

**0121a938-0**

Rodrigo Reyes

**COLLABORATORS**

	<i>TITLE :</i> 0121a938-0		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Rodrigo Reyes	February 12, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>0121a938-0</b>	<b>1</b>
1.1	GccOpts, main page. . . . .	1
1.2	The false commercials . . . . .	2
1.3	I. What Is It ? . . . . .	2
1.4	II. How Does it work ? . . . . .	3
1.5	III. The PROJECt group. . . . .	3
1.6	IV. The LIBRARIES group . . . . .	4
1.7	V. The OPTIONS group . . . . .	4
1.8	VI. The Makefile . . . . .	4
1.9	VII. Example . . . . .	5
1.10	VIII. Misc . . . . .	6
1.11	History . . . . .	6
1.12	What is a makefile ? . . . . .	7

---

# Chapter 1

## 0121a938-0

### 1.1 GccOpts, main page.

GCCOPTS v1.3

Summary of this  
tremendous  
file:

Introduction  
... DON'T read it...

I.

What~is~it  
... A quick overview.

II.

How~does~it~work~?  
... Another overview...

III.

The~Project~group  
... Guess what it is...

IV.

The~Libraries~group  
... About... well, you know

V.

The~Options~group  
... Another chapter...

VI.

The~Makefile  
... Still the same boring things

VII.

Example  
... Forget it.

VIII.

Misc  
... Yeepee ! The last chapter !

[click here ->](#)

~~~~~

---

if you really don't know what a makefile is.

## 1.2 The false commercials

Introduction : The FALSE commercials.

FANTASTIC !!! TOTALLY FREE !!! Try it now !

GccOpts is the last product of our secret lab,~product~of~several~years~of~intensive research, by the greatest scientists~in~the~world.

This program will make you the life EASIER~and~INCREDIBLY~FUN,~women~will go into coma, when crossing you in the~street,~your friends~will~invit~you in LUXURIOUS restaurant EVERY DAY, your~boss~will DOUBLE~your~pay, and~on your way, the blind will recover the~vision,~the leaf~will~recover the~hearing...

Try it now, and get SUPERNATURAL POWERS, that~will~make~your~ennemies~envious, and turn the girls "enamoradas"...

Try it now, and your programs will NEVER be~bugged~AGAIN,~GccOpts~garantees ALL your PROGRAMS to be TOTALLY~BUG~FREE~and~completly~OPTIMIZED...

Only GccOpts can make it so EASY...

## 1.3 I. What Is It ?

I. What is it ?

-----

GccOpts~is~a~program~that~allows~you~to~automatically~and~quickly~write~a

Makefile

~for~the~Amiga~GCC.~It's~simple~enough,~I~hope,~not~to~need~any~documentation,~so~don't~be~too~much~disturbed~if~I~don't~offer~you~megabytes~of~precise~informations~about~this~program~(anyway~I~am~reachable thru~many~net~addresses).

This~program~provides you~a~useful~and~graphical~interface~to~easily~make~your~

Makefiles

~for~Amiga~GCC.~It~needs~AmigaOS 2.0~or +, and of course the ixemul.library. An example of a project with a graphical interface is provided in the package (see

Example

), have a look on it if you really don't know how works a makefile.

Just install it in your path, and enjoy (yes,~this~is~real plug-and-play~:))

## 1.4 II. How Does it work ?

II. How the does it work ?

-----

Firstly you should install GccOpts by copying it to the gnu:bin/ directory.

Launch GccOpts from the CLI or the Workbench, and have a look on the panel that just opened on the screen. On the right, a group named "Project" contains a listview and some other gadgets. There are also two other groups, "Libraries" and "Options". You'll have to refer to the concerned section of the manual for further information about each of this groups.

After having filled some gadget with some random value of your own, press the gadget "Save Makefile", and the GccOpts will save a Makefile which purpose is to make you compile your project with the gnu Make command.

If you don't want to save the Makefile you have done or modified, just click on the "Cancel" gadget or on the upper-left CloseWindow gadget.

## 1.5 III. The PROJECT group.

III. The "Project" group.

-----

In this group, the programmer (you!) will need to give some useful informations about the project he (you, ever) want the Makefile to be done for.

First of all, the "Project" listview must contain all the files that are part of the projet. The given files can be any \*.c or \*.o file (GccOpts will process them differently in the makefile, however). Include files are accepted, and gcc will be told to use them, with the directive under the form : "-i your\_include.h -i another\_include.h"... If you add any \*.o or \*.a file in the project listview, it will be added to the Makefile as an archive that should not to be deleted by the "clean" rule.

To add a name into the listview, just type it inside the string gadget below, and press enter. To delete one entry, select it with the mouse, and click on the DEL gadget below the listview. The gadget "?" will pop up a filerequester, and if you select a file with it, this one will be added to the listview. You can also let the program automatically choose the project files, by clicking the "SCAN" gadget. GccOpts will then scan all the files in the current directory, and insert in the listview every .c file.

Three cycle gadgets below the listview can be used to select some options about the code generation. This gadgets are :

"Code" : which refers to the processor your project will be generated for : 68000 - 68020 - 68030 - 68040/020-881 - 68040.

"Fpu" : only allows two value (could have been a boolean thing), fpu or not : 68881 or NONE, default is NONE.

---

"Optimize" will permit to select the level of optimization for your program: NONE, LEVEL 1, LEVEL 2, or LEVEL 3. Choose it in peace with your conscience and your memory (and you'll need some for the third level).

Another, last but not least, gadget allows you to select the name of the output file. So the name of the gadget, you'd have guessed it, you clever programmer, is "Output". The name you give it will be used as the first rule of the Makefile, so that "make NAME\_OF\_YOUR\_PROGRAM\_IN\_OUTPUT" will just compile all the newer part of your project. This is the same than "Make" alone, as this is the first rule of the Makefile.

That's all for the "Projet" group.

## 1.6 IV. The LIBRARIES group

IV. The "Libraries" group.

-----

This group contains only one gadget, a listview, into which the clever programmer will put the names of all the libraries that its project will need to be linked with. To know how to add an entry to the listview, please refer to the previous "Project group" section.

The name of the library can be either the normal file name, i.e. "libamiga.a" (without quotes), or either "amiga". "libamiga" will also be accepted. This names will be inserted into the project as libraries, and will be referred in the Makefile as "-lamiga".

## 1.7 V. The OPTIONS group

V. The "Options" group.

-----

This group contain a listview that will contain some of the options you can need for your project. To flag or un-flag an option, just go onto the listview gadget, and click on the option you want to select/deselect. An option is currently selected if a "\*" is added at its beginning.

This options are divided into 6 parts: Code generation, Linking, Stack, Optimizing, Debugging, and Warning. For more information about them, please read the gcc documentation (click here --> READ GCC.GUIDE to do so). However, Names are quite clear, you shouldn't have problems with them, and if you have, you can contact me for more informations.

## 1.8 VI. The Makefile

-----

VI. The Makefile.

-----

---

The generated~

Makefile

~should~not~be~modified~if~you~want~it~to~be~

correctly re-processed~by~GccOpts.~The~makefile~should~be~run~by~the Make command.

This

Makefile

contains three useful entries for the clever programmer :

- all : This rules will call the two others, clean and compile, and your project will be fully recompiled.

- clean : this will clean-up the project, ie. just remove every \*.o file leading around.

~YOUR\_PROJECT\_NAME~::~~will~recompile~only~the~new~or~modified~files, and~create~an~executable~named~YOUR\_PROJECT\_NAME,~according~to~the~informations~and~options~given. The program will recursively scan the given c-source file for non-system include files (the token " #include" should be at the beginning of the line, with no blank or tabulation). Non-system include files are those bracketed with \" (while standart system includes should be bracketed with '<' and '>', and won't be scanned). All this includes will be linked to the correct c files that call them.

The default rule executed by Make (alone) is YOUR\_PROJECT\_NAME. In order to be able to re-read it correctly, GccOpts puts special informations inside the created Makefile, so don't modify it if you want to re-use later the result of your (not so) hard work.

## 1.9 VII. Example

VII. The example.

A very simple example is provided, and will quickl show you how GccOpts works. The program to compile is simply an "Hello world", and has strictly no importance. If Gccopts is not installed in your system, you should copy it in the gnu:bin/ directory.

In the workbench screen, open the drawer "GccOpts\_Example", and click on the Options icon. This icon will call the GccOpts program (which should already be installed in the gnu:bin/ or C: directory).

There you can modify the compiler options. Once done, just click on the "Save Makefile" gadget. The makefile will be saved, and the program will close its window and shut down. Then, you can double-click on the MakeProject icon. This script will open a shell window and call the make program.

You can also start GccOpts from a shell, and do just the same as above. To compile the project, just type "make" or "make all" in the shell.

Please note that the program will be recompiled only if the c-source file is modified.



## 1.10 VIII. Misc

VIII. Misc.

-----

This program is totally freeware, if you paid for this, be~sure~I~have~not seen a penny of your money. The source of this program~is~also~free,~although not distributed, just ask for it, and I'll~E-Mail~it~uuencoded to~you.

Of course the author gives no warranty at all for this~program,~it~may~not work at all, crash your computer, destroy your~terminal,~invit~your~stepmother at home for a full month, or any other~terrible~disaster,~that's YOUR business and YOUR responsibility in using~this~program.

Thanks to Philippe Brand for its wise advices.

TO DO:

-----

- Flex & bison support
- coffee-machine support
- Whatever you clever programmer will tell me...

You can contact me for comments, requests, anything :

E-Mail : rodrigo.reyes@ramses.fdn.org  
rodrigo.reyes@electron.fr

Fido : 2:320/104.16  
2:320/214.7

AmigaNet : 39:180/1.16

UCNet : 88:320/1.7

or snailmail : Rodrigo Reyes  
ELECTRON BBS  
Boîte postale 101  
91801 Brunoy France  
France

## 1.11 History

History

-----

v1.3 (6 August 1995)

- Automagically scan sources and generate dependencies with non-system include files.

- Some minor bugfixes.
- The makefile is made more readable.

v1.2 (30 July 1995)

- GCC2.7.0 options added : stackcheck and stackextend.
- The useless \*.h files in the Project group are removed.
- Added the version number of GccOpts in the makefile to prevent future option changes.
- Icons providing a graphical interface.

v1.1 (25 May 1995)

- Scan gadget added to the Project Group.
- The generated MakeFile is improved.

v1.0 (22 May 1995)

- First Release.

## 1.12 What is a makefile ?

What is a makefile ?

A "makefile" is a scriptfile that describes your project (the different components of your program: c files, includes, objects files, libraries, etc).

This file (which may be called makefile, or Makefile, or GNUMakefile) will be used by a program, called "make", whose job is to always keep your project correctly updated. So, if the program does not exist yet, it will simply compile it entirely. Or else it will compare the date and time of the source file with the date and time of the compiled file, and will compile it if, and only if, needed. All of this according to the rules described in the makefile.

This provide a very powerly and easy-to-use developping system for the programmer.

---