

**C++**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i>  C++		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		November 18, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>c++</b>	<b>1</b>
1.1	Welcome . . . . .	1
1.2	SAS/C . . . . .	1
1.3	SAS/C menu . . . . .	2
1.4	SAS/C toolbar . . . . .	2
1.5	SAS/C keyboard . . . . .	3
1.6	Templates . . . . .	3
1.7	vbcc . . . . .	3
1.8	vbcc menu . . . . .	4
1.9	vbcc toolbar . . . . .	4
1.10	vbcc keyboard . . . . .	5
1.11	GeekGadgets (gcc) . . . . .	5
1.12	gcc menu . . . . .	5
1.13	GeekGadgets (gcc) toolbar . . . . .	6
1.14	GeekGadgets (gcc) keyboard . . . . .	6
1.15	StormC . . . . .	7
1.16	StormC menu . . . . .	7
1.17	StormC toolbar . . . . .	8
1.18	StormC keyboard . . . . .	8
1.19	Generic C/C++ . . . . .	9
1.20	C/C++ menu . . . . .	9
1.21	C/C++ toolbar . . . . .	10
1.22	C/C++ keyboard . . . . .	10
1.23	Templates . . . . .	10
1.24	Create new project . . . . .	11
1.25	Generate makefile . . . . .	11
1.26	Run . . . . .	12
1.27	Make . . . . .	12
1.28	Choose makefile . . . . .	13
1.29	Insert empty function . . . . .	14

1.30 Insert empty struct . . . . .	14
1.31 Add fold markers . . . . .	14
1.32 Add comment . . . . .	15
1.33 Insert other element . . . . .	15
1.34 Documentation . . . . .	15
1.35 Help . . . . .	16
1.36 Change compiler . . . . .	16
1.37 Options . . . . .	16
1.38 Generate GST file . . . . .	17
1.39 Project · Debug . . . . .	18
1.40 Errors and warnings . . . . .	18
1.41 Errors and warnings - Show/Hide . . . . .	18
1.42 Errors and warnings - Navigate . . . . .	19
1.43 Errors and warnings - Delete/Clear . . . . .	19
1.44 Open message browser . . . . .	20
1.45 Show next error . . . . .	20
1.46 Documentation . . . . .	20
1.47 Documentation - Lookup GST symbol . . . . .	21
1.48 Documentation - Explain SAS/C function . . . . .	21
1.49 Exit SAS/C . . . . .	22
1.50 Options . . . . .	22
1.51 Explain error . . . . .	22
1.52 Documentation . . . . .	23
1.53 Compiler wechseln . . . . .	23
1.54 Project · Debug . . . . .	23
1.55 Options . . . . .	24
1.56 Documentation . . . . .	24
1.57 Start StormC environment . . . . .	24
1.58 Show window . . . . .	25
1.59 Project management . . . . .	25
1.60 Project · Make . . . . .	26
1.61 Project · Compile . . . . .	26
1.62 Run . . . . .	26
1.63 Project · Debug . . . . .	27
1.64 Navigate with debugger . . . . .	27
1.65 Save breakpoints . . . . .	28
1.66 Search address . . . . .	29
1.67 Compile unchanged files . . . . .	29
1.68 Hierarchical make . . . . .	29

1.69	Errors and warnings . . . . .	30
1.70	Willkommen . . . . .	30
1.71	SAS/C . . . . .	30
1.72	SAS/C Menü . . . . .	31
1.73	SAS/C Symbolleiste . . . . .	31
1.74	SAS/C Tastatur . . . . .	32
1.75	Templates . . . . .	32
1.76	vbcc . . . . .	33
1.77	vbcc-Menü . . . . .	33
1.78	vbcc-Symbolleiste . . . . .	33
1.79	GeekGadgets (gcc) . . . . .	34
1.80	GeekGadget-(gcc)-Menü . . . . .	34
1.81	GeekGadget-(gcc)-Symbolleiste . . . . .	35
1.82	GeekGadget-(gcc)-Tastatur . . . . .	35
1.83	vbcc-Tastatur . . . . .	35
1.84	StormC . . . . .	36
1.85	StormC-Menü . . . . .	36
1.86	StormC-Symbolleiste . . . . .	37
1.87	StormC-Tastatur . . . . .	37
1.88	Generische C/C++-Umgebung . . . . .	38
1.89	C/C++-Menü . . . . .	38
1.90	C/C++-Symbolleiste . . . . .	39
1.91	C/C++-Tastatur . . . . .	39
1.92	Templates . . . . .	40
1.93	Neues Projekt anlegen . . . . .	40
1.94	Makefile erzeugen . . . . .	40
1.95	Starten . . . . .	41
1.96	Make . . . . .	41
1.97	Makefile auswählen . . . . .	43
1.98	Leere Funktion einfügen . . . . .	43
1.99	Leere Struktur einfügen . . . . .	43
1.100	Faltmarkierungen einfügen . . . . .	44
1.101	Kommentieren . . . . .	44
1.102	Anderes Element einfügen . . . . .	44
1.103	Dokumentation . . . . .	45
1.104	Hilfe . . . . .	45
1.105	Compiler wechseln . . . . .	45
1.106	Einstellungen . . . . .	46
1.107	GST-Datei erzeugen . . . . .	46

1.108Projekt · Debuggen . . . . .	47
1.109Fehler und Warnungen . . . . .	48
1.110Fehler und Warnungen - Zeigen/Verbergen . . . . .	48
1.111Fehler und Warnungen - Navigation . . . . .	48
1.112Fehler und Warnungen - Löschen . . . . .	49
1.113Eigenes Fenster öffnen . . . . .	49
1.114Nächsten Fehler anzeigen . . . . .	50
1.115Dokumentation . . . . .	50
1.116Dokumentation - GST-Symbol nachschlagen . . . . .	50
1.117Dokumentation - SAS/C-Funktion erklären . . . . .	51
1.118SAS/C beenden . . . . .	51
1.119Einstellungen . . . . .	52
1.120Fehler erklären . . . . .	52
1.121Dokumentation . . . . .	52
1.122Compiler wechseln . . . . .	53
1.123Projekt · Debuggen . . . . .	53
1.124Einstellungen . . . . .	53
1.125Dokumentation . . . . .	54
1.126StormC-Umgebung starten . . . . .	54
1.127Fenster zeigen . . . . .	54
1.128Projektmanagement . . . . .	55
1.129Projekt · Make . . . . .	55
1.130Projekt · Modul übersetzen . . . . .	56
1.131Starten . . . . .	56
1.132Projekt · Debug . . . . .	56
1.133Navigation im Debugger . . . . .	57
1.134Breakpunkte speichern . . . . .	58
1.135Adressse suchen . . . . .	58
1.136Neu übersetzen . . . . .	59
1.137Hierarchisches Make . . . . .	59
1.138Fehler und Warnungen . . . . .	59

# Chapter 1

## C++

### 1.1 Welcome

ENGLISH DEUTSCH

Documentation for the C/C++ mode of GoldED Studio

AUTHOR

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

Overview

Menus, toolbars and other configurable elements to some degree depend on the variant of the C/C++ mode currently in use. Please chose a C/C++ mode to get an overview of its properties and features:

Supported Compilers

SAS/C  
vbcc  
Geek Gadgets (gcc)  
StormC 3  
Other

### 1.2 SAS/C

ENGLISH DEUTSCH

Options available in the 'SAS/C' variant of the C/C++ mode:

SAS/C:

SAS/C-Menu  
Toolbar  
Keyboard

---

Templates

Notes:

The SAS/C mode is targeted at SAS 6.58 and may or may not work with other versions (untested).

SEE ALSO:

Other Compilers

## 1.3 SAS/C menu

ENGLISH DEUTSCH

SAS/C Menu

- Create new project
- Generate makefile
- Options
- Generate GST file
- Project · Make
- Project · Run
- Project · Debug
- Choose makefile
- Insert empty function
- Insert empty struct
- Add fold markers
- Add comment
- Insert other element
- Errors and warnings
- Open message browser
- Documentation
- Change compiler
- Exit SAS/C

SEE ALSO:

The SAS/C mode    Other Compilers

## 1.4 SAS/C toolbar

ENGLISH DEUTSCH

SAS/C Toolbar

- New document
- Make
- Options
- Next error
- Run
- Debugger



Wizard  
Help

SEE ALSO:

The SAS/C mode    Other Compilers

## 1.5 SAS/C keyboard

ENGLISH DEUTSCH

SAS/C Shortcuts

F2    Set compiler options  
F3    Make  
F4    Debug with shell startup  
S-F4    Debug with workbench startup  
F6    Run executable  
F7    Show SCMSG browser  
S-F7    Hide SCMSG browser  
F8    Help  
F9    Quit SAS/C

SEE ALSO:

The SAS/C mode    Other Compilers

## 1.6 Templates

ENGLISH DEUTSCH

TEMPLATES

The C/C++ mode comes with templates to insert typical C/C++ elements into a source code by just typing the first few letters. For example, type "for(" and a full for-loop will be inserted. The following templates are detected:

```
if(  
switch(  
for(  
else{  
while(  
do{
```

## 1.7 vbcc

ENGLISH DEUTSCH

Options available in the vbcc variant of the C/C++ mode

---

vbcc:

- Menu
- Toolbar
- Keyboard
- Templates

Notes:

The vbcc mode is targeted at vbcc 0.8 and may or may not work with other versions (untested).

SEE ALSO:

- Other Compilers

## 1.8 vbcc menu

ENGLISH DEUTSCH

vbcc Menu

- Create new project
- Generate makefile
- Options
- Project · Make
- Project · Run
- Choose makefile
- Insert empty function
- Insert empty struct
- Add fold markers
- Add comment
- Insert other element
- Explain error
- Documentation
- Change compiler

SEE ALSO:

- The vbcc mode
- Other Compilers

## 1.9 vbcc toolbar

ENGLISH DEUTSCH

vbcc Toolbar

- New document
- Make
- Options
- Run

---

Wizard  
Help

SEE ALSO:

The vbcc mode    Other Compilers

## 1.10 vbcc keyboard

ENGLISH DEUTSCH

vbcc Shortcuts

F2    Set compiler options  
F3    Make  
F6    Run executable  
F8    Help

SEE ALSO:

The vbcc mode    Other Compilers

## 1.11 GeekGadgets (gcc)

ENGLISH DEUTSCH

Options available in the GeekGadgets (gcc) variant of the C/C++ mode

GeekGadgets (gcc):

Menu  
Toolbar  
Keyboard  
Templates

Notes:

The GeekGadget (gcc) mode is targeted at gcc 2.95.3-4 for AmigaOS/m68K and may or may not work with other gcc versions or other CPU families (untested).

SEE ALSO:

Other Compilers

## 1.12 gcc menu

ENGLISH DEUTSCH

GeekGadgets (gcc) Menu

---

- Create new project
- Generate makefile
- Switch compiler
- Options
  - Project · Make
  - Project · Run
  - Project · Debug
- Choose makefile
- Insert empty function
- Insert empty struct
- Add fold markers
- Add comment
- Insert other element
- Documentation
- Change compiler

#### Notes:

The GeekGadgets (gcc) mode is targeted at gcc 2-95-3 for AmigaOS/m68K and may or may not work with other versions or CPU families (untested).

#### SEE ALSO:

[The GeekGadgets \(gcc\) mode](#)   [Other Compilers](#)

## 1.13 GeekGadgets (gcc) toolbar

ENGLISH DEUTSCH

GeekGadgets (gcc) Toolbar

- New document
- Make
- Options
- Run
- Debug
- Wizard
- Help

#### SEE ALSO:

[The GeekGadgets \(gcc\) mode](#)   [Other Compilers](#)

## 1.14 GeekGadgets (gcc) keyboard

ENGLISH DEUTSCH

GeekGadgets (gcc) Shortcuts

F2   Set compiler options

---

F3 Make  
F4 Debug  
F6 Run executable  
F8 Help

SEE ALSO:

The GeekGadgets (gcc) mode    Other Compilers

## 1.15 StormC

ENGLISH DEUTSCH

Options available in the StormC variant of the C/C++ mode

StormC:

Menu  
Toolbar  
Keyboard  
Templates

Notes:

Usually the first things you want to do in the StormC mode are to start StormC and to open a project (or to create a new project). Until you open a project, almost all StormC-related functions in menus and toolbars will not be available (have no effect) because they need a project to work on.

SEE ALSO:

Start StormC Open project    Other Compilers

## 1.16 StormC menu

ENGLISH DEUTSCH

StormC Menu

Start StormC environment  
Show window  
Project management  
Project · Make  
Project · Compile  
Project · Run  
Project · Debug  
Compile unchanged files  
Hierarchical make  
Insert empty function  
Insert empty struct  
Add fold markers

- Add comment
- Insert other element
- Errors and warnings
- Documentation
- Change compiler

SEE ALSO:

[The StormC mode](#)   [Other Compilers](#)

## 1.17 StormC toolbar

ENGLISH DEUTSCH

StormC Toolbar

- New document
- Make
- Next error
- Run
- Debugger
- Show PC
- Go
- Singlestep
- Stepover
- Stepend
- Wizard
- Help

SEE ALSO:

[The StormC mode](#)   [Other Compilers](#)

## 1.18 StormC keyboard

ENGLISH DEUTSCH

StormC Shortcuts

F1	Open new editor window
F2	Open file
F3	Save all open buffers
F4	Create new project
F5	Open project
F6	Save project
F7	Compile
F8	Make
F9	Run
F10	Debug
S-F1	Show current variables
S-F2	Show global variables
S-F3	Show watched variables

S-F4 Show registers  
S-F6 Debugger: Go  
S-F7 Debugger: Single step  
S-F8 Debugger: Step over  
S-F9 Debugger: Go to RTS  
S-F10 Show PC

SEE ALSO:

The StormC mode    Other Compilers

## 1.19 Generic C/C++

ENGLISH DEUTSCH

Options available in the generic variant of the C/C++ mode

Generic C/C++:

Menu  
Toolbar  
Keyboard  
Templates

Notes:

The generic C/C++ mode is reserved for a user-defined compiler and can be used with almost any compiler. You will be prompted to configure this mode for your compiler when you start using functions like Create new project.

StormC users may want to use this variant of the C/C++ mode to control their compiler via a classic makefile (as opposed to the StormC user interface that is addressed by the StormC variant of the C/C++ mode).

SEE ALSO:

Other Compilers

## 1.20 C/C++ menu

ENGLISH DEUTSCH

C/C++ Menu

Create new project  
Generate makefile  
Project · Make  
Project · Run  
Choose makefile  
Insert empty function

---

- Insert empty struct
- Add fold markers
- Add comment
- Insert other element
- Documentation
- Change compiler

#### Notes

The generic C/C++ mode will try to detect the compiler type by examining the files in the local directory. For example, if it finds the file "makefile", it will use GNU make with it (as it will smake from SAS/C with any "smakefile" and dmake from DICE with any "dmakefile").

#### SEE ALSO:

The Generic C/C++ mode    Other Compilers

## 1.21 C/C++ toolbar

ENGLISH DEUTSCH

#### C/C++ Toolbar

- New document
- Make
- Run
- Wizard
- Help

#### SEE ALSO:

The Generic C/C++ mode    Other Compilers

## 1.22 C/C++ keyboard

ENGLISH DEUTSCH

#### C/C++ Shortcuts

- F3    Make
- F6    Run executable
- F8    Help

#### SEE ALSO:

The Generic C/C++ mode    Other Compilers

## 1.23 Templates

---



ENGLISH DEUTSCH

FUNCTION

Empty program (ISO C)  
Empty Amiga program

DESCRIPTION

These functions will create a new empty C source code from templates shipped with the C/C++ mode.

## 1.24 Create new project

ENGLISH DEUTSCH

FUNCTION

Create new project

DESCRIPTION

Use this function to create a new project. A new folder with an empty source code and a suitable makefile will be generated. The generated project can be compiled into an executable via the make functions in toolbar and menu.

Depending on the installed compiler you should continue with setting the compiler options. If your compiler comes with a frontend for that task (SAS/C) or if GoldED's C/C++ mode provides such a program (cprefs for gcc and vbcc), use it. Do not edit the makefile in that case. Otherwise, edit the compilation and link commands in the makefile.

SEE ALSO:

Generate Makefile

## 1.25 Generate makefile

ENGLISH DEUTSCH

FUNCTION

Generate makefile

DESCRIPTION

Adds an automatically generated makefile to an existing folder containing C/C++ and/or assembler source codes. The source codes and the C/C++ header files included by the

source codes are analyzed and a suitable makefile is created.

SEE ALSO:

Create new Project

## 1.26 Run

ENGLISH DEUTSCH

FUNCTION

Run

DESCRIPTION

Runs the compiled program.

Note: This function locates the executable by looking at the options file and/or at the makefile. You should have the assignment "EXE = <executable>" somewhere in the makefile so that the file can be located. This is automatically the case if your makefiles are generated by the C/C++ mode.

SEE ALSO:

Project/Make

## 1.27 Make

ENGLISH DEUTSCH

FUNCTION

Make

DESCRIPTION

Starts the build process for the current project. The files specified in the makefile are compiled and linked according to the rules in the makefile. You may run the program after the build process has completed.

NOTES

You should have the assignment "EXE = <executable>" somewhere in the makefile so that the C/C++ mode can locate the executable. This will automatically be the case if your makefiles are generated by the C/C++ mode.

SAS/C:

---

The 'make' commands in toolbar and menu will either use the SAS/C 'smake' program to build the project (if a "smakfile" is found in the project folder) or GNU make (if a "makefile" is found). Use whatever make tool you prefer. The 'new project' command defaults to use GNU make.

You should have the string "sc:" (the path to the SAS/C compiler) somewhere in the GNU "makefile" so that the C/C++ mode knows that the makefile is for SAS/C and will launch the SAS/C message browser before proceeding with the make process.

See also: Options

vbcc:

You should have the string "vbcc" (the name of the compiler) somewhere in the makefile so that the C/C++ mode knows that the makefile is for vbcc.

See also: Options

GeekGadgets (gcc):

You should have the string "gcc" (the name of the compiler) somewhere in the makefile so that the C/C++ mode knows that the makefile is for vbcc.

See also: Options

Dice:

The 'make' commands in toolbar and menu will either use the DICE 'dmake' program to build the project (if a "dmakfile" is found in the project folder) or GNU make (if a "makefile" is found). Use whatever make tool you prefer. The 'new project' command defaults to use GNU make.

You should have the string "dcc:" (the path to the Dice compiler) somewhere in the GNU "makefile" so that the C/C++ mode knows that the makefile is for Dice.

## 1.28 Choose makefile

ENGLISH DEUTSCH

FUNCTION

Choose makefile

DESCRIPTION@

Enable or disable manual selection of the makefile before

---

eack make process (the makefile is selected automatically if manual selection is disabled).

## 1.29 Insert empty function

ENGLISH DEUTSCH

FUNCTION

Insert empty function

DESCRIPTION

Run this macro to insert an empty C function into the source code. The inserted function is optionally preceded by a comment and/or enclosed by fold markers for easy folding.

SEE ALSO:

Add fold markers  
Add comment

## 1.30 Insert empty struct

ENGLISH DEUTSCH

FUNCTION

Insert empty structure

DESCRIPTION

Run this macro to insert an empty structure into the source code. The structure is optionally preceded by a comment and/or enclosed by fold markers for easy folding.

SEE ALSO:

Add fold markers  
Add comment

## 1.31 Add fold markers

ENGLISH DEUTSCH

FUNCTION

Add fold markers

DESCRIPTION

---

Enable or disable insertion of fold markers when inserting an empty function or structure into the source code.

SEE ALSO:

Insert empty function  
Insert empty struct

## 1.32 Add comment

ENGLISH DEUTSCH

FUNCTION

Add comment

DESCRIPTION

Enable or disable insertion of a comment block when inserting an empty function or structure into the source code.

SEE ALSO:

Insert empty function  
Insert empty struct

## 1.33 Insert other element

ENGLISH DEUTSCH

FUNCTION

Insert switch  
Insert for  
Insert if  
Insert else  
Insert do  
Insert while

DESCRIPTION

These functions will insert an empty switch/for/if/else/do/while block into the source code.

SEE ALSO:

Templates

## 1.34 Documentation

---

ENGLISH DEUTSCH

FUNCTION

GoldED Studio and C/C++  
ISO C Quick Reference  
GNU Make

DESCRIPTION

These functions will provide quick access to documentation shipping with the C/C++ mode.

## 1.35 Help

ENGLISH DEUTSCH

FUNCTION

Help (toolbar button)

DESCRIPTION

Show this document.

## 1.36 Change compiler

ENGLISH DEUTSCH

FUNCTION

Change compiler · SAS/C  
Change compiler · vbcc  
Change compiler · GeekGadgets (gcc)  
Change compiler · StormC  
Change compiler · Other compiler

DESCRIPTION

These functions will switch to another variant of the C/C++ mode with compiler-specific toolbars, menus etc., provided that the selected variant is installed. The "other compiler" variant is reserved for a user-defined compiler and can be used with almost any C/C++ compiler. StormC users may want to use this variant to control the compiler via a classic makefile (as opposed to the StormC user interface).

## 1.37 Options

---

ENGLISH DEUTSCH

FUNCTION

Options

DESCRIPTION

Use this function to set the compiler and linker options. refer to the SAS/C documentation for a detailed description of the available options. A short description for input fields can be obtained by moving the mouse pointer over the input field and pressing the HELP button.

## 1.38 Generate GST file

ENGLISH DEUTSCH

FUNCTION

Generate GST file

DESCRIPTION

Creates a precompiled header file for SAS/C, also known as "GST" file. Precompiled header files are used to reduce the compilation time. More details about GST files can be found in the SAS/C documentation. The precompiled headers will be generated by compiling either "gst.c" or "main.c". In other words, you should have the file "gst.c" in your project folder and it should include all headers required for your project. Otherwise, "gst.c" should be empty and not be compiled by the makefile because its sole purpose is to pull in the headers.

Note: To use precompiled header with a project, the user-selected name of the GST file, for example "all.gst", must be set prior to using this function via the SAS/C options program, on the compiler options page (enter "all.gst" into the "GST" textfield).

Tip: for larger projects, create a master header file (for example "defs.h") that includes all other header files and is itself included by all source code of the project. In that case, there will only be one '#include "defs.h"' line in "gst.c".

SEE ALSO:

SAS/C options

---

## 1.39 Project · Debug

ENGLISH DEUTSCH

FUNCTION

- Debug (toolbar button)
- Debug as CLI program
- Debug as WB program

DESCRIPTION

These functions start the SAS/C debugger and run the compiled executable in the debugger. For this function to be useful, the project must have been compiled with embedded debug information (the corresponding compilation flags are set via the SAS/C options program on the "Compiler" page). The program is started as if run via CLI respectively as if run via the Workbench (in the latter case, an icon for the executable must be present or the debugger will fail to start).

SEE ALSO:

- SAS/C options

## 1.40 Errors and warnings

ENGLISH DEUTSCH

FUNCTION

- Show browser
- Hide browser
- Show message · Top
- Show message · Previous
- Show message · Current
- Show message · Next
- Show message · Bottom
- Delete message
- Clear all messages

## 1.41 Errors and warnings - Show/Hide

ENGLISH DEUTSCH

FUNCTION

- Show browser
- Hide browser

DESCRIPTION

---



These functions will show or hide the SAS/C message browser window. The message browser shows errors and warnings generated during the compilation process. Note that hiding the window differs from closing the window in that the messages from the last make are not discarded. If you want the window to show up automatically during a make, check the "Open message browser" menu item.

SEE ALSO:

Open message browser

## 1.42 Errors and warnings - Navigate

ENGLISH DEUTSCH

FUNCTION

Show message · Top  
Show message · Previous  
Show message · Current  
Show message · Next  
Show message · Bottom

DESCRIPTION

These functions will take you to the source code position of the first/last/next/previous error message. Depending on whether the message browser window is enabled or disabled, details on the error will be displayed in the browser window.

SEE ALSO:

Open message browser  
Show browser  
Hide browser

## 1.43 Errors and warnings - Delete/Clear

ENGLISH DEUTSCH

FUNCTION

Delete message  
Clear all messages

DESCRIPTION

These functions will delete the current or all messages from the SAS/C message browser.

---

SEE ALSO:

Show browser

## 1.44 Open message browser

ENGLISH DEUTSCH

FUNCTION

Open message browser

DESCRIPTION

Enable or disable automatic opening of the SAS/C message browser window. If the browser window is disabled, the message browser will run in the background (without a window) during the make process. You will still be able to navigate the errors by using the functions in the "Errors and warnings" menu or the "Next error" button in the toolbar.

SEE ALSO:

Errors and warnings  
Show next error

## 1.45 Show next error

ENGLISH DEUTSCH

FUNCTION

Show next error (toolbar button)

DESCRIPTION

This function will show the location of the next error in the source code. If the SAS/C message browser window is enabled, details on the error are displayed by the browser.

SEE ALSO:

Enable/Disabled message browser

## 1.46 Documentation

ENGLISH DEUTSCH

FUNCTION

---

GoldED Studio and C/C++  
SAS/C® Contents  
SAS/C® Compiler options  
SAS/C® Browser  
SAS/C® Debugger  
SAS/C® Libraries  
SAS/C® Troubleshooting  
ISO C Quick Reference  
GNU Make  
Lookup GST symbol...  
Explain SAS/C function  
Explain SAS/C function...

#### DESCRIPTION

These functions will provide quick access to documentation shipping with the C/C++ mode and SAS/C.

## 1.47 Documentation - Lookup GST symbol

ENGLISH DEUTSCH

#### FUNCTION

Lookup GST symbol...

#### DESCRIPTION

This function looks up information on symbols in the current GST file. The term "GST file" refers to a SAS/C-style precompiled header file generated with "Generate GST file" menu.

#### SEE ALSO:

Generate GST file

## 1.48 Documentation - Explain SAS/C function

ENGLISH DEUTSCH

#### FUNCTION

Explain SAS/C function  
Explain SAS/C function

#### DESCRIPTION

These functions look up information on standard C library function (like malloc) in the SAS/C manuals.

---

## 1.49 Exit SAS/C

ENGLISH DEUTSCH

FUNCTION

Exit SAS/C

DESCRIPTION

Stops the SAS/C message browser, closes open help windows and removes the compiler from memory.

SEE ALSO:

SAS/C message browser

## 1.50 Options

ENGLISH DEUTSCH

FUNCTION

Options

DESCRIPTION

Use this function to set the compiler and linker options via the cprefs program. The options are saved into the local "vbcc.options" file in the project folder. Refer to the cprefs guide for a description of cprefs and how to use the option files in the command line or in makefiles.

## 1.51 Explain error

ENGLISH DEUTSCH

FUNCTION

Explain error

DESCRIPTION

This function explains numeric vbcc error codes.

SEE ALSO:

Documentation

---

## 1.52 Documentation

ENGLISH DEUTSCH

FUNCTION

GoldED Studio and C/C++  
Errors and Warnings  
vbccprefs  
ISO C Quick Reference  
GNU Make

DESCRIPTION

These functions will provide quick access to documentation shipping with the C/C++ mode and vbcc.

## 1.53 Compiler wechseln

ENGLISH DEUTSCH

FUNCTION

gcc (ANSI C)  
g++ (C++)

DESCRIPTION

Use these menu items to select a gcc frontend. The selected frontend is inserted into the makefile by replacing the pattern "<tab>gcc" with "<tab>g++". Makefiles which were generated outside the C/C++ mode can not be changed with this function (because these patterns may not be present). Please edit such makefiles manually.

## 1.54 Project · Debug

ENGLISH DEUTSCH

FUNCTION

Debug (toolbar button)  
Debug (menu)

DESCRIPTION

These functions start the GCC debugger and run the compiled executable in the debugger. For this function to be useful, the project must have been compiled with embedded debug information (the corresponding compilation flags are set via the gccprefs program on the "Debugger Options" page).

---

SEE ALSO:

GCC options

## 1.55 Options

ENGLISH DEUTSCH

FUNCTION

Options

DESCRIPTION

Use this function to set the compiler and linker options via the cprefs program. The options are saved into the local "gcc.options" file in the project folder. Refer to the cprefs guide for a description of cprefs and how to use the option files in the command line or in makefiles.

## 1.56 Documentation

ENGLISH DEUTSCH

FUNCTION

GoldED Studio and C/C++  
gcc  
gcc and AmigaOS  
Appendix: Libnix  
Appendix: gccprefs  
ISO C Quick Reference  
GNU Make

DESCRIPTION

These functions will provide quick access to documentation shipping with the C/C++ mode and GCC.

## 1.57 Start StormC environment

ENGLISH DEUTSCH

FUNCTION

Start StormC environment

DESCRIPTION

This function will start the StormC development environment.

---

Usually the next thing you want to do is to open a project or to create a new project. Until you open a project, almost all other StormC-related functions will not be available (have no effect) because they need a project to work on.

SEE ALSO:

Other Compilers

## 1.58 Show window

ENGLISH DEUTSCH

FUNCTION

Show window · Modules  
Show window · Errors  
Show window · Variables  
Show window · Variables, page "Actual"  
Show window · Variables, page "Globals"  
Show window · Variables, page "Watched"  
Show window · Variables, page "Registers"  
Show window · History  
Show window · Breakpoints  
Show window · Stack  
Show window · Profiler protocol  
Show window · Hex editor  
Show window · Disassembler

DESCRIPTION

These functions will open various windows of the StormC development environment. Note that some windows are not available until you have opened a project, until you have compiled a project or until you are running or debugging an executable.

## 1.59 Project management

ENGLISH DEUTSCH

FUNCTION

Open project...  
Create new project...  
Add source code to project

DESCRIPTION

Use these functions to open a StormC project, to create a new project or to add the text in the editor window to the current project. Note that almost all functions in StormC are

---

project-centric. For example, you can not compile the source code in the current window unless a project has been opened and the source code actually belongs to that project. For information on StormC projects, please read the compiler manual.

## 1.60 Project · Make

ENGLISH DEUTSCH

FUNCTION

Project · Make

DESCRIPTION

Starts the build process for the current project. The files specified in the project configuration are compiled and linked. You may run the program after the build process has completed.

Note that some details of the make process depend on the state of checked/unchecked menu items in the menu:

Compile unchanged files  
Hierarchical make

## 1.61 Project · Compile

ENGLISH DEUTSCH

FUNCTION

Project · Compile

DESCRIPTION

Compiles the current source code (the file displayed in the editor window). Note that due to the project-centric design of StormC, this function requires that a project has been opened and that the file is part of the project.

SEE ALSO:

Open project

## 1.62 Run

---



ENGLISH DEUTSCH

FUNCTION

Run

DESCRIPTION

Runs the compiled program.

SEE ALSO:

Project · Make

## 1.63 Project · Debug

ENGLISH DEUTSCH

FUNCTION

Start Debugger  
Show PC  
Go  
Single step  
Step over  
Go to RTS  
Save Breakpoints  
Search address

DESCRIPTION

<Start debugger> runs the StormC debugger and executes the compiled executable in its environment (the debug-related options should be set in the StormC compiler options dialog prior to using this function). The navigation functions are used to control execution in the debugger.

## 1.64 Navigate with debugger

ENGLISH DEUTSCH

FUNCTION

Show PC  
Go  
Single step  
Step over  
Go to RTS

DESCRIPTION

---

The debugger navigation functions control execution while a file is execute inside the StormC debugger. They work in conjunction with the breakpoints displayed by the editor (to show or hide the breakpoints, check or uncheck the "Show breakpoints" item in the editor's "view" menu).

Show PC

Show the current point of execution in the source code.

Go

Continue execution until the next activated breakpoint is reached. Note that you can set or unset breakpoints with a mouse click but you can not change their position because the compiler determines if and where it can hold execution.

Single step

Execute the next line (or rather, continue execution to the next breakpoint, set or unset, usually located in the next line). If execution reaches a function, the debugger will step into the function (see also: "step over").

Step over

Execute the next line (or rather, continue execution to the next breakpoint, set or unset, usually located in the next line). If execution reaches a function, the debugger will execute it but will NOT single-step through it (see also: "single step").

Go to RTS

Continue execution until the end of a function is reached.

SEE ALSO:

Start debugger

## 1.65 Save breakpoints

ENGLISH DEUTSCH

FUNCTION

Save breakpoints

DESCRIPTION

This function saves the current state of breakpoints (to continue debugging with the same set of activated breakpoints

---

later).

## 1.66 Search address

ENGLISH DEUTSCH

FUNCTION

Search address

DESCRIPTION

Locate a memory address in the debugged file.

## 1.67 Compile unchanged files

ENGLISH DEUTSCH

FUNCTION

Compile unchanged files

DESCRIPTION

Depending on the state of this menu item (checked or unchecked), the make process will compile modified files only or recompile all files.

SEE ALSO:

Make

## 1.68 Hierarchical make

ENGLISH DEUTSCH

FUNCTION

Hierarchical make

DESCRIPTION

Depending on the state of this menu item (checked or unchecked), the make process will perform a normal or a hierararchical make.

SEE ALSO:

Make

---

## 1.69 Errors and warnings

ENGLISH DEUTSCH

FUNCTION

Show StormC browser  
Next error  
Previous error

DESCRIPTION

The next/previous navigation functions will show the location of the next respectively previous error in the source code. Detailed information about the error is displayed in the StormC error browser (if open).

## 1.70 Willkommen

ENGLISH DEUTSCH

Dokumentation der C/C++-Umgebung von GoldED Studio

AUTOR

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

Überblick

Menüs, Symbolleisten und andere konfigurierbare Elemente erscheinen in Abhängigkeit von der benutzten Variante der C/C++-Umgebung. Wählen Sie eine Variante aus der folgenden Liste aus, um einen Überblick über die vorhandenen Optionen zu erhalten:

Unterstützte Compiler

SAS/C  
vbcc  
GeekGadgets (gcc)  
StormC 3  
Anderer Compiler

## 1.71 SAS/C

ENGLISH DEUTSCH

Beschreibung der SAS/C-Variante der C/C++-Umgebung

SAS/C:

---

Menü  
Symbolleiste  
Tastatur  
Templates

Hinweise:

Die SAS/C-Umgebung ist auf SAS/C 6.58 ausgerichtet; die Kompatibilität zu anderen Versionen wurde nicht getestet.

SIEHE AUCH:

Andere Compiler

## 1.72 SAS/C Menü

ENGLISH DEUTSCH

SAS/C Menü

Neues Projekt anlegen  
Makefile erzeugen  
Einstellungen  
GST-Datei erzeugen  
Projekt · Make  
Projekt · Starten  
Projekt · Debuggen  
Makefile auswählen  
Leere Funktion einfügen  
Leere Struktur einfügen  
Faltmarkierungen einfügen  
Kommentieren  
Anderes Element einfügen  
Fehler und Warnungen  
Eigenes Fenster öffnen  
Dokumentation  
Compiler wechseln  
SAS/C beenden

SIEHE AUCH:

Der SAS/C mode    Andere Compiler

## 1.73 SAS/C Symbolleiste

ENGLISH DEUTSCH

SAS/C Toolbar

Neues Dokument  
Make  
Einstellungen

Nächster Fehler  
Starten  
Debugger  
Neues Projekt  
Hilfe

SIEHE AUCH:

Der SAS/C-Modus    Andere Compiler

## 1.74 SAS/C Tastatur

ENGLISH DEUTSCH

SAS/C-Tastenbelegung

F2    Compileroptionen  
F3    Make  
F4    Debug mit Shell-Startup  
S-F4   Debug mit Workbench-Startup  
F6    Programm starten  
F7    SCMSG-Browser zeigen  
S-F7   SCMSG-Browser verbergen  
F8    Hilfe  
F9    SAS/C beenden

SIEHE AUCH:

Der SAS/C-Modus    Andere Compiler

## 1.75 Templates

ENGLISH DEUTSCH

TEMPLATES

In der C/C++-Umgebung stehen verschiedene Templates bereit, mit denen typische C/C++-Elemente einfach – durch die Eingabe weniger Buchstaben – in den Quelltext eingefügt werden können. Geben Sie beispielsweise "for(" ein, dann wird eine komplette leere for-Scheife eingefügt. Folgende Templates sind abrufbar:

```
if(  
switch(  
for(  
else{  
while(  
do{
```

## 1.76 vbcc

ENGLISH DEUTSCH

Beschreibung der vbcc-Variante der C/C++-Umgebung

vbcc:

- Menü
- Symbolleiste
- Tastatur
- Templates

Hinweise:

Die vbcc-Umgebung ist auf vbcc 0.8 ausgerichtet; die Kompatibilität zu anderen Versionen wurde nicht getestet.

SIEHE AUCH:

Andere Compiler

## 1.77 vbcc-Menü

ENGLISH DEUTSCH

vbcc Menü

- Neues Projekt anlegen
- Makefile erzeugen
- Einstellungen
- Projekt · Make
- Projekt · Starten
- Makefile auswählen
- Leere Funktion einfügen
- Leere Struktur einfügen
- Faltmarkierungen einfügen
- Kommentieren
- Anderes Element einfügen
- Fehler erklären
- Dokumentation
- Compiler wechseln

SIEHE AUCH:

Der vbcc-Modus    Andere Compiler

## 1.78 vbcc-Symbolleiste

ENGLISH DEUTSCH

vbcc-Symbolleiste

---

Neues Dokument  
Make  
Einstellungen  
Starten  
Neues Projekt  
Hilfe

SIEHE AUCH:

Der vbcc-Modus    Andere Compiler

## 1.79 GeekGadgets (gcc)

ENGLISH DEUTSCH

Beschreibung der GeekGadget-(gcc)-Variante der C/C++-Umgebung

GeekGadget (gcc)

Menü  
Symbolleiste  
Tastatur  
Templates

Hinweise:

Die GeekGadget-(gcc)-Umgebung ist auf 2.95.3-4 für AmigaOS/m68K ausgerichtet. Die Kompatibilität zu anderen gcc-Versionen und CPU-Familien wurde nicht getestet.

SIEHE AUCH:

Andere Compiler

## 1.80 GeekGadget-(gcc)-Menü

ENGLISH DEUTSCH

GeekGadget-(gcc)-Menü

Neues Projekt anlegen  
Makefile erzeugen  
Compiler wechseln  
Einstellungen  
Projekt · Make  
Projekt · Starten  
Projekt · Debuggen  
Makefile auswählen  
Leere Funktion einfügen  
Leere Struktur einfügen  
Faltmarkierungen einfügen



Kommentieren  
Anderes Element einfügen  
Fehler erklären  
Dokumentation  
Compiler wechseln

SIEHE AUCH:

Der GeekGadget (gcc)-Modus    Andere Compiler

## 1.81 GeekGadget-(gcc)-Symbolleiste

ENGLISH DEUTSCH

GeekGadget-(gcc)-Symbolleiste

Neues Dokument  
Make  
Einstellungen  
Starten  
Debuggen  
Neues Projekt  
Hilfe

SIEHE AUCH:

Der GeekGadget (gcc)-Modus    Andere Compiler

## 1.82 GeekGadget-(gcc)-Tastatur

ENGLISH DEUTSCH

GeekGadget-(gcc)-Tastenbelegung

F2    Compileroptionen  
F3    Make  
F4    Debuggen  
F6    Programm starten  
F8    Hilfe

SIEHE AUCH:

Der GeekGadget (gcc)-Modus    Andere Compiler

## 1.83 vbcc-Tastatur

ENGLISH DEUTSCH

vbcc-Tastenbelegung

---

F2 Compileroptionen  
F3 Make  
F6 Programm starten  
F8 Hilfe

SIEHE AUCH:

Der vbcc-Modus    Andere Compiler

## 1.84 StormC

ENGLISH DEUTSCH

Beschreibung der StormC-Variante der C/C++-Umgebung

StormC:

Menü  
Symbolleiste  
Tastatur  
Templates

Hinweis:

Die ersten Schritte bei Verwendung des StormC-Modus sind normalerweise das Starten der StormC-Umgebung und anschließend das Laden oder Erstellen eines Projektes. Die meisten StormC-bezogenen Funktionen in den Menüs und Symbolleisten stehen nicht zur Verfügung (oder bleiben wirkungslos), bis ein Projekt geladen wurde.

SIEHE AUCH:

Start StormC Projekt öffnen    Andere Compiler

## 1.85 StormC-Menü

ENGLISH DEUTSCH

StormC Menü

StormC-Umgebung starten  
Fenster zeigen  
Projektmanagement  
Projekt · Make  
Projekt · Modul übersetzen  
Projekt · Starten  
Projekt · Debuggen  
Neu übersetzen  
Hierarchisches Make  
Leere Funktion einfügen  
Leere Struktur einfügen

Faltmarkierungen einfügen  
kommentieren  
Anderes Element einfügen  
Fehler und Warnungen  
Dokumentation  
Compiler wechseln

SIEHE AUCH:

Der StormC-Modus    Andere Compiler

## 1.86 StormC-Symbolleiste

ENGLISH DEUTSCH

StormC Toolbar

Neues Dokument  
Make  
Nächster Fehler  
Starten  
Debugger  
Position zeigen  
Go  
Einzelschritt  
Einzelschritt über  
Weiter bis Ende  
Neues Projekt  
Hilfe

SIEHE AUCH:

Der StormC-Modus    Andere Compiler

## 1.87 StormC-Tastatur

ENGLISH DEUTSCH

StormC-Tastenbelegung

F1	Neues Fenster
F2	Datei öffnen
F3	Alle Texte speichern
F4	Neues Projekt anlegen
F5	Projekt öffnen
F6	Projekt speichern
F7	Compilieren
F8	Make
F9	Programm starten
F10	Debugger starten
S-F1	Zeige aktuelle Variablen
S-F2	Zeige globale Variablen

S-F3    Zeige beobachtete Variablen  
S-F4    Zeige Register  
S-F6    Debugger: Weiter  
S-F7    Debugger: Einzelschritt  
S-F8    Debugger: Einzelschritt über  
S-F9    Debugger: Weiter bis Funktionsende  
S-F10   Position zeigen

SIEHE AUCH:

Der StormC-Modus    Andere Compiler

## 1.88    Generische C/C++-Umgebung

ENGLISH    DEUTSCH

Beschreibung der generischen Variante der C/C++-Umgebung

Generische C/C++-Umgebung:

Menü  
Symbolleiste  
Tastatur  
Templates

Hinweise:

Die generische Umgebung steht für einen benutzerdefinierten C/C++-Compiler zur Verfügung. Sie kann mit fast jedem Compiler verwendet werden, allerdings müssen einige Parameter passend konfiguriert werden (Sie werden automatisch zur Konfiguration aufgefordert, sobald erstmals Funktionen wie Neues Projekt anlegen verwendet werden.

StormC-Anwender/innen können die generische Umgebung verwenden, wenn sie den Compiler auf Basis eines klassischen Makefiles ohne die StormC-Entwicklungsumgebung (die vom der StormC-Variante der C/C++-Umgebung adressiert wird) verwenden möchten.

SIEHE AUCH:

Andere Compiler

## 1.89    C/C++-Menü

ENGLISH    DEUTSCH

C/C++ Menü

Neues Projekt anlegen  
Makefile erzeugen

---

- Projekt · Make
- Projekt · Starten
- Makefile auswählen
- Leere Funktion einfügen
- Leere Struktur einfügen
- Faltmarkierungen einfügen
- Kommentieren
- Anderes Element einfügen
- Dokumentation
- Compiler wechseln

#### Hinweis

Die generische C/C++-Umgebung versucht den verwendeten Compiler anhand der Dateien im Projektverzeichnis zur erkennen. Wenn sich dort beispielsweise die Datei "makefile" befindet, dann wird mit 'GNU make' übersetzt. Andererseits wird smake aus dem SAS/C-Paket zusammen mit jedem "smakefile" und dmake aus dem Dice-Paket mit jedem "dmakefile" verwendet.

#### SIEHE AUCH:

Der generische C/C++-Modus    Andere Compiler

## 1.90 C/C++-Symbolleiste

ENGLISH DEUTSCH

Generische C/C++-Symbolleiste

- Neues Dokument
- Make
- Starten
- Neues Projekt
- Hilfe

#### SIEHE AUCH:

Der generische C/C++-Modus    Andere Compiler

## 1.91 C/C++-Tastatur

ENGLISH DEUTSCH

Tastenbelegung

- F3    Make
- F6    Programm starten
- F8    Hilfe

#### SIEHE AUCH:

---

Der generische C/C++-Modus    Andere Compiler

## 1.92 Templates

ENGLISH   DEUTSCH

FUNKTION

Leeres Programm (ISO C)  
Leeres Amiga-Programm

BESCHREIBUNG

Diese Funktionen erzeugen einen neuen (funktional leeren)  
Quelltext auf Basis der mitgelieferten Muster.

## 1.93 Neues Projekt anlegen

ENGLISH   DEUTSCH

FUNKTION

Neues Projekt anlegen

BESCHREIBUNG

Mit dieser Funktion wird ein neues Projekt angelegt: ein Verzeichnis wird erstellt, ein leerer Quelltext wird hinzugefügt und ein passendes Makefile wird generiert. Anschließend kann das Projekt mit den Make-Funktionen in Symbolleiste und Menü übersetzt werden.

Vor dem ersten Übersetzen sollten die Compileroptionen kontrolliert und angepasst werden. Wird Ihr Compiler mit einem Konfigurationsprogramm ausgeliefert (SAS/C) oder wird ein Konfigurationsprogramm von GoldED's C/C++ Umgebung bereitgestellt (cprefs für gcc und vbcc), dann verwenden Sie bitte dieses Programm. Die Optionen dürfen in diesem Fall nicht in das Makefile eingetragen werden, da es sonst zu Konflikten kommen kann. Bei Verwendung anderer Compiler müssen die Optionen dagegen direkt in das Makefile eingetragen werden.

SIEHE AUCH:

Makefile erzeugen

## 1.94 Makefile erzeugen

---

ENGLISH DEUTSCH

FUNKTION

Makefile erzeugen

BESCHREIBUNG

Generiert ein Makefile für einen Ordner, der C/C++ und/oder Assembler-Quelltexte enthält. Die C/C++-Quelltexte und die von ihnen eingebundenen Header werden untersucht und Abhängigkeiten entsprechend in das Makefile übernommen.

SIEHE AUCH:

Neues Projekt anlegen

## 1.95 Starten

ENGLISH DEUTSCH

FUNKTION

Starten

BESCHREIBUNG

Startet das übersetzte Programm.

Hinweis: diese Funktion lokalisiert das Programm, indem die Einstellungen-Datei und/oder das Makefile nach einem Programmnamen durchsucht werden. Im Makefile sollte es deshalb eine Zuweisung der Form "EXE = <executable>" geben. Dies ist automatisch der Fall, wenn Sie Makefiles verwenden, die von der C/C++-Umgebung erzeugt wurden.

SIEHE AUCH:

Projekt/Make

## 1.96 Make

ENGLISH DEUTSCH

FUNKTION

Make

BESCHREIBUNG

Startet den Übersetzungs-Vorgang für das aktuelle Projekt.

---

Die im Makefile angeführten Quelltexte werden entsprechend der dort niedergelegten Regeln übersetzt und gelinkt. Anschliessend kann das Programm gestartet werden.

#### HINWEISE

Sie sollten eine Zuweisung der Form "EXE = <executable>" im Makefile haben, damit die anderen Funktionen der C/C++-Umgebung den Programmnamen erkennen können. Dies ist automatisch der Fall, wenn Sie Makefiles verwenden, die von der C/C++-Umgebung erzeugt wurden.

#### SAS/C:

Die 'make'-Funktionen in Toolbar und Menü verwenden zum Übersetzen entweder das 'smake'-Programm aus dem SAS/C-Paket (wenn ein "smakefile" im Verzeichnis gefunden wird) oder GNU 'make' (wenn ein "makefile" vorhanden ist). Sie können für manuell erzeugte Projekte das Programm verwenden, das Ihnen besser gefällt. Die mit der C/C++-Umgebung erzeugten Projekte verwenden immer GNU 'make'.

Die Zeichenkette "sc:" (der Pfad zum SAS/C-Compiler) sollte im Makefile auftauchen, wenn Sie ein "makefile" verwenden, damit die C/C++-Umgebung erkennt, daß SAS/C verwendet wird und automatisch den Message-Browser vor dem Übersetzen startet.

Siehe auch: Einstellungen

#### vbcc:

Die Zeichenkette "vbcc" (der Compilername) sollte im Makefile auftauchen, damit die C/C++-Umgebung erkennt, daß vbcc verwendet wird.

Siehe auch: Einstellungen

#### GeekGadgets (gcc):

Die Zeichenkette "gcc" (der Compilername) sollte im Makefile auftauchen, damit die C/C++-Umgebung erkennt, daß gcc verwendet wird.

Siehe auch: Einstellungen

#### Dice:

Die 'make'-Funktionen in Toolbar und Menü verwenden zum Übersetzen entweder das 'dmake'-Programm aus dem Dice-Paket (wenn ein "dmakefile" im Verzeichnis gefunden wird) oder GNU 'make' (wenn ein "makefile" vorhanden ist). Sie können für manuell erzeugte Projekte das Programm verwenden, das Ihnen besser gefällt. Die mit der C/C++-Umgebung erzeugten Projekte verwenden immer GNU 'make'.

---



Die Zeichenkette "dcc:" (der Pfad zum Dice-Compiler) sollte im Makefile auftauchen, wenn Sie ein "makefile" verwenden, damit die C/C++-Umgebung erkennt, daß Dice verwendet wird.

## 1.97 Makefile auswählen

ENGLISH DEUTSCH

FUNKTION

Makefile auswählen

BESCHREIBUNG

Mit diesem Menüpunkt kontrollieren Sie, ob vor dem Make-Vorgang das Makefile abgefragt wird oder das Makefile automatisch ausgewählt wird.

## 1.98 Leere Funktion einfügen

ENGLISH DEUTSCH

FUNKTION

Leere Funktion einfügen

BESCHREIBUNG

Dieses Makro fügt eine leere C-Funktion in den Text ein. Optional werden Faltmarkierungen erzeugt und ein Kommentar vorangestellt.

SIEHE AUCH:

Faltmarkierungen einfügen  
kommentieren

## 1.99 Leere Struktur einfügen

ENGLISH DEUTSCH

FUNKTION

Leere Struktur einfügen

BESCHREIBUNG

Dieses Makro fügt eine leere C-Strukturdefinition in den Text

---

ein. Optional werden Faltmarkierungen erzeugt und ein Kommentar vorangestellt.

SIEHE AUCH:

Faltmarkierungen einfügen  
kommentieren

## 1.100 Faltmarkierungen einfügen

ENGLISH DEUTSCH

FUNKTION

Faltmarkierungen einfügen

BESCHREIBUNG

Mit diesem Menüpunkt kontrollieren Sie, ob beim Einfügen leerer Funktionen und Strukturen Faltmarkierungen hinzugefügt werden.

SIEHE AUCH:

Leere Funktion einfügen  
Leere Struktur einfügen

## 1.101 Kommentieren

ENGLISH DEUTSCH

FUNKTION

Kommentieren

BESCHREIBUNG

Mit diesem Menüpunkt kontrollieren Sie, ob beim Einfügen leerer Funktionen und Strukturen Kommentare hinzugefügt werden.

SIEHE AUCH:

Leere Funktion einfügen  
Leere Struktur einfügen

## 1.102 Anderes Element einfügen

---

ENGLISH DEUTSCH

FUNKTION

Einfügen ... switch  
Einfügen ... for  
Einfügen ... if  
Einfügen ... else  
Einfügen ... do  
Einfügen ... while

BESCHREIBUNG

Diese Funktionen fügen leere switch/for/if/else/do/while  
Blöcke in den Text ein.

SIEHE AUCH:

Templates

## 1.103 Dokumentation

ENGLISH DEUTSCH

FUNKTION

GoldED Studio und C/C++  
ISO C Kurzreferenz  
GNU Make

BESCHREIBUNG

Mit diesen Funktionen erhalten Sie schnellen Zugriff auf die  
mitgelieferte Dokumentation.

## 1.104 Hilfe

ENGLISH DEUTSCH

FUNKTION

Hilfe (Knopf in der Symbolleiste)

BESCHREIBUNG

Dieses Dokument anzeigen.

## 1.105 Compiler wechseln

---

ENGLISH DEUTSCH

FUNKTION

Compiler wechseln · SAS/C  
Compiler wechseln · vbcc  
Compiler wechseln · GeekGadgets (gcc)  
Compiler wechseln · StormC  
Compiler wechseln · Anderer Compiler

BESCHREIBUNG

Diese Funktionen schalten auf eine andere Variante der C/C++-Umgebung mit jeweils eigenen Symbolleisten, Menüs usw. um (sofern die gewählte Variante installiert wurde).

StormC-Anwender/innen können die generische Umgebung ("Anderer Compiler") verwenden, wenn sie den Compiler auf Basis eines klassischen Makefiles ohne die StormC-Entwicklungsumgebung (die vom der StormC-Variante der C/C++-Umgebung adressiert wird) verwenden möchten.

## 1.106 Einstellungen

ENGLISH DEUTSCH

FUNKTION

Einstellungen

BESCHREIBUNG

Mit dieser Funktion können Sie Compiler- und Linkereinstellungen vornehmen. Bitte wenden Sie sich für eine detaillierte Beschreibung an die SAS/C-Dokumentation. Kurze Erläuterungen zu einzelnen Eingabefeldern erhalten Sie, indem Sie den Mauszeiger über das Eingabefeld bewegen und die HELP-Taste drücken.

## 1.107 GST-Datei erzeugen

ENGLISH DEUTSCH

FUNKTION

GST-Datei erzeugen"

BESCHREIBUNG

Erzeugt eine vorübersetzte Headerdatei, kurz GST-Datei. Vorübersetzte Header werden üblicherweise dazu benutzt, den

Übersetzungsvorgang großer Projekte zu beschleunigen. Details zu diesem Thema finden Sie in der SAS-Dokumentation. Die GST-Datei wird durch Übersetzen der Datei "gst.c" (oder, falls nicht vorhanden, "main.c") erzeugt. Mit anderen Worten: es sollte eine Datei "gst.c" im Projektverzeichnis geben und diese Datei solle alle vom gesamten Projekt benötigten Header einbinden. Abgesehen von den dafür notwendigen #include-Statements sollte "gst.c" leer sein und auch nicht im Makefile auftauchen (da diese Datei nur zum Erzeugen der GST-Datei benötigt nicht zum Programm gehört).

Hinweis: um ein Projekt auf vorübersetzte Headern umzustellen muß der Name der GST-Datei (frei wählbar, oft "all.gst") vorab im SAS/C-Konfigurationsprogramm auf der Compiler-Seite im Textfeld 'GST' eingetragen werden.

Tip: Für größere Projekte sollte eine übergeordnete Headerdatei (z.B. "defs.h") erstellt werden, die alle anderen Header einbindet und die ihrerseits von jedem am Projekt beteiligten Quelltext eingebunden wird. Unter diesen Umständen würde "gst.c" nur aus der Zeile '#include "defs.h"' bestehen.

SIEHE AUCH:

SAS/C options

## 1.108 Projekt · Debuggen

ENGLISH DEUTSCH

FUNKTION

Debuggen (Knopf in der Symbolleiste)  
Debuggen als CLI-Programm  
Debuggen als WB-Programm

BESCHREIBUNG

Diese Funktionen starten den SAS/C-Debugger und führen das übersetzte Programm im Debugger aus. Das in der Regel nur sinnvoll, wenn das Projekt mit aktivierten Debug-Optionen übersetzt worden ist (Menu Einstellungen, Seite "Compiler"). Der Programmstart erfolgt entweder als CLI-Programm oder es wird ein WB-Start simuliert (im letzteren Fall muß ein Icon für das Executable existieren, ansonsten bricht der Debugger ab).

SIEHE AUCH:

Einstellungen

---

## 1.109 Fehler und Warnungen

ENGLISH DEUTSCH

FUNKTION

Browser zeigen  
Browser verbergen  
Meldung zeigen · Erste  
Meldung zeigen · Vorherige  
Meldung zeigen · Aktuelle  
Meldung zeigen · Nächste  
Meldung zeigen · Letzte  
Meldung löschen  
Alle Meldungen löschen

## 1.110 Fehler und Warnungen - Zeigen/Verbergen

ENGLISH DEUTSCH

FUNKTION

Browser zeigen  
Browser verbergen

BESCHREIBUNG

Diese Funktionen zeigen bzw. verbergen das Fenster des SAS/C-Message-Browsers, in dem Fehlermeldungen und Warnungen, die beim Übersetzen anfallen, angezeigt werden. Beachten Sie, das das Verbergen dieses Fensters eine andere Funktion als das normale Schließen ist, da die Meldungen dabei nicht gelöscht werden. Wenn Sie es vorziehen, daß das Fenster automatisch beim Übersetzen des Projektes erscheint, dann aktivieren Sie den Menüpunkt "Eigenes Fenster öffnen".

SIEHE AUCH:

Eigenes Fenster öffnen

## 1.111 Fehler und Warnungen - Navigation

ENGLISH DEUTSCH

FUNKTION

Meldung zeigen · Erste  
Meldung zeigen · Vorherige  
Meldung zeigen · Aktuelle  
Meldung zeigen · Nächste  
Meldung zeigen · Letzte

---

## BESCHREIBUNG

Diese Navigationsfunktionen zeigen die mit der ersten/vorherigen/nächsten/letzten Meldung verbundene Textstelle an. Gleichzeitig erscheint eine Beschreibung des Problems im Fenster des SAS/C-Message-Browsers, sofern dessen Fenster geöffnet ist.

## SIEHE AUCH:

Browser zeigen  
Browser verbergen  
Eigenes Fenster öffnen

## 1.112 Fehler und Warnungen - Löschen

ENGLISH DEUTSCH

## FUNKTION

Meldung löschen  
Alle Meldungen löschen

## BESCHREIBUNG

Diese Funktionen löschen die aktuelle Meldung bzw. alle Meldungen, die im Fenster des SAS/C-Message-Browsers angezeigt werden.

## SIEHE AUCH:

Browser zeigen

## 1.113 Eigenes Fenster öffnen

ENGLISH DEUTSCH

## FUNKTION

Eigenes Fenster öffnen

## BESCHREIBUNG

Mit dieser Funktion wird festgelegt, ob beim Start des Übersetzungsvorgangs automatisch das Fenster des SAS/C-Message-Browsers erscheint oder ob der Browser im Hintergrund ohne Fenster arbeitet. Die Fehler und Warnungen können auch ohne sichtbares Browser-Fenster mit der Funktion "Nächster Fehler" in der Symbolleiste abgerufen werden.

## SIEHE AUCH:

Fehler und Warnungen  
Show next error

## 1.114 Nächsten Fehler anzeigen

ENGLISH DEUTSCH

FUNKTION

Nächsten Fehler anzeigen (Knopf in der Symbolleiste)

BESCHREIBUNG

Zeigt die mit dem nächsten Fehler verbundene Textstelle an. Gleichzeitig erscheint eine Beschreibung des Problems im Fenster des SAS/C-Message-Browsers, sofern dessen Fenster geöffnet ist.

SIEHE AUCH:

Eigenes Fenster öffnen

## 1.115 Dokumentation

ENGLISH DEUTSCH

FUNKTION

GoldED Studio und C/C++  
SAS/C® Inhalt  
SAS/C® Copileroptionen  
SAS/C® Browser  
SAS/C® Debugger  
SAS/C® Bibliotheken  
SAS/C® Troubleshooting  
ISO C Kurzreferenz  
GNU Make  
GST-Symbol nachschlagen...  
SAS/C-Funktion erklären  
SAS/C-Funktion erklären...

BESCHREIBUNG

Mit diesen Funktionen erhalten Sie schnellen Zugriff auf die verfügbare Dokumentation.

## 1.116 Dokumentation - GST-Symbol nachschlagen



ENGLISH DEUTSCH

FUNKTION

GST-Symbol nachschlagen

BESCHREIBUNG

Mit dieser Funktion können Informationen abgerufen werden, die in der aktuellen GST-Datei gespeichert sind. Der Begriff "GST-Datei" bezeichnet eine vorab übersetzte Header-Datei (siehe "GST-Datei erzeugen").

SIEHE AUCH:

GST-Datei erzeugen

## 1.117 Dokumentation - SAS/C-Funktion erklären

ENGLISH DEUTSCH

FUNKTION

SAS/C-Funktion erklären  
SAS/C-Funktion erklären...

BESCHREIBUNG

Diese Funktionen liefern Erklärungen zu Funktionen aus den Standard-C-Bibliotheken (wie malloc), die Bestandteil von SAS/C sind.

## 1.118 SAS/C beenden

ENGLISH DEUTSCH

FUNKTION

SAS/C beenden

BESCHREIBUNG

Beendet den SAS/C-Messagebrowser, schließt offene Hilfstexte und entfernt den Compiler aus dem Speicher.

SIEHE AUCH:

SAS/C message browser

---

## 1.119 Einstellungen

ENGLISH DEUTSCH

FUNKTION

Einstellungen

BESCHREIBUNG

Diese Funktion startet das cprefs-Programm, mit dem die Compiler- und Linkeroptionen eingestellt werden können. Die Einstellungen werden in der Datei "vbcc.options" im Projektverzeichnis gespeichert. Eine Beschreibung von cprefs und Hinweis zur Verwendung der Konfigurationsdateien in der Compiler-Kommandozeile oder in Makefiles finden Sie im cprefs-Guide.

## 1.120 Fehler erklären

ENGLISH DEUTSCH

FUNKTION

Fehler erklären

BESCHREIBUNG

Diese Funktion erklärt einen numerischen vbcc-Fehlercode.

SIEHE AUCH:

Dokumentation

## 1.121 Dokumentation

ENGLISH DEUTSCH

FUNKTION

GoldED Studio und C/C++  
Fehler und Warnungen  
vbccprefs  
ISO C Kurzreferenz  
GNU Make

BESCHREIBUNG

Mit diesen Funktionen erhalten Sie schnellen Zugriff auf die mitgelieferte Dokumentation.

---

## 1.122 Compiler wechseln

ENGLISH DEUTSCH

FUNKTION

gcc (ANSI C)  
g++ (C++)

BESCHREIBUNG

Mit diesen Menüpunkten können Sie eines der beiden gcc-Frontends auswählen. Der gewählte Compiler wird im Makefile eingetragen, indem die Zeichenfolge "<tab>gcc" gegen "<tab>g++" ausgetauscht wird. Makefiles, die nicht mit der C/C++-Umgebung erstellt wurden, müssen manuell bearbeitet werden, da diese Zeichenfolgen dort normalerweise nicht vorhanden sind.

## 1.123 Projekt · Debuggen

ENGLISH DEUTSCH

FUNKTION

Debuggen (Knopf in der Symbolleiste)  
Debuggen (Menü)

BESCHREIBUNG

Diese Funktionen starten den GCC/C-Debugger und führen das übersetzte Programm im Debugger aus. Das in der Regel nur sinnvoll, wenn das Projekt mit aktivierten Debug-Optionen übersetzt worden ist (Menu Einstellungen, Seite "Debugger Options" im gccprefs-Programm).

SIEHE AUCH:

Einstellungen

## 1.124 Einstellungen

ENGLISH DEUTSCH

FUNKTION

Einstellungen

BESCHREIBUNG

Diese Funktion startet das cprefs-Programm, mit dem die Compiler- und Linkeroptionen eingestellt werden können. Die

---

Einstellungen werden in der Datei "gcc.options" im Projektverzeichnis gespeichert. Eine Beschreibung von cprefs und Hinweis zur Verwendung der Konfigurationsdateien in der Compiler-Kommandozeile oder in Makefiles finden Sie im cprefs-Guide.

## 1.125 Dokumentation

ENGLISH DEUTSCH

FUNKTION

GoldED Studio und C/C++  
gcc  
gcc und AmigaOS  
Anhang: Libnix  
Anhang: gccprefs  
ISO C Kurzreferenz  
GNU Make

BESCHREIBUNG

Mit diesen Funktionen erhalten Sie schnellen Zugriff auf die mitgelieferte Dokumentation.

## 1.126 StormC-Umgebung starten

ENGLISH DEUTSCH

FUNKTION

StormC-Umgebung starten

BESCHREIBUNG

Diese Funktion startet die SAS/C-Entwicklungsumgebung. Die nächste Schritt ist normalerweise das Laden oder Erstellen eines Projektes. Die meisten StormC-bezogenen Funktionen in den Menüs und Symbolleisten stehen nicht zur Verfügung (oder bleiben wirkungslos), bis ein Projekt geladen wurde.

SIEHE AUCH:

Andere Compiler

## 1.127 Fenster zeigen

ENGLISH DEUTSCH

FUNKTION

---

Fenster zeigen · Module  
Fenster zeigen · Fehler  
Fenster zeigen · Variablen  
Fenster zeigen · Variablen, Seite "aktuelle Variablen"  
Fenster zeigen · Variablen, Seite "globale Variablen"  
Fenster zeigen · Variablen, Seite "beobachtete Variablen"  
Fenster zeigen · Variablen, Seite "Register"  
Fenster zeigen · History  
Fenster zeigen · Breakpunkte  
Fenster zeigen · Stack  
Fenster zeigen · Profiler-Protokoll  
Fenster zeigen · Hex-Editor  
Fenster zeigen · Disassembler

#### BESCHREIBUNG

Diese Funktionen öffnen verschiedene von StormC bereitgestellte Fenster. Beachten Sie, daß einige Fenster nur zur Verfügung stehen, wenn ein Projekt geöffnet wurde, ein Programm übersetzt wurde oder der Debugger gestartet wurde.

## 1.128 Projektmanagement

ENGLISH DEUTSCH

#### FUNKTION

Projekt öffnen...  
Neues Projekt anlegen...  
Dokument zum Projekt hinzufügen

#### BESCHREIBUNG

Mit diesen Funktionen werden StormC-Projekte geöffnet, angelegt oder erweitert. Beachten Sie, daß viele Funktionen von StormC projektorientiert sind und erst arbeiten, nachdem ein Projekt geladen wurde. Beispielsweise kann ein Text nicht übersetzt werden, bis das Projekt geöffnet wird, zu dem der Text gehört. Weitere Information zu StormC-Projekten finden Sie in der Copilerdokumentation.

## 1.129 Projekt · Make

ENGLISH DEUTSCH

#### FUNKTION

Projekt · Make

#### BESCHREIBUNG

---

Startet den Übersetzungs-Vorgang für das aktuelle Projekt. Die in der Projektkonfiguration eingetragenen Dateien werden übersetzt und gelinkt. Anschließend kann das Programm gestartet werden.

Beachten Sie, daß die Art des Make-Prozesses davon abhängt, ob die folgenden Menüpunkte aktiviert sind:

Neu übersetzen  
Hierarchisches Make

## 1.130 Projekt · Modul übersetzen

ENGLISH DEUTSCH

FUNKTION

Projekt · Modul übersetzen

BESCHREIBUNG

Die aktuelle (im Fenster dargestellte) Datei wird übersetzt. Beachten Sie, daß aufgrund der projektzentrischen Ausrichtung von StormC dieser Vorgang nur ausgeführt werden kann, wenn ein Projekt geöffnet wurde und die Datei zu diesem Projekt gehört.

SIEHE AUCH:

Projekt öffnen

## 1.131 Starten

ENGLISH DEUTSCH

FUNKTION

Starten

BESCHREIBUNG

Startet das übersetzte Porgramm.

SIEHE AUCH:

Projekt · Make

## 1.132 Projekt · Debug

---

ENGLISH DEUTSCH

## FUNKTION

Debugger starten  
Position zeigen  
Weiter  
Einzelschritt  
Einzelschritt über  
Weiter bis Funktionsende  
Breakpunkte speichern  
Adresse suchen

## BESCHREIBUNG

<Debugger starten> startet den StormC-Debugger und führt das übersetzte Programm in dessen Umgebung aus (die mit dem Debuggen zusammenhängenden Compiler-Optionen sollten vorab in der StormC-Compiler-Konfiguration eingestellt werden). Mit den angeführten Navigationsfunktionen wird der Ablauf im Debugger gesteuert.

## 1.133 Navigation im Debugger

ENGLISH DEUTSCH

## FUNKTION

Position zeigen  
Weiter  
Einzelschritt  
Einzelschritt über  
Weiter bis Funktionsende

## BESCHREIBUNG

Die angeführten Navigationsfunktionen kontrollieren im Zusammenspiel mit den Breakpunkten den Ablauf im StormC-Debugger (die Breakpunkte können über das Ansicht-Menü des Editors ein- bzw. ausgeblendet werden).

Position zeigen

Zeigt die Position im Quelltext

Weiter

Der Ablauf wird bis zum nächsten aktivierten Breakpunkt fortgesetzt. Beachten Sie, daß Breakpunkte nur an den Stellen ein- oder ausgeschaltet werden können, die der Compiler dafür freigibt. Sie können mit dem Editor keine zusätzlichen Breakpunkte erstellen.

Einzelschritt

---

Setzt den Ablauf bis zum nächstmöglichen Unterbrechungspunkt fort, unabhängig davon, ob sich an dieser Stelle ein aktivierter Breakpunkt befindet oder nicht (normalerweise ist das die nächste Zeile im Quelltext). Befindet sich an der aktuellen Position eine Funktion, dann betritt der Debugger die Funktion (siehe auch: Einzelschritt über).

#### Einzelschritt über

Setzt den Ablauf bis zum nächstmöglichen Unterbrechungspunkt fort, unabhängig davon, ob sich an dieser Stelle ein aktivierter Breakpunkt befindet oder nicht. Befindet sich an der aktuellen Position eine Funktion, dann führt der Debugger die Funktion aus, betritt die Funktion aber nicht mit Einzelschritten (siehe auch: Einzelschritt)

#### Weiter bis Funktionsende

Der Ablauf wird bis zum Ende der Funktion fortgesetzt.

SIEHE AUCH:

Debugger starten

## 1.134 Breakpunkte speichern

ENGLISH DEUTSCH

FUNKTION

Breakpunkte speichern

BESCHREIBUNG

Diese Funktion speichert den aktuellen Zustand der Breakpunkte (gesetzt/ungesetzt), so daß der Debug-Vorgang später mit demselben Zustand fortgesetzt werden kann.

## 1.135 Adresse suchen

ENGLISH DEUTSCH

FUNKTION

Adresse suchen

BESCHREIBUNG

Lokalisiert eine Speicheradresse in vom Debugger geladenen Programm.



## 1.136 Neu übersetzen

ENGLISH DEUTSCH

FUNKTION

Neu übersetzen

BESCHREIBUNG

Dieser Menüpunkt legt fest, ob beim Make-Vorgang nur die veränderten Quellen oder alle Module übersetzt werden.

SIEHE AUCH:

Make

## 1.137 Hierarchisches Make

ENGLISH DEUTSCH

FUNKTION

Hierarchisches Make

BESCHREIBUNG

Dieser Menüpunkt legt fest, ob das normale Make oder ein hierarchisches Make verwendet wird.

SIEHE AUCH:

Make

## 1.138 Fehler und Warnungen

ENGLISH DEUTSCH

FUNKTION

StormC-Browser zeigen  
Meldung zeigen · Nächste  
Meldung zeigen · Vorherige

BESCHREIBUNG

Die Funktionen "Nächste Meldung" und "Vorherig Meldung" zeigen die entsprechenden Problemstellen im Quelltext. Gleichzeitig erscheint eine Beschreibung des Problems im Fenster des StormC-Fehler-Browsers, sofern dessen Fenster geöffnet ist.