

**gccprefs**

COLLABORATORS

	TITLE : gccprefs		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		November 18, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>gccprefs</b>	<b>1</b>
1.1	Welcome . . . . .	1
1.2	Willkommen . . . . .	2

# Chapter 1

## gccprefs

### 1.1 Welcome

ENGLISH DEUTSCH

Documentation for cprefs (gccprefs and vbccprefs)

AUTHOR

Dietmar Eilert (mail@dietmar-eilert.de)  
<http://www.dietmar-eilert.de>

Introduction

In a nutshell, cprefs is for the gcc and vbcc compilers what scoptions is for the SAS/C compiler: a visual frontend for setting compiler options. The idea behind this program is to have an options file ("gcc.options" or "vbcc.options") in your source code folders and to use the compiler without any command line options. Instead, the compiler is made to include the options from the options file. The purpose of cprefs is to create and maintain that file. Two flavours of cprefs are provided: "gccprefs" is for gcc users and "vbccprefs" is the same program for vbcc users.

Starting cprefs

CD to the source code folder of your project and run cprefs with the EDIT/S argument, ie: GCCPREFS EDIT. Or, if you are using vbcc: VBCCPREFS EDIT. cprefs will load the local options file, show its user interface and allow you to change compiler options with simple mouse clicks. If no local options file exists, one is created for you.

The options presented by cprefs should be self-explanatory if you are familiar with your compiler, otherwise you will have to have a look at the compiler documentation. To help you get started, the default options are always greyed-out (selecting such options does not produce any entries in the options file). Some options may not have any effect depending on the other compiler options you are using (for example, PPC options have no effect if compiling for Motorola 68000 as target). cprefs does not take these dependencies into account but leaves the choice of on-conflicting options to the user.

Note that compiler options may not be specified twice: if you set an option in

---

cprefs, do not set it again in the command line.

## Usage

### Command Line

To use the options file generated by cprefs with your compiler, it must be included in the command line via shell output redirection. For this purpose, run cprefs without any argument. cprefs then simply echoes the content of the local options file (actually, it echoes the content of that file minus linefeeds). Here are two examples. Note that what looks like the apostroph character in these lines is actually the backtick character:

```
gcc `gccprefs` -o programname main.c
```

```
vbcc `vbccprefs` -o programname main.c
```

How it works: The Amiga Shell understands that the backticks delimit a command and it evaluates that command. The output of that command is included in the command line before this line is finally executed.

### Makefiles

In GNU makefiles, you can use the "shell operator" instead of backticks to include the options in a command line. Here is an example:

```
$(EXE) : main.o
    gcc main.o -o $(EXE) $(shell gccprefs)
```

Note that the indentation of command lines in GNU makefiles (what appears as spaces before "gcc" in the example above) must be made with TABS, not spaces. Make sure that your editor is properly configured. If you are using GoldED, go to the TAB configuration dialog and verify that the TAB key is configured to insert TABS.

### Structure of options files

The configuration file format is fairly simple: one compiler option per line, using the same syntax as if used in a command line.

## 1.2 Willkommen

ENGLISH DEUTSCH

Dokumentation für cprefs (gccprefs und vbccprefs)

AUTOR

Dietmar Eilert (mail@dietmar-eilert.de)  
<http://www.dietmar-eilert.de>

Einführung

In aller Kürze: cprefs ist für die gcc- und vbcc-Compiler, was scoptions für den SAS/C-Compiler ist: ein grafisches Frontend zum Setzen von Compileroptionen.

---

Die Idee hinter diesen Programmen ist es, alle Compileroptionen in eine lokale Datei im Verzeichnis der Sourcecodes ("gcc.options" oder "vbcc.options") zu schreiben und sie nicht mehr umständlich in der Befehlszeile anzugeben. Der Compiler wird anschliessend dazu gebracht, die Optionen aus der Options-Datei zu lesen und ansonsten ohne Compileroptionen in der Kommandozeile benutzt. Der Zweck von cprefs ist es, die Options-Datei zu erstellen und zu warten. Von cprefs gibt es zwei verschiedene Versionen: gccprefs ist für Benutzer des gcc-Compilers und vbccprefs ist entsprechend für Benutzer des vbcc-Compilers.

### cprefs starten

Wechseln sie mit CD in den Sourcecode-Ordner Ihres Projektes und starten Sie cprefs mit der EDIT/S-Option: GCCPREFS EDIT. Oder, für Benutzer von vbcc: VBCCPREFS EDIT. cprefs wird dann die lokale Optionsdatei laden und zum Einstellen der Optionen eine einfach zu bedienende grafische Oberfläche anzeigen. Sollte zu diesem Zeitpunkt noch keine Konfigurationsdatei existieren, so wird automatisch eine erzeugt.

Die von cprefs präsentierten Optionen sollten selbsterklärend sein, andernfalls ist ein Blick in die Compilerdokumentation unumgänglich. Um Ihnen den Start zu vereinfachen werden Standardwerte in grauer Schrift dargestellt. Derartige Einstellungen führen nicht zu Einträgen in der Optionsdatei. Einige Optionen haben abhängig von gewählten anderen Optionen keine Auswirkung. Beispielsweise bleiben die PPC-Optionen ohne Auswirkung, wenn Sie Motorola 68000 als Target ausgewählt haben. cprefs berücksichtigt diese Zusammenhänge nicht. Es bleibt Ihnen überlassen, konfliktfreie Einstellungen zu wählen.

Beachten Sie, daß verschiedene Compiler-Optionen nicht mehrmals angegeben werden dürfen: Wenn sie Optionen mit gccprefs setzen, können Sie die gleichen Optionen nicht in der Kommandozeile verwenden.

### Verwendung

#### Kommandozeile

Um die von cprefs erstellte Optionsdatei mit Ihrem Compiler zu benutzen muss sie mit Hilfe der Shell-Ausgabeumleitung in die Kommandozeile aufgenommen werden. Zu diesem Zweck wird das Programm cprefs ohne Optionen aufgerufen: cprefs gibt dann lediglich den Inhalt der Optionsdatei abzüglich der enthaltenen Zeilenumbrüche aus. Hier zwei Beispiele für die Verwendung (beachten Sie, dass die Apostroph-ähnlichen Zeichen in der Zeile tatsächlich das "Backtick"-Zeichen sind):

```
gcc `gccprefs` -o programname main.c
```

```
vbcc `vbccprefs` -o programname main.c
```

Wie es funktioniert: Die Shell erkennt das Backtick-Zeichen als Begrenzung eines Programmnamens und führt diese Programm aus. Dessen Ausabe wird in die Kommandozeile aufgenommen, bevor diese letztlich in ihrer Gesamtheit ausgeführt wird,

#### Makefiles

In GNU-Makefiles verwenden Sie statt des Backticks den "Shell Operator", um die cprefs-Optionen in eine Zeile einzubinden. Hier ist ein Beispiel:

```
$(EXE) : main.o
        gcc main.o -o $(EXE) $(shell gccprefs)
```

Beachten Sie, dass in GNU-Makefiles Kommandozeilen mit TABS (und nicht mit Leerzeichen) eingerückt werden müssen. Im voranstehenden Beispiel muss demnach die gcc-Zeile mit einem TAB eingerückt werden. Stellen Sie sicher, dass Ihr Editor entsprechend konfiguriert ist. Benutzen Sie GoldED, dann öffnen Sie den TAB-Konfigurationsdialog und überprüfen dort, dass die TAB-Taste als "TABS einfügen" konfiguriert ist.

#### Aufbau der Konfigurationsdatei

Das Format der Konfigurationsdatei "gcc.config" ist sehr einfach: eine Compileroption pro Zeile im gleichen Format, wie Sie es bei der Verwendung in einer Kommandozeile verwendet würden.