

GUICreatorWB

Markus Hillenbrand

COLLABORATORS

	TITLE : GUICreatorWB		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Markus Hillenbrand	December 7, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GUICreatorWB	1
1.1	Contents	1
1.2	Preface	1
1.3	1.2 Why should I use this program?	2
1.4	1.3 System requirements	2
1.5	1.4 Installation	3
1.6	1.5 Author	3
1.7	2.3 Registering	4
1.8	Concept	5
1.9	2.2 Legal issues	6
1.10	2.4 Preferences Program	6
1.11	2.5 Main program	7
1.12	2.5 Main program	8
1.13	2.6 Programming hints	10
1.14	2.7 Wanted...	11
1.15	2.8 Future Developments	11
1.16	EBNF of the meta language	12

Chapter 1

GUICreatorWB

1.1 Contents

Contents

Version 1.2 (Jul 04, 95)

I. Introduction

- 1.1 Preface
- 1.2 Why should I use this program?
- 1.3 System requirements
- 1.4 Installation
- 1.5 Author

II. Program description"

- 2.1 The concept
- 2.2 Legal issues
- 2.3 Registering
- 2.4 Preferences program
- 2.5 Main program
- 2.6 Programming hints
- 2.7 Wanted...
- 2.8 Future developments

1.2 Preface

I. Preliminaries

1.1 Preface

What? Another GUI creator?

One should think there are enough programs around for designing GUIs. Yet none of the existing programs could satisfy me completely. Some were too playful (↔ which

impairs performance), others aren't supported any more and cannot use the system
 calls new with OS 3.0.
 But the main reason for developing this program was that almost every one of these
 GUI creators requires an external library for which you may even have to register.

1.3 1.2 Why should I use this program?

1.2 Why should I use this program?

So what does this program have to offer in comparison to other GUI creators?

For one, GUI-Creator uses the functions in GadTools.library v39 exclusively. Thus
 there is no external library that has to provide any functions.

Furthermore, GUI-Creator generates easily readable and modifiable code without any
 GLOBAL variables.

1.4 1.3 System requirements

1.3 System requirements

The program requires at least OS 3.0. (Further versions may support OS 2.04 as well.)

If you don't use sizing gadgets and short cuts for windows or gadgets and rewrite
 the SleepWindow() and WakeWindow() functions, the generated code (not necessarily
 the whole program) should work with OS 2.04 already.)

A fast CPU is recommended, 2 MB of free RAM should be available, otherwise you might
 run out on larger projects.

A hard disk is not required, but recommended. (Anyone who develops programs should
 have one anyway.)

Finally I'd like to mention that a screen size of at least 640x400 is required. For
 productive work 800x600 (-> graphics card) is recommended (there are a lot of
 windows).

I'll also mention that you need a compiler to translate the generated source code.

The following compilers are supported directly at the moment:

- Maxon C/C++

- SAS/C
- GNU C
- DICE

Generators for the following languages are in preparation:

- Oberon
- Modula-2

1.5 1.4 Installation

1.4 Installation

The installation is done by the Commodore Installer program.

When the installation process is finished please start the Preferences Program to configure GUI-Creator for your system.

1.6 1.5 Author

1.5 Author

Address:

Markus Hillenbrand
Hauptstraße 6
36124 Eichenzell-Löschenrod
Germany

Phone: -49/6659/3176 (Sa-Su) and -49/631/99658 (Mo-Fr)
(Germany)

E-Mail: m_hillen@informatik.uni-kl.de
(not during holidays)

Banking:
Raiffeisenbank Eichenzell (BLZ 53060180)
Account # 2103850

This program was developed on an A4000/040.

I'd be grateful for:

- presents
- SCSI hard drives
- SCSI tape drives
- lottery winnings

and of course

- bug reports
 - suggestions for improvement
 - comments
-

and most of all

- registrations

1.7 2.3 Registering

2.3 Registering

The fee for registering is DM 30 or \$30 US.

----> print order form ---> (German) (English)

If you're interested in receiving new versions by mail, each update costs DM 7.50. ↩

(This fee should be transferred to my account upon receiving the update, see ↩

Author.) If you're interested in receiving new versions by e-mail, each ↩
update

costs DM 4. (This fee should be transferred to my account upon receiving the ↩
update,
see Author.)

Anyone who uses GUI-Creator for longer than about 2 weeks (for testing) ↩
without

registering violates the conditions for using this program and infringes on ↩
my

copyright. The unregistered version is not limited in functionality, but from ↩
time

to time requesters will pop up reminding you of registering. Testing GUI-Creator ↩
one

can find out whether the program runs satisfactorily on one's system. If this isn't ↩

the case, simply delete GUI-Creator and forget about it (of course ↩
including

registration), else write to the author of the program (only by e-mail: ↩

m_hillen@informatik.uni-kl.de) and ask for help.

Please keep in mind that many months of labor have gone into GUI-Creator, for ↩
which

the registration fee is not too much in my opinion, considering p&p are included ↩
in

the price. In addition, feedback from the users in the form of registrations ↩
will

decide over possible further development and bug fixes for GUI-Creator.

On registering you will receive a personal "keyfile" that licenses you for the ↩
use

of GUI-Creator and two possible major updates. Of course I cannot guarantee if ↩
there

will be further updates, or how many of them. After installing the keyfile ↩
the

"about" and other requesters will of course disappear. Further major updates (←
after
the second one) will be offered to registered users for a fair price.

There are two ways to register:

a) E-Mail:

You transfer the registration fee to my account (see Author) and send me ←
a
message telling me that you want to register GUI-Creator and when you sent ←
the
money. After I have received the money I will send the keyfile to you by e- ←
mail.
Since it is known that messages can get lost, don't hesitate to contact me ←
again
if my reply takes longer than three weeks. I will then send it iff again ←
of
course. If this won't work at all send me a regular letter and I will send you ←
a
disk containing the keyfile. To prevent problems it is important that you ←
give
your e-mail address and possibly also your normal address on the money ←
transfer
slip - this should rule out any mixups.

b) Mail

You transfer the registration fee to my account (see Author) and send me ←
a
letter telling me that you want to register GUI-Creator and when you sent ←
the
money. Instead of the transfer you can also simply include the money with ←
the
letter, either cash or as euro cheque. No matter how - as soon as I receive ←
the
money I will mail the disk containing the keyfile to you.

If you want to register now, fill in the included order form (German or English ←
)
where the important points are spelled out again...

Many thanks for supporting the shareware concept!

1.8 Concept

II. Program description

2.1 The concept of the program

... is fairly new. The program implements a raster, ie. gadgets and windows in ←
the
project are not calculated by pixels, but according to the font size of the ←
system
default font. This means a window can have eg. a raster like this.

Thus all settings you can adjust in the program are not absolute values, but multiples of the system default font.
 Example: You specify a gadget width of 20 units, and the system default font is 8 pixels wide, then the gadget will be 160 pixels wide.

1.9 2.2 Legal issues

2.2 Legal issues

GUICreator © 1995 Markus Hillenbrand.

GUICreator is shareware. The author retains the copyright for the program. Who wants to use GUI-Creator "seriously" resp. for a longer period of time has to register. Distributing the unregistered version is still allowed as long as there is no commercial gain. A disk containing GUI-Creator must not cost more than DM 5. Including GUI-Creator in a PD series is explicitly allowed under these restrictions, as is the distribution via networks. Regardless of the mode of distribution all data contained in the original GUI-Creator archive have to be transferred together and unchanged. The author offers no guarantee for the correctness or fitness for use of GUI-Creator and takes no liability for any negative consequences caused by the use of GUI-Creator. Updates and bug fixes are not guaranteed. GUI-Creator was created using Maxon C/C++ V3.0 by Maxon Computer, Eschborn. The GUI was created with GUI-Creator itself, so you can get a first impression of the generated code.

MagicWB and thus some of the icons delivered with GUI-Creator are copyright by Martin Huttenloher.

1.10 2.4 Preferences Program

2.4 Preferences Program

To adjust parameters there is an external preferences program. It offers the following options:

- Screen mode preferences

Here you select the screen mode that you like best working with. To get an ASL screen mode requester simply click on the "<<" gadget. Then you can select mode and size of the screen. Keep in mind that a minimum size of 640 x 400 pixels is required. Working is most pleasurable with 800 x 600 and above (with a corresponding system default font).

- Code generator options

Here you set some default values for the program:

- a) Code generator: The shell command used for generating the code. These commands are located in the "C" directory of the main program.
- b) Create main part: Whether a main program is to be generated in the code, that opens all required libraries and displays the first window.
- c) Create all function templates: Here you select whether so-called templates are created, ie function stubs you only have to fill in later.
- d) Create a Makefile: Here you select whether a Makefile (not a real one) is to be created. This Makefile is a shell script that you have to execute to generate the code.
- e) Create Foldmarks: Here you select whether fold marks should be generated for the functions. At this time only '///' is supported (-> GoldED).

- Save and Cancel

cause the selected parameters to be saved or leave the preferences program. If you leave the program, no modifications will be saved. Also, new values don't change the behavior of the running program; to activate them you have to restart the program.

1.11 2.5 Main program

Window Prefs:

- Width : The width of the window as a multiple of the width of the default font.
- Height: The height of the window as a multiple of the height of the default font .

- Name : The name of the window, that prefixes the functions in the source code ←
(so you can differentiate between the windows)
- Title : The window title
- Close : If you enter CLOSEWINDOW here, the window is simply closed, if you enter ←
anything else, a BOOL function is called that you have to supply yourself. ←
If this function returns TRUE, the window is closed, if it returns FALSE, ←
it remains open.
- SizeXX: Should the window have a size gadget (bottom or right hand side or both) ←
?
- Setup : This function is called after the window has been opened and all gadgets ←
have been drawn.
- Refr. : A function that is called when the window's size gadget has been activated ←
(and you may have to redraw something).

1.12 2.5 Main program

2.5 Main program

After starting the program you see two gadgets, that let you either load an existing file or create a new one. Since loading should be self-explanatory, we will now discuss creating a new GUI:

- Name of GUI: The Name of the GUI, that is prefixed when creating source code, so ←
that different projects do not overwrite each other.
- Path of GUI: The path where the source code is to be generated later. You can ←
either enter a path by hand here or click on the adjacent symbol to ←
get an ASL requester.
- Picture : The name of the picture that is displayed when the user of your later ←
program selects "about" from the menu. This path must either be ←
absolute or relative to the path of your later program. (Best use ←
PROGDIR>About.IFF or PROGDIR:pictures/about.IFF or something similar). ←
- About Text : If you click on "Edit" here, you get a number of string gadgets that ←
accept an "about" text, ie a text to accompany the above picture. ←
This can for example be a reference about the program, the author and the copyright.

If you click on "Continue" now, or if you've loaded a GUI, the layout window appears

where you can start the following actions:

- Application:

A now familiar window appears.

- Window :

Click here to add, select, delete windows or to change their order (since the first window is opened first when creating a Makefile). ↩

- Button :

Enter here the parameters that control the look of the gadget.

- String :

Enter here the parameters that control the look of the gadget.

- Integer :

Enter here the parameters that control the look of the gadget.

- Text :

Enter here the parameters that control the look of the gadget.

- Number :

Enter here the parameters that control the look of the gadget.

- Listview :

Enter here the parameters that control the look of the gadget.

- Checkbox :

Enter here the parameters that control the look of the gadget.

- Cycle :

Enter here the parameters that control the look of the gadget.

ATTENTION: The form of the label is: "<Text_1>|<Text_2>|...|Text_n|".

- Radio :

Enter here the parameters that control the look of the gadget.

ATTENTION: The form of the label is: "<Text_1>|<Text_2>|...|Text_n|".

- Scroller :

Enter here the parameters that control the look of the gadget.

- Slider :

Enter here the parameters that control the look of the gadget.

- Palette :

Enter here the parameters that control the look of the gadget.

- Label :

Enter here the parameters that control the look of the gadget.

- Frame :

Enter here the parameters that control the look of the gadget.

- Line :

Enter here the parameters that control the look of the gadget.

- GadgetList :

This is for changing gadget data.

- Save :

A requester for saving the GUI appears.

- Source :

Here you can change the parameters (see preferences program) and have the
 source
 code generated. The application sleeps until code generation has finished.
 There
 is no check (yet) if code generation was successful.

1.13 2.6 Programming hints

2.6 Programming hints

- The "about" picture must be 200 x 100 pixels in size and should not use more
 than
 8 colors (it is probably best to use the 4 workbench colors). If you open
 your
 own screen for the windows, the picture should use that screen's palette.
- The parameters "APTR userdata" are intended for your use. They are simply
 passed
 on.
- The parameters of the function Handle<WindowName>() are a screen on which
 the
 windows are to be opened, coordinates for the upper left corner of the windows
 (-1
 for arrange) and userdata (see above).

- If a window should be able to be closed you can also do it like this:

```
void CancelClicked( ... )
{
    Signal(FindTask(NULL),SIGBREAKF_CTRL_C);
}
```
- The generated Makefile of GUI-Creator_C has a "cc" in the beginning. By making this an alias you can easily compile programs. For the Maxon C/C++ compiler this could look like this:

```
Alias cc MCPP:Bin/MCPPC3 -w1500 -pc -i MCPP:Include|T: []
```
- The stack of the program should be set to 50 000 bytes to avoid stack overflow.

1.14 2.7 Wanted...

2.7 Wanted...

I'm looking for:

- suggestions for improving code generation and the handling of the main program.
- bug reports for the program or the documentation.
- programmers who are interested in writing code generators for other languages (Oberon, E, Modula, Lisp :-). If you are interested and have an e-mail address, you can get a keyfile for GUI-Creator free. Write to me and I'll send you the source for my C generator as a guideline. And for starters here is the EBNF of the meta language for GUI projects: ->
- writing addicts who want to translate the documentation into English, French, Italian, Chinese :-). Send me your translation and you receive a free keyfile. (But first ask if the translation is already being worked on!)

1.15 2.8 Future Developments

2.8 Future Developments

The following is planned for the future:

- menu support
 - new gadget types
 - screen support
 - notifications
-

- online help using the help key and AmigaGuide or MultiView
- generating OS 2.04 code
- integrated locale support
- free choice of fonts
- built in support of exec lists for listviews and similar

All these things should be realized some time. The order in which they are listed doesn't reflect priority. When registering, state which is most important. Based on this information I will (if there are sufficient registrations) implement these things.

1.16 EBNF of the meta language

```

boolean    = "TRUE" | "FALSE".
digit      = "1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"|"0".
letter     = "a"|"b" | ... | "z"|"A"|"B" | ... | "Z".
allletters = letter|"ä"|"ü" | ... | "ß".
dosstring  = letter* [ ":" ] (letter* | "/" )*.

meta       = "GUICreator" version NEWLINE comment definition.
version    = "V" digit+.digit+.
comment    = (letter|digit)*.

NEXT       = NEWLINE TAB|SPACE.
NEXT2      = NEWLINE TAB|SPACE TAB|SPACE.

definition = "DEFINITION " letter+ NEXT path NEXT aboutpic NEXT abouttext NEXT
             "BEGIN" windowdef* NEWLINE "END" letter+.
path        = "PATH=" dosstring.
aboutpic    = "ABOUTPIC=" dosstring.
abouttext   = allletters "|" allletters "|" allletters "|" allletters "|"
             allletters "|" allletters "|" allletters "|" allletters "|"
             allletters "|" allletters "|" allletters "|" allletters "|"
             allletters "|" allletters "|".

windowdef   = "WINDOW " letter+ NEXT2 name NEXT2 title NEXT2 height NEXT2 width
             NEXT2 close NEXT2 size NEXT2 setup NEXT2 refresh NEXT2 gadgetdef*
             NEXT letter+.
name        = "NAME=" letter+.
title       = "TITLE=" allletters*.
height      = "HEIGHT=" digit+.
width       = "WIDTH=" digit+.
close       = "CLOSEWINDOW" | letter+.
size        = "SIZEGADGET=" boolean.
setup       = "SETUPFUNC=" boolean.
refresh     = "REFRESHFUNC=" boolean.

gadgetdef   = (bt | lv | st | cb | cy | in | mx | nm | sc | sl | tx | pa | la | li
             | fr)*.

position    = left "," top "," width "," height.
```

```
rest      = name "," gadgetid "," disabled "," shortcut.

left      = digit+.
top       = digit+.
width     = digit+.
height    = digit+.

name      = letter+.
gadgetid  = letter+.
disabled  = boolean.
shortcut  = letter | "".

text      = allletters*.

bt        = "BUTTON "    position "," rest "," text "," function.
readonly  = boolean.
spacing   = digit+.
string    = allletters*.
justify   = "GACT_STRINGLEFT" | "GACT_STRINGRIGHT" | "GACT_STRINGCENTER".
nmjustify = "GTJ_LEFT" | "GTJ_RIGHT" | "GTJ_CENTER".
replace   = boolean.
maxchars  = digit+.
checked   = boolean.
labels    = allletters "|" (allletters "|")*.
active    = digit+.
number    = digit*.
border    = boolean.
arrows    = digit+.
top       = digit+.
total     = digit+.
visible   = digit+.
freedom   = "LORIENT_HORIZ" | "LORIENT_VERT".
min        = digit+.
max        = digit+.
level     = digit+.
firstcolor = digit+.
depth     = digit+.
color     = digit+.

lv        = "LISTVIEW "  position "," rest "," readonly "," spacing "," function.

st        = "STRING "    position "," rest "," maxchars "," string "," justify ↵
            "," replace "," function.

cb        = "CHECKBOX "  position "," rest "," checked "," function.

cy        = "CYCLE "      position "," rest "," labels "," active "," function.

in        = "INTEGER "   position "," rest "," maxchars "," number "," justify ↵
            "," replace "," function.

mx        = "RADIO "      position "," rest "," labels "," active "," function.

nm        = "NUMBER "     position "," rest "," number "," border "," nmjustify ↵
            "," function.
```

```
sc      = "SCROLLBAR " position "," rest "," arrows "," top "," total "," ↵
        visible "," freedom "," function.

sl      = "SLIDER "    position "," rest "," level "," min "," max "," freedom ↵
        "," function.

tx      = "TEXT "      position "," rest "," text "," border "," function.

pa      = "PALETTE "   position "," rest "," active "," firstcolor "," depth ↵
        "," function.

la      = "LABEL "     position "," gadgetid "," text "," border "," function.

li      = "LINE "      position "," color "," function.

fr      = "FRAME "     position "," text "," color "," function.
```
