

ScreenMenuEd

Nick Christie

Copyright © ScreenMenuEd.guide1.0 Copyright ©1995 Nick Christie

COLLABORATORS

| | | | |
|------------|-------------------------|---------------|-----------|
| | TITLE : ScreenMenuEd | | |
| ACTION | NAME | DATE | SIGNATURE |
| WRITTEN BY | Nick Christie | July 20, 2024 | |

REVISION HISTORY

| | | | |
|--------|------|-------------|------|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

Contents

| | | |
|----------|---|----------|
| 1 | ScreenMenuEd | 1 |
| 1.1 | ScreenMenuEd.guide | 1 |
| 1.2 | ScreenMenuEd: Introduction | 1 |
| 1.3 | ScreenMenuEd: Requirements | 1 |
| 1.4 | ScreenMenuEd: Using the Program | 2 |
| 1.5 | ScreenMenuEd: ScreenMenuEd's Window | 2 |
| 1.6 | ScreenMenuEd: ScreenMenuEd's Menu | 3 |
| 1.7 | ScreenMenuEd: Edit Menu Definition Window | 3 |
| 1.8 | ScreenMenuEd: Menu Definition Rules | 4 |
| 1.9 | ScreenMenuEd: Tooltypes and Shell Arguments | 4 |
| 1.10 | ScreenMenuEd: MENUFILE Keyword | 5 |
| 1.11 | ScreenMenuEd: PUBSCREEN Keyword | 5 |
| 1.12 | ScreenMenuEd: GUIDENAME Keyword | 5 |
| 1.13 | ScreenMenuEd: LANGUAGE Keyword | 5 |
| 1.14 | ScreenMenuEd: PROJICON Keyword | 6 |
| 1.15 | ScreenMenuEd: NOWARNCHANGE Keyword | 6 |
| 1.16 | ScreenMenuEd: WINDOWFONT Keyword | 6 |
| 1.17 | ScreenMenuEd: SAVEWINSIZE Keyword | 6 |

Chapter 1

ScreenMenuEd

1.1 ScreenMenuEd.guide

ScreenMenuEd V1.0 By Nick Christie 10th December 1995

In this ScreenMenuEd Guide:

Introduction. What it's all about. **Requirements.** What you need. **Using the Program.** Finding your way around. **ScreenMenuEd's Window.** A tour of the window. **ScreenMenuEd's Menu.** A tour of the menus. **Edit Menu Definition Window.** The defining moment. **Menu Definition Rules.** Do's and don'ts. **Tooltypes & Arguments.** Doing it your way.

These are in ScreenMaster's Guide:

Distribution. Some rules. Copyrights. It's mine, all mine! Disclaimer. But I'm not responsible. Package Contents. What's in the box. Installation. How to get started. Bugs and Improvements. What could be better. Version History. In the beginning... Credits. I'd like to thank... Author. The person responsible.

1.2 ScreenMenuEd: Introduction

Introduction =====

ScreenMenuEd lets you create, edit and test custom menu definitions that you can later attach to a public screen using ScreenMaster, or directly via ARexx and ScreenBacker.

You can define a set of menus with items that launch your favourite commands, in a manner similar to the Workbench Tools menu. If you wish you can tailor the environment that these commands are launched in to suit your requirements, including current directory, standard output, stack and priority.

Don't forget, however, that ScreenBacker itself automatically provides one menu with a suite of useful public screen management functions - see the Standard Menu section of ScreenBacker's guide for details. You will also find the technical specification of menu definition files in that guide.

1.3 ScreenMenuEd: Requirements

Requirements =====

ScreenMenuEd requires:

- AmigaOS 2.04 / Kickstart V37 or higher.

The BGUI version requires:

- bgui.library V38 or higher. (windows, gadgets, etc)

The MUI version requires:

- MUI V2.3 or higher. (windows, gadgets, etc)

The following disk-based Commodore library is required, but is part of AmigaOS 2.04 (and later releases) anyway:

- asl.library V37 or higher. (requesters)

The following disk-based libraries are optional, but their absence will limit the functionality of ScreenMenuEd:

- locale.library V38 or higher. (localization)
- diskfont.library V36 or higher. (custom font support)
- amigaguide.library V33 or higher. (on-line help)

1.4 ScreenMenuEd: Using the Program

Using the Program =====

To start ScreenMenuEd, simply double click on its icon from Workbench, or type its name into a Shell (assuming it's in your Shell's search path). You can also specify an initial menu definition file to load on the command line, or by standard shift-selection of a project icon from the Workbench.

For further information, consult one of the sections below:

- [ScreenMenuEd's Window](#) · [ScreenMenuEd's Menu](#) · [Edit Menu Definition Window](#) · [Menu Definition Rules](#)

1.5 ScreenMenuEd: ScreenMenuEd's Window

ScreenMenuEd's Window =====

Most of ScreenMenuEd's window is occupied by a listview in which the currently defined menus and items are shown. The labels of the menus and items run down the left of the list, with menus in a highlight colour. If an item has a shortcut key defined, this is displayed on the right hand side.

A special kind of item, a separator bar, is shown as a horizontal line. You can insert these separator bars into your menu definitions in order to separate logically distinct sets of items.

At the bottom of the window lie nine button gadgets:

Insert... ----- Inserts a new menu or item into the list, before the current one. The [Edit Menu Definition](#) window opens, to let you set the attributes of the new menu or item.

Edit... ----- Edits the current menu or item in the list. The [Edit Menu Definition](#) window opens, to let you edit the attributes of the menu or item.

Remove ----- Removes the current menu or item completely.

Up -- Moves the current menu or item up one place.

Down ---- Moves the current menu or item down one place.

Test ---- Press this button to test your menu definition. ScreenMenuEd will check it over (see [Menu Definition Rules](#)) and if it's okay, open a new window with your custom menu attached. If you activate this window, you can view and test all your menu items, making sure they perform as desired. When you're done, close the window, or select "End Test" from the Project menu.

Open... ----- Opens a file requester, for you to choose a menu definition file to load. If the current menu definition has been edited but not saved, you will be asked for confirmation first.

Save ---- Saves the current menu definition with the current filename. If the definition has not yet been named, a file requester will open to let you choose one.

Quit ---- Quits ScreenMenuEd. If the current menu definition has been edited but not saved, you will be asked for confirmation first.

1.6 ScreenMenuEd: ScreenMenuEd's Menu

ScreenMenuEd's Menu =====

ScreenMenuEd has just one menu, titled "Project", which contains the following items:

New --- Clears the current list of menus and items completely, to let you start a new menu definition. If the current menu definition has been edited but not saved, you will be asked for confirmation first.

Open... ----- Opens a file requester, for you to choose a menu definition file to load. If the current menu definition has been edited but not saved, you will be asked for confirmation first.

Edit... ----- Edits the current menu or item in the list. The **Edit Menu Definition** window opens, to let you edit the attributes of the menu or item.

Test... ----- Select this item to test your menu definition. ScreenMenuEd will check it over (see **Menu Definition Rules**) and if it's okay, open a new window with your custom menu attached. If you activate this window, you can view and test all your menu items, making sure they perform as desired. When you're done, close the window, or select "End Test" from the Project menu.

Save ---- Saves the current menu definition with the current filename. If the definition has not yet been named, a file requester will open to let you choose one.

Save As... ----- Lets you save the current menu definition under a new filename. A file requester will open to let you choose the new name.

Help... ----- Opens ScreenMenuEd's AmigaGuide documentation. If the guide is not in AmigaGuide's search path, tell ScreenMenuEd where to find it using the **GUIDENAME** tooltype.

About... ----- Opens a requester showing information about ScreenMenuEd.

Quit ---- Quits ScreenMenuEd. If the current menu definition has been edited but not saved, you will be asked for confirmation first.

1.7 ScreenMenuEd: Edit Menu Definition Window

Edit Menu Definition Window =====

When you select to edit or insert a menu or item highlighted in ScreenMenuEd's main window, the Edit Menu Definition window opens. In this window you will find gadgets that let you set and test the attributes of that menu or item.

Type ---- This cycle gadget selects the type for the entry, from Menu, Item and Bar (ie. separator bar). The other gadgets in the window will be appropriately enabled or disabled according to the type selected.

Label ----- Menus and Items. The Label string gadget lets you define what text should appear in the menu for this entry. A label is required for menus and items but is not applicable to separator bars.

Key --- Items only. If you wish to give an item a shortcut key, enter the desired character in the Key gadget. You will then be able to select this item in the menu by pressing the right Amiga key and this character. Remember that the AmigaOS does not distinguish between upper and lower case shortcut keys; shortcuts are generally displayed in upper case by convention. A shortcut key is optional.

Command ----- Items only. A command is required for each item; this command will be executed when you select that item in your custom menu. You can either type the command into the string gadget, or pick it using a file requester by pressing the button to the right.

You can embed up to one instance of the formatting specifier "%s" in the command string. This will, before the command is executed, be replaced by the name of the public screen that the menu is attached to. This makes it possible to get programs to open on that public screen, for example, "MultiView PUBSCREEN %s".

Cur.Dir. ----- Items only. If you wish, you can use this string gadget to specify the desired current directory for the command. You can either type the directory name in, or select one from a requester by pressing the button to the right.

Output ----- Items only. this string attribute is optional; it allows you to specify a standard output channel for the command. Typically this would be a console window, but you could specify a file. As with the command, you can embed up to one instance

of the formatting specifier "%s" in this string, which will be replaced by the name of the public screen that the menu is attached to.

If you leave the Output string blank, you will get the default, which is: "CON:20/20/600/100/ScreenBacker Output/AUTO/-CLOSE/WAIT/SCREEN %s". Notice how "%s" is used to get this console window to open on the public screen.

Stack ----- Items only. Set the stack size in bytes for the command with this integer gadget. The default is 8000 bytes, which should be fine for most programs. Be careful when setting this lower, as many programs crash if they run out of stack space.

Priority ----- Items only. Set the task priority for the command with this integer gadget. This provides a basic method of determining (relatively) how much CPU attention the program will get when it is run. The default is 0, which is a standard value. Setting a priority higher than 3 is strongly discouraged and may upset your Amiga.

OK -- When you are done setting the attributes for this menu or item, select the OK button. ScreenMenuEd will check the values for validity and close the Edit Menu Definition window if all is well. Otherwise you will receive an error message and be returned to the window to correct the problem.

Test ---- Items only. You can test the correct operation of an item by selecting the Test button. This launches the command you have specified with the current directory, output, stack and priority attributes currently set in the window. The Edit Menu Definition window will stay open.

Cancel ----- Select this button to discard any changes you may have made to the current menu or item and close the Edit Menu Definition window.

1.8 ScreenMenuEd: Menu Definition Rules

Menu Definition Rules =====

1. The first entry in a menu definition must be a menu.
2. The last entry in a menu definition must be an item.
3. A separator bar must be preceeded and followed by an item. This means you cannot have a separator bar as the first, or last, entry in a menu. There would be no point in this anyway.
4. Every menu and item must have a label.
5. Every item must have a command.

The first three rules are checked when you try to save or test a menu definition. The last two are checked when you try to load a menu definition, or select OK or Test in the **Edit Menu Definition** window.

1.9 ScreenMenuEd: Tooltypes and Shell Arguments

Tooltypes and Shell Arguments =====

The tooltypes ScreenMenuEd looks for in its program icon are described below. The arguments used from a Shell are identical, except where stated otherwise. The full Shell template (with abbreviations) is:

BGUI: MF=MENUEFILE, PS=PUBSCREEN/K, GN=GUIDENAME/K, LA=LANGUAGE/K, PI=PROJICON/K, NW=NOWARNCHANGE, WF=WINDOWFONT/K, SW=SAVEWINSIZE/S

MUI: MF=MENUEFILE, PS=PUBSCREEN/K, GN=GUIDENAME/K, LA=LANGUAGE/K, PI=PROJICON/K, NW=NOWARNCHANGE

ScreenMenuEd's tooltypes consist of two kinds:

String - You must supply a string of some kind; a filename, public screen name, etc. For example, "LANGUAGE=deutsch".

Boolean - In tooltypes you must supply one of the following: YES, NO, TRUE or FALSE. For example, "NOWARNCHANGE=YES". You can also omit the value entirely, which indicates the YES or TRUE state. On a Shell command line, just state the keyword without a value to get the TRUE state, omit the keyword for FALSE.

BGUI and MUI:

Tooltype Type Short Description ----- **MENUEFILE** STR Menu file to load (Shell only).
PUBSCREEN STR Name of public screen to open on. **GUIDENAME** STR Where to find ScreenMenuEd.guide. **LANGUAGE**
 STR Specify language catalog. **PROJICON** STR Project icon for menu definition files. **NOWARNCHANGE** BOOL Suppress
 change warning on exit.

BGUI only:

Tooltype Type Short Description ----- **WINDOWFONT** STR Font to use in window.
SAVEWINSIZE BOOL Save window size/position on exit.

In the MUI version, an option equivalent to WINDOWFONT can be set using the MUI Preferences program, and SAVEWINSIZE isn't required as MUI remembers window sizes automatically.

1.10 ScreenMenuEd: MENUFILE Keyword

MENUEFILE =====

Versions: BGUI and MUI Type: String Default: <none> Shell: MF=MENUEFILE Tooltype: <not applicable>

This keyword, which is only usable from a Shell, lets you specify a menu definition file that ScreenMenuEd should load initially. If you don't specify a file, the program simply starts out with an empty, unnamed menu definition.

From Workbench, you can specify an initial menu definition file to load by shift-selecting a project icon in the standard manner, or by setting a project icon's default tool to ScreenMenuEd and double clicking on that.

1.11 ScreenMenuEd: PUBSCREEN Keyword

PUBSCREEN =====

Versions: BGUI and MUI Type: String Default: <system default> Shell: PS=PUBSCREEN/K Tooltype: PUBSCREEN

The name of the public screen that ScreenMenuEd should initially open its window on. If omitted, ScreenMenuEd opens on the default public screen, which is normally Workbench, or in the case of the MUI version, on the screen specified using the MUI Preferences program. Be aware that public screen names are case-sensitive, ie. "MyScreen" is not the same as "MYSCREEN".

1.12 ScreenMenuEd: GUIDENAME Keyword

GUIDENAME =====

Versions: BGUI and MUI Type: String Default: ScreenMenuEd.guide Shell: GN=GUIDENAME/K Tooltype: GUIDENAME

If ScreenMenuEd's guide is not on AmigaGuide's search path, set this tooltype to the pathname of this guide so that on-line help is available.

1.13 ScreenMenuEd: LANGUAGE Keyword

LANGUAGE =====

Versions: BGUI and MUI Type: String Default: <system default> Shell: LA=LANGUAGE/K Tooltype: LANGUAGE

ScreenMenuEd supports the locale library of AmigaOS 2.1 and above, allowing it to display text and messages in your preferred language. You would normally specify the language with the Locale Preferences program; this tooltype allows you to override that setting if necessary.

1.14 ScreenMenuEd: PROJICON Keyword

PROJICON =====

Versions: BGUI and MUI Type: String Default: def_ScreenMenuEd Shell: PI=PROJICON/K Tooltype: PROJICON

Use this tooltype to tell ScreenMenuEd what project icon to use when saving menu definition files. If the default tool of this icon is blank, ScreenMenuEd will set it to "ScreenMenuEd" for all icons it saves.

If you set PROJICON to the special value "NONE", no project icon will be saved with menu definition files.

1.15 ScreenMenuEd: NOWARNCHANGE Keyword

NOWARNCHANGE =====

Versions: BGUI and MUI Type: Boolean Default: FALSE Shell: NW=NOWARNCHANGE/S Tooltype: NOWARNCHANGE

Normally when you try to exit ScreenMenuEd having edited the current menu definition, it asks for confirmation first, as you will lose the changes that have been made but not saved. By setting NOWARNCHANGE to TRUE, you can suppress this warning, if you wish.

1.16 ScreenMenuEd: WINDOWFONT Keyword

WINDOWFONT =====

Versions: BGUI only Type: String Default: <screen default> Shell: WF=WINDOWFONT/K Tooltype: WINDOWFONT

ScreenMenuEd normally uses the default font of the screen it opens on. If you wish to override this setting, specify the font with this tooltype, using the following format: "example.font/12", where the 12 indicates the desired point size.

MUI: The "Fonts" page in the MUI Preferences program provides a facility similar to this option.

1.17 ScreenMenuEd: SAVEWINSIZE Keyword

SAVEWINSIZE =====

Versions: BGUI only Type: Boolean Default: FALSE Shell: SW=SAVEWINSIZE/S Tooltype: SAVEWINSIZE

ScreenMenuEd normally opens its window centred under the mouse pointer. If, however, you would like the window to open with a particular size and position each time, set the SAVEWINSIZE tooltype to TRUE.

This will cause ScreenMenuEd to save the window co-ordinates to a small file called "ScreenMenuEd.win" in the ENV: directory each time the program quits. Every time you start ScreenMenuEd, it will read and re-use this window definition.

Once you have positioned the window as you like it, quit ScreenMenuEd, unset the SAVEWINSIZE tooltype, and copy the "ScreenMenuEd.win" file from ENV: to ENVARC:. It will then be copied back to ENV: by the standard Startup-Sequence each time you boot your Amiga.

MUI: The MUI system automatically remembers the window's size and position.
