

guienvironment

COLLABORATORS

	<i>TITLE :</i> guienvironment		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	guienvironment	1
1.1	GUIEnvironment.guide	1
1.2	What is GUIEnvironment ?	2
1.3	Introduction	2
1.4	A brief overview	4
1.5	The sourcecode	5
1.6	History / Changes	5
1.7	Copyright	6
1.8	Disclaimer	6
1.9	Installation	7
1.10	GUIEnvironment with M2Amiga	7
1.11	GUIEnvironment with OberonA - Oberon in general	7
1.12	GUIEnvironment with SAS/C - C in general	8
1.13	Other compilers / programming languages	8
1.14	Plans / Ideas / ToDo	8
1.15	What you should know...	9
1.16	The Author	9
1.17	rcs	10

Chapter 1

guienvironment

1.1 GUIEnvironment.guide

GUIEnviroment

Version 37.4

General Guide

=====

Copyright

© 1994 Carsten Ziegeler
Augustin-Wibbelt-Str.7
D-33106 Paderborn
Germany

All rights reserved

FREWARE

=====

What is GUIEnviroment ?

Introduction

A brief overview

The sourcecode

History

Copyright / Disclaimer

Copyright

Disclaimer
Contents of this distribution

Installation

The Future / ToDo

What you should know

The Author

1.2 What is GUIEnvironment ?

This chapter gives a short description of what GUIEnvironment can do and why YOU should use it in your own programs. ↔

Introduction

A brief overview

The sourcecode

History

If you are familiar with older versions of GUIEnvironment and you only want to have a look what's new in version 37.4, see chapter History ↔

.

1.3 Introduction

GUIEnvironment is a package which simplifies the creating and interacting with GUIs (Graphical User Interfaces). It consists of two parts:

- 1) The code generator which creates ready to run application modules out of GadToolsBox (© Jaba Development) files. So simply design your GUI with this great "interface builder" and then run GUIEnvironment on this and you don't have to worry anymore about the GUI !
- 2) The guienv.library. This library helps creating and handling GUIs.

Using GUIEnvironment will save you a lot of time when writing applications which take use of the Amiga GUI, although the guienv.library is very small. Your ready to run application will of course later only need the library !

Here are some examples of what the guienv.library does for you:

The developer must implement only the important parts of an application. He must not care about the GUI and its handling, as the `guienv.library` does a lot of work by itself. The interacting with the input events is nearly completely managed by `GUIEnvironment`.

The `guienv.library` combines the `gadtools.library` features, the BOOPSI gadgets and some own GUI stuff.

The library interfaces are very simple but powerful. It was an important aim when designing the `guienv.library` to have short and simple to use interfaces to the GUI combined with the possibility of interaction whenever desired or needed.

This was achieved by using the OS2.04+ facilities, like Amiga callback hooks and tag lists.

So there's no difference in handling `gadtools` gadgets and BOOPSI gadgets anymore.

But the most important fact of `GUIEnv` is - in my opinion - that you don't have to learn "a new programming method". There exist a lot of GUI libraries, but all of them offer a new method of describing the GUI with childrens and parents and so on.

If you only want some support for your GUI and you don't need resizing, it is very frustrating to use those libraries.

`GUIEnvironment` offers the "old fashioned method" of defining GUIs together with the new one in a compatible way !

You only have to replace the names of the functions and structures from intuition and `gadtools` to those of `GUIEnvironment`. (OK, you will have to do just a little bit more, but not very much ...)

`GUIEnvironment` includes very powerful functions for fonts, screens, windows, gadgets, menus and GUI message handling.

All gadget kinds of the `gadtools.library` (OS2.04) are supported along with all BOOPSI gadgets and some own `GUIEnvironment` gadgets.

Due to this, it is now possible to resize and reposition all gadgets, even `gadtools` gadgets !

But this resizing is of course only possible if you tell `GUIEnvironment` how to resize ! So you have to describe the GUI - just for this purpose - in an "object orientated way": You say, e.g. this gadget is so and so many points away from that gadget and the distance between the border and the gadget is so and so.

The GUI can also be adapted to every font automatically. So it is easy to design GUIs for the workbench which fit to the user's preferences without any extra work !

All the GUI's message handling is done by `GUIEnvironment`, this means: E.g. gadgets like cycle kind, mx kind or string kind are handled automatically by `GUIEnv`. The application only has to care for "action" gadgets like button gadgets or menus.

This was achieved by "binding" a variable to a gadget. Each time the gadget status changes, the variable is updated by `GUIEnvironment`. So if you need the current state of the gadget, you simply check the variable ! There is no boring message handling any more for such gadgets !

For example, you are writing a configuration editor with a lot of cycle, mx, string and checkbox gadgets. There will also be two button

gadgets for saving or cancelling !

If you don't use GUIEnvironment, you have to check every message and then remember the status of the gadget. Even worse, if you need string gadgets, then you have to update all this string gadgets before saving, because the user could have left the string gadget without pressing return. (In this case you don't get any message !)

But WITH GUIEnvironment, you tell, when creating the gadgets, which variable belongs to which gadget. (For example a string gadget gets an array of chars, a cycle gadget an integer and so on.)

And now GUIEnv updates this variables for you each time a message for such a gadget arrives. Your application only needs to check the two button gadgets !

Key equivalents for gadgets are detected and handled automatically. The application sees no difference if the gadget was activated by a key or by the mouse pointer.

Callback hooks can be set for each gadget and menu item, so these functions are then called if the gadget resp menu item was activated. This method could be called "object orientated programming", because the whole application is controlled by sending messages to different objects, there is no (visible) main message loop.

GUIEnvironment offers an easy to use help function for gadgets and menus using the AmigaGuide without bothering the application !

All gadgets and menu items can be automatically localized without any further functions !

Also included are very easy to use requester functions which are even able to use ReqTools (© Nico François).

Because GUIEnv was designed as a real shared amiga library you can use it within (nearly) every programming language.

1.4 A brief overview

Here a some "headlines" which illustrate the possibilities and features of GUIEnv. This list is NOT complete, but it gives a brief overview !

Features of GUIEnvironment

- GadToolsBox (© Jaba Development) Code Generator (M2Amiga and OberonA)
- Supports all gadgets kinds of gadtools.library
- Access of BOOSI gadgets
- Simplified handling of fonts, screens, windows, gadgets and menus
- Independent handling of IDCMP messages, e.g.
 - do refreshing automatically
 - notify changes of gadgets
- Supports automatically key equivalents
- Font adaptable GUIs
- Access to all important structures dealing with GUIs

- Own gadget kinds like a progress indicator or bevel boxes
- Resizable gadgets
- Object orientated description of the GUI together with the usual way of describing GUIs
- Request and AmigaGuide support

- ReqTools (© Nico François) requester support

Sorry, but as I was not able to reach Nico François (it seems that his EMail address is not valid anymore...) I didn't want to include the reqtools.library without asking him. If you haven't got it yet, you can get it also in the AMINET !

1.5 The sourcecode

GUIEnv was developed using the great commercial M2Amiga-Modula2-Compiler Version 4.3 by the A+L AG.

GUIEnv actually was a Modula2 library module which was converted into a amiga shared library using m2lmc (© C. Ziegeler) !

1.6 History / Changes

This is a short extract of the developing history of GUIEnv. Every version explains the main differences to the preciding version.

Version 37.2

GUIEnv now checks for the locale.library (Thanks to F.J. Copeland and all the others from the Oberon-A mailing list !)
Removed the old hooks
Changed gadget handling
Font adaptive GUIs
ATTENTION: Because of the huge changes in this release, the guienv.library is not upward compatible ! But don't worry, as GUIEnv is public domain this will be no problem. And I swear from now on the library will be upward compatible !
First working version

Version 37.3

Font adaptive code now works correctly

Version 37.4

New font adaptive code for better results with proportional fonts
ReqTools (© Nico François) support
GadToolsBox (© Jaba Development) support for M2Amiga and OberonA

Take a look at the Changes guide to see a detailed description of all changes.

1.7 Copyright

GUIEnvironment software and documentation are ©1994 by Carsten ←
Ziegeler.

All rights reserved.

This package is freeware, this means that you can copy it freely as long as you don't ask any more money for it than a nominal fee for copying and as long as the complete GUIEnvironment distribution is included. (See chapter Contents).

This package cannot be used for commercial purposes without the written permission from the author.

Also commercial distribution of the GUIEnvironment package is not allowed without written permission from the
author
.

None of the files of the GUIEnvironment package may be modified. Crunching or achiving is allowed only if none of the GUIEnvironment files get modified by it.

If you use GUIEnv you HAVE TO include a remark into your program, that shows that the program uses GUIEnv which is © by C.Ziegeler. Special permission is hereby given to include GUIEnvironment into every real public domain software. But only if this software is fully available without any restrictions for everybody ! You can then include the guienv.library into your package, but you HAVE TO include the above mentioned remark in your documentation as well as in your running program !

It is allowed and highly welcomed to include interface-modules for other languages/compilers. This files may be added to the package. Also it is allowed to add documentation for other languages. But it is NOT allowed to charge extra for this "new" documentation or interface-modules !

It would be very nice, if you could send me such work to include it in the next release of GUIEnv. (see chapter
The Future
)

1.8 Disclaimer

The
author

cannot be held liable for the suitability or accuracy of this manual and/or the program(s) it describes. Any damage directly or indirectly caused by the use or misuse of this manual and/or the program(s) it describes is the sole responsibility of the user himself.

The author is not responsible for any loss of data, damages to software or hardware that may result directly or indirectly from the use of this package. The author reserves the right to make changes to the software and/or documentation without notice.

1.9 Installation

GUIEnv requires at least OS2.04. It was designed as a real amiga ↔
shared
library, so you can use it with nearly every compiler / language which
is able to use the standard libraries.

The first step of installation is to copy the guienv.library to your
libs: directory. (Use the GUIEnv-Install script)

Copy the code generator GEGen to a drawer of your choice. If you want
the online documentation for GEGen available, copy the GEGen.guide (found
in the docs drawer) to the same drawer or to the HELP: directory (if you
have assigned it)!

This package includes the interface modules (header files) for the
M2Amiga-Modula2-Compiler (version 4.3 or above), SASC-Compiler
(version 6.51 or higher) and the OberonA-Compiler (release 1.4 Update 2).

Choose the chapter for your compiler / language :

M2Amiga

OberonA - Oberon in general

SAS/C - C in general

Other compilers / programming languages

1.10 GUIEnvironment with M2Amiga

Using GUIEnvironment with M2Amiga is very simple because GUIEnv was
developed using this compiler version 4.3 !
Please read the M2Amiga document for more information !

1.11 GUIEnvironment with OberonA - Oberon in general

Using GUIEnvironment with OberonA is very simple because GUIEnv was
tested under OberonA release 1.4 (Update 2) very carefully !
For different Oberon compilers there should be no real problems to
convert the interface modules.

Please read the OberonA document for more information !

1.12 GUIEnvironment with SAS/C - C in general

Using GUIEnvironment with the SAS/C is very simple, because the header files are written using SAS/C version 6.51 !

Usually these header files should work with other C compilers as well, perhaps apart from the function declarations.

Please read the SAS/C document for more information !

1.13 Other compilers / programming languages

For different programming languages or compilers you need the library file `guienv.library` in the `libs: drawer` and appropriate interface modules , which are unfortunately not (yet) part of GUIEnv.

You can find in the `libs drawer` of this package a FD file for GUIEnv which could be important for some compilers to create interface modules. (Or perhaps you could use GUIEnv with BASIC ! - No, just a joke !)

Please read one of the documentation for the supported compilers for more information !

1.14 Plans / Ideas / ToDo

Well, at this time I am really not sure if it is worse to continue ←
this
project, because - as you all know - the future of Commodore is very doubtful !
I don't believe that there will be any new Amigas out until the end of next year (1995) ! In all probability that's the end of the Amiga as a different operating system....(Sorry, but someday you have to face the truth !)

So, actually, I am looking for a new operating system to do my work on, so I will not spend much time in ANY amiga projects and as GUIEnvironment is now - in my opinion - complete, there is no reason for new versions of the `guienv.library`, unless some of YOU have some ideas/suggestions about what is missing in the library or unless there are still some more bugs or problems...

Still missing are, of course, the code generator for SAS/C and the `GetFileImage boopsi class`, but currently I have no time to work on this ! Perhaps I can do the `boopsi class` until end of this year, but this is no promise !

I have many new ideas to implement in the GUIEnvironment, but at this time I see really no sense in it. But if the Amiga rises again to the top of the computer world, I will be there !

If you have any interface modules for different languages as currently supported or if you have written some nice examples, please mail them to me and I will include them in the next release of GUIEnvironment.

or german stamps, but not e.g. english stamps or french food).
All posting will be at your own risk and of your own free will.

So please, if possible use electronic mail to avoid any problems !

Enjoy it !

"...It's the same old ending - time to go..." - GENESIS/The Lamb Lies
Down On Broadway

1.17 rcs

\$RCSfile: GUIEnvironment.guide \$

\$Revision: 1.5 \$

\$Date: 1994/11/03 15:49:35 \$

GUIEnvironment Project Guide

Copyright © 1994, Carsten Ziegeler

Augustin-Wibbelt-Str.7, 33106 Paderborn, Germany
