gegen

COLLABORATORS

	TITLE : gegen	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

gegen

1.1 GUIEnvironment Code Generator Guide

GUIEnvironment

Code Generator

==========

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

==========

1.2 GEGen Introduction

GEGen is a code generator for M2Amiga and OberonA which reads the .gui files created by the GadToolsBox (© Jaba Development) and writes a new module.
Using both, the GadToolsBox and GEGen, it is very easy to implement GUI's into your application.
First draw you GUI with the GadToolsBox and then run GEGen. Finally, implement all the actions and functions for your application.

1.3 Getting started

Copy GEGen to a place you want. It is the best way to put in a drawer
which is in your path, e.g. C: or sys:utilitities.
If you want to have the online documentation available, you have to install
also the GEGen.guide. You can copy this guide either in your current work-
ing directory or into the HELP: drawer ! (Using the HELP: drawer is of
course much easier, because GEGen always looks there first.)
If you have a localized workbench copy the catalog file from the libs
drawer to your LOCALE: drawer.
And that's all.

GEGen can be started from the Workbench and the CLI, but it ignores all
parameters given neither by the CLI or the Workbench.

1.4 The Main GUI

GUIFile : The file name for the GUI. This file has first to be created
with the GadToolsBox.

Module : The name and path of the output module.

Language: You can choose between M2Amiga and OberonA. For M2Amiga two
files will be created (.def and .mod), and of course for
OberonA only one.
If you want to create the GUI for both, M2Amiga and OberonA,
you will have to do two creation runs, one for each language.

Gadget IDs: The code for the gadget IDs:
- export : Create the identifiers and export them
- write : Create the identifiers for internal use
- ignore : Don't create the constants

The identifier is a constant for each gadget which contains
the appropriate ID.

Resizable: If this flag is turned on, the GUI will be resizable. Make
sure that you have set on the right IDCMP flags in the
GadToolsBox.

Notify: If this flag is turned on, a notify variable for each gadget
will be created. This variable has the name <gadgetlabel>Value
and is linked to the gadget using the GEG_VarAddress tag.
All message for such a gadget are handled internal and will
not appear in the main input event loop ! (Unless you use
the GEG_HandleInternal tag.)
See also the GUIEnvironment Gadget Guide for notification.

Begin : Start the code generator
Quit : End GEGen

The status field displays a possible error or the succesfull creation.

You can also reach some information about GEGen using the menu strip:

About : Some information about GEGen

```
Help  : Displays this guide (if available)
Quit  : End GEGen
```

1.5 GEGen Output

```
As I totally wrote a new code generator for the GadToolsBox there will
be some differences between the existing code generators !
First of all here are some VERY BIG differences:
- It is currently not possible to reach a public screen with a definite
  name. Always the current public screen will be used.
- Intuitexts are not supported
- The window size is calculated for inner width and inner height. This
  code is not always correct. To avoid any problems, it is the easiest
  way to use the inner width and inner height flags of the GadToolsBox.
- The module created open's always a font ! This is either the font
  used to display the GUI or if the GUI is font adaptiv, the font the
  GUI was created with ! (Font styles and flags are not supported !)
- For each GetFileImage used within a GUI an object is created, this means
  if you have 5 gadgets displaying the GetFileImage, you will have 5
  objects of this class ! This was done for easier resizing of GUIs.
- The GUI can be self-resizable

For each project the module exports the following "things" :

  <projectname>GUI : GUIInfoPtr;

  PROCEDURE Open<projectname>GUI() : BOOLEAN;

  PROCEDURE Close<projectname>GUI;

Call Open<projectname>GUI to open your GUI, this includes opening the
screen, font and window ! Now you can handle your GUI using the GUIInfoPtr.
And, of course, Close<projectname>GUI will close your GUI (and window,
font and screen).
If Open<>GUI returns FALSE, the GUI (or the window, screen...) couldn't
be created ! So you can't use the GUI by now !
If you have more than one project, the screen and font will be opened
with the first GUI opened and closed with the last one to be closed !

GEGen is a very quick (and dirty) implementation, so there probably will
be a lot of errors and bugs, because I couldn't test everything !
Sorry, but it did work with all of the 8 test GUIs I created with the
GadToolsBox. So, please, if you find any problems/bugs/errors or if
you have any suggestions, please contact me !
```

1.6 rcs

```
$RCSfile: GEGen.guide $

$Revision: 1.3 $
    $Date: 1994/12/16 21:22:28 $
```

GUIEnvironment Code Generator Guide

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