guienvironment

COLLABORATORS			
	TITLE : guienvironment		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 22, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 guienvironment **1**

1.1 GUIEnvironment Project 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 Past - Present - Future . . . 8
1.15 What you should know... . . . 9
1.16 The Author . . . 9
1.17 rcs . . . 10

Chapter 1

guienvironment

1.1 GUIEnvironment Project Guide

```
                                GUIEnviroment

                                Version 37.6

                                General Guide

 ========================================================================

                                   Copyright

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

                              All rights reserved

                                    FREEWARE

 ========================================================================


 What is GUIEnviroment ?

       Introduction
       A brief overview
       The sourcecode
       History

  Copyright / Disclaimer

       Copyright
       Disclaimer
       Contents of this distribution


 Installation
```

Past - Present - Future

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 familar with older versions of GUIEnvironment and you only want to have a look what's new in version 37.6, 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 and the geclass.library. The first library helps creating and handling GUIs and the second one is for sharing private BOOPSI classes.

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 importan parts of an applications. 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 complete 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 powerfull. 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 childs 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 libaries.
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 descripe 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 do 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 writting 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 these 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 these 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 BOOPSI gadgets
- Simplified handling of fonts, screens, windows, gadgets and menus
- Indipendent 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
- Requester 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 from the AMINET !

1.5 The sourcecode

GUIEnvironment was developed using the 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) !

As I will stopp developing on the Amiga, I offer the possibility to buy
all the source of the whole GUIEnvironment for someone who is interested
in expanding GUIEnvironment. If you are interested, please contact me !

1.6 History / Changes

This is a short extract of the developing history of GUIEnvironment.
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

Version 37.5
------------
Get File Image BOOPSI Class
GEClass Libray for making private BOOPSI classes "public"
Many bugs in GEGen removed and improved GEGen
Totally localized requesters (guienv.catalog)
68020 version of the libraries
```

```
Version 37.6
------------
Minor bug fixes in the guienv.library
Several bug fixes in GEGen
Notify Option for GEGen
Textfield gadget (© Mark Thomas) support via GEClass
Fixed problems under OS3.0 with Multiview/Amigaguide and guienv.library

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

1.7 Copyright

1.8 Disclaimer

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 libaries.

The first step of installation is to copy the libraries 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)!

If you have a localized workbench you have to copy the appropriate
catalog files by hand ! (Sorry, but I didn't have the time to write a
good Installer script...) The catalogs for the guienv.library and GEGen
are both in the libs/catalogs drawer !

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 it 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 it 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 were tested and created 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 progamming languages or compilers you need the library files
guienv.library and geclass.library in the libs: drawer and appropriate
interface modules , which are unfortunately not (yet) part of this package.

You can find in the libs drawer of this package FD files for the libraries,
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 Past - Present - Future

Well, at this time I am really not sure if it is worth 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....and I don't believe any of the big
rumours about new Amigas. (Sorry, but someday you have to face the truth !)

As GUIEnvironment is now - in my opinion - complete, there is no reason
for new versions, 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 IS, of course, the code generator for SAS/C but I have
absolutely NO time to do this, so if someone of YOU is interested in
doing such a great work, please let me know..

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 (and a new and better GUIEnvironment, of course, too !)

But beginning with 1995 I will move my developing to NeXtStep, so I will freeze all amiga projects. But there will be still bug fixes for all reported and known errors/problems ! (And there will also be an update for the OberonA 1.5 interface modules !)
So, if someone of YOU is interested in buying all the source of the whole GUIEnvironment project for M2Amiga to keep on my work, please contact me !

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. (But please mail only examples which are documented !)

(See Author for the address.)

1.15 What you should know...

Before you are delivered from this boring documentation, I have to do some apologies:
The whole documentation is not entitled to be complete, without any mistakes or at least understandable, although I worked really very hard on it.
If you don't know how to program an Amiga, you can throw away the documentation (but PLEASE NOT GUIEnvironment !). In many cases I left some explanations you can find in the RKRMs. If you are familar with them, you will probably have no problems...

To reach a "wider audience" I tried to translate the definitions and the documentation into the english language. I hope you can understand it and that there will be noone who died by laughing at these texts !

I am NOT a C programmer. In fact: I don't like this very confusing and often very strange language. I even HATE it, because if you are familiar with C you write programs noone else can understand because of the "encryption" ! So my advice: If you have any problems with C, take a look at Modula2 or Oberon2. Actually, there is a great Oberon-2 compiler in the public domain. Look for the OberonA compiler © by Frank J. Copeland !
Sorry, but I don't do advertising for commercial products. (But there exists a very good Modula2 developement software...)

But the thing I wanted to say was: As I'm not a C programmer at all, the C programs and also the documentation are not the best way of using C. I translated it only, because a lot of people think that C is the only way of programming the Amiga - BUT THEY ARE WRONG !

1.16 The Author

Suggestions, bug reports, improvements, extensions and of course
appreciation, too, can be send to the following address:

    Carsten Ziegeler   EMail: tarot@uni-paderborn.de
    Augustin-Wibbelt-Str. 7
    D-33106 Paderborn
    Germany

Even if you don't like GUIEnvironment, please tell me why...

ATTENTION: There will be no warranty or responsibility for posted goods.
           If you want to have your work back, you must include enough
           return postage I can use here in germany (e.g. german money
           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

   "It's OK !" - Roy Cakes better known as Ray Cokes

   Thanks to Steve Mason for his fantastic Experience Radio Show every
   week ! It's the best time to write programs ... and to fall in TRANCE.

1.17 rcs

$RCSfile: GUIEnvironment.guide $

$Revision: 1.7 $
    $Date: 1994/12/17 15:42:29 $

  GUIEnvironment Project Guide