
- NEW: menu multiselect support and more menu shortcuts.
- NEW: menu items are execable via shortcuts from active string gadgets.
- NEW: if string gadgets active press <RAMiga G> for next and <Shift RAMiga G> for previous 'Arg'.
- NEW: gadget for file requester to select 'Command'.

1.x (25.3.93 - 4.7.93)

[old text removed]
