

Den frei verfügbaren Code::Blocks-Compiler installieren

Einmal ein Programm starten

Kapitel 1

In zehn Schritten zum ersten Compiler

Heutzutage vereinigen Entwicklertools Compiler und Editor in einem einzigen Paket – einer Entwicklungsumgebung (IDE). Nachdem Sie mit der Eingabe des Programms fertig sind, müssen Sie nur auf einen Knopf drücken, um das fertige Programm zu erzeugen und zu starten.

Glücklicherweise gibt es frei verfügbare C-Entwicklungsumgebungen. Sie sehen in diesem Kapitel, wie eine dieser Umgebungen – Code::Blocks – installiert wird. Die Version ist für alle aktuellen Windows-Versionen, Linux und Mac OS X erhältlich und nutzt den sogenannten GNU-Compiler.

Die Installation der IDE

In diesem Kapitel wird die Installation auf einem Windows-System gezeigt. Der grundsätzliche Ablauf für die anderen Systeme ist vergleichbar.

Die IDE und die Website sind in englischer Sprache. Sie werden damit häufiger konfrontiert werden, dass viele Programmierertools am besten auf Englisch funktionieren. Das ist so.

- ✓ Öffnen Sie die Webseite <http://www.codeblocks.org/downloads>, wählen Sie die *binary release* aus. Aus der Liste suchen Sie die neueste Version aus, die typischerweise am obersten Rand der Liste steht.



Auch wenn Englisch nicht gerade Ihre Lieblingssprache ist und es die Entwicklungsprogramme sogar auf Deutsch gäbe – die Masse der Programmierer arbeitet weltweit auf Englisch. Wenn Sie eine Fehlermeldung erhalten und versuchen zu verstehen, was hier los ist – eine Google-Suche mit dem englischen Fehler-text wird Sie schneller zur Lösung bringen, als wenn Sie den deutschen Fehler-text verwenden.

Code::Blocks installieren

Navigieren Sie zu der gespeicherten Datei und starten Sie sie, um die Umgebung automatisch zu installieren.

Bestätigen Sie den Begrüßungsbildschirm mit einem Klick auf NEXT.

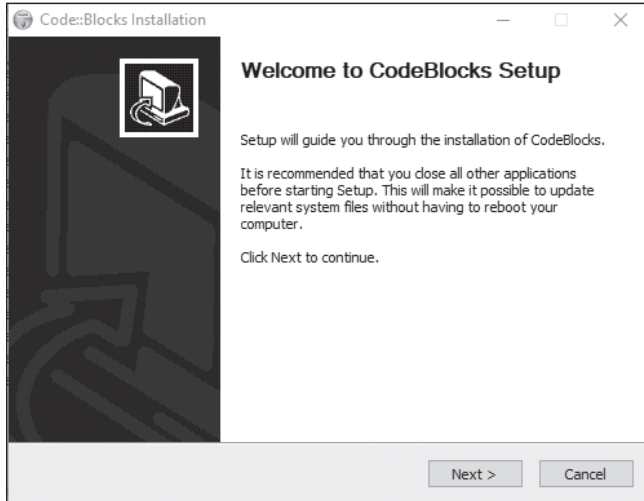


Abbildung 1.1: So geht es los mit Code::Blocks.

Lesen Sie die Lizenzbedingungen und klicken Sie dann auf I AGREE, wenn Sie mit den Bedingungen leben können. Wenn Sie auf CANCEL klicken, wird die Installation abgebrochen.

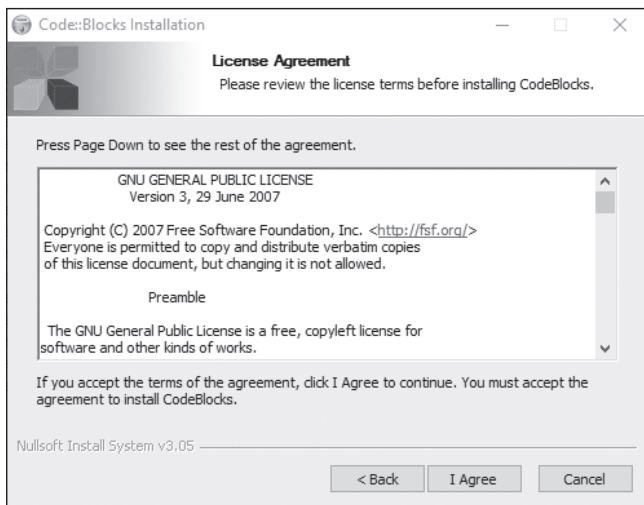


Abbildung 1.2: Ohne Lizenzen geht nichts.

Wenn Sie die Lizenzvereinbarung akzeptieren, wird ein Dialogfeld mit einigen Installationsoptionen geöffnet (siehe Abbildung 1.3). Bis auf die folgenden drei Ausnahmen sind die Standardeinstellungen harmlos:

- ✓ Die Option `DEFAULT INSTALL` muss aktiviert sein.
- ✓ Die Option `CONTRIB PLUGINS` muss aktiviert sein.
- ✓ Die Option `MINGW COMPILER SUITE` muss aktiviert sein.
- ✓ Die anderen Optionen sind derzeit nicht wichtig, Sie können sie einschalten.

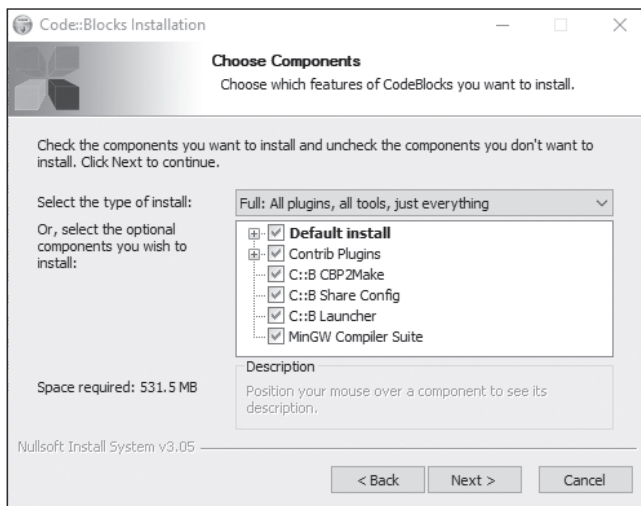


Abbildung 1.3: Die Installationsoptionen der IDE

Klicken Sie auf **NEXT**.

Das Installationsprogramm fragt Sie, wo Code::Blocks installiert werden soll (siehe Abbildung 1.4). Akzeptieren Sie das Standardverzeichnis `C:\PROGRAMME\CODEBLOCKS`.

Klicken Sie auf **INSTALL**.

Nun kopiert Code::Blocks zahlreiche Dateien in das Installationsverzeichnis und fragt am Ende, ob Sie das Programm sofort starten wollen: `DO YOU WANT TO RUN CODE::BLOCKS NOW?` Klicken Sie auf **JA**, um Code::Blocks gleich zu starten.

Wenn Sie das Bild aus Abbildung 1.5 sehen, beginnt in Kürze die Vorstellung Ihres C-Compilers.

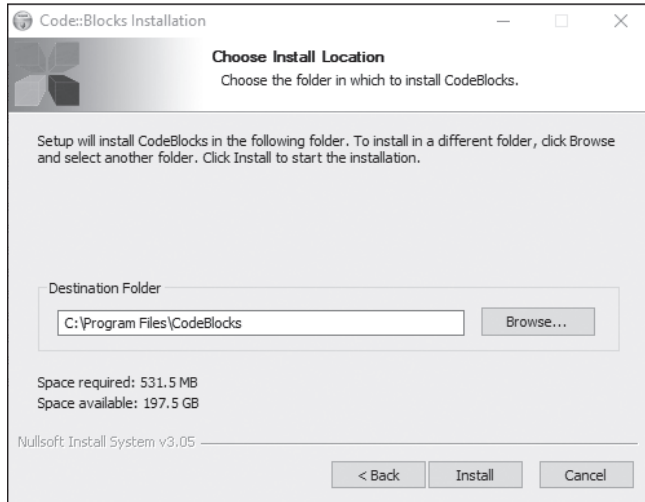


Abbildung 1.4: Wohin soll das Programm installiert werden?

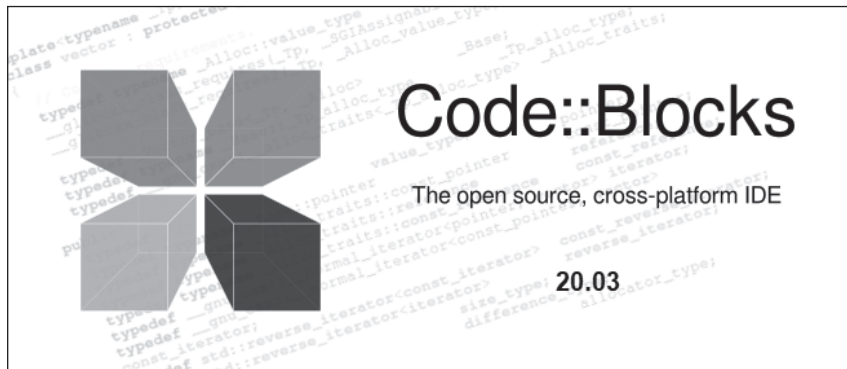


Abbildung 1.5: So sieht es aus, wenn Code::Blocks loslegt.

Die Compilereinstellungen

Im gestarteten Programm müssen Sie noch die Einstellungen für den Compiler anpassen. Zu Beginn wird der Compiler ausgewählt (siehe Abbildung 1.6) und die Dateiendungen werden verbunden.

Die IDE – also insbesondere der Editor zur Eingabe der C-Programme – und der Compiler sind unterschiedliche Programme (siehe Abbildung 1.7). Es wäre möglich, mit der IDE auch andere Compiler zu verbinden. Wir bleiben aber beim GNU GCC Compiler.

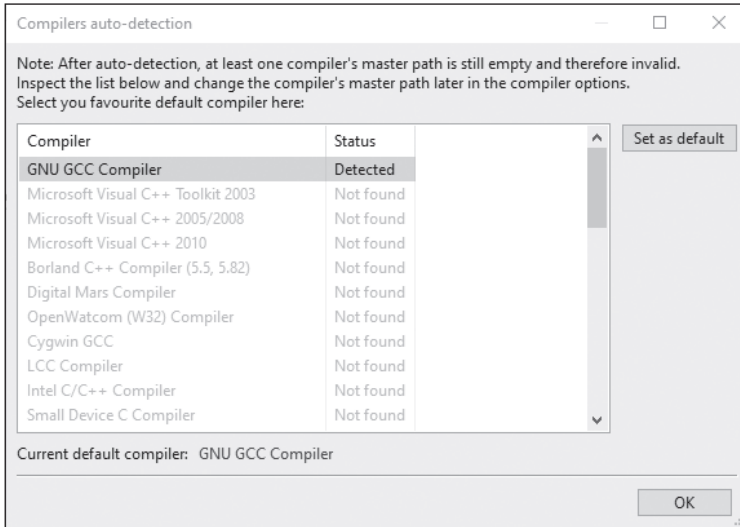


Abbildung 1.6: Der Compiler wird ausgewählt.

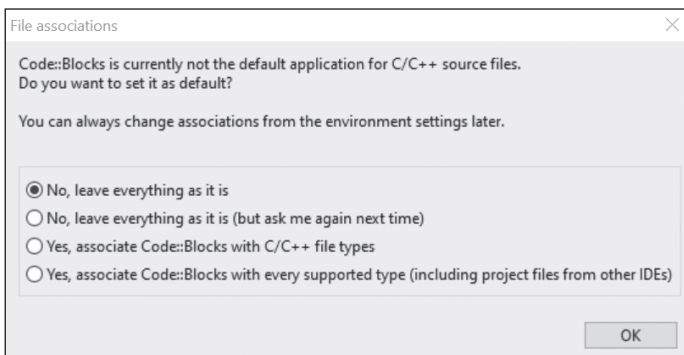


Abbildung 1.7: Code::Blocks für C-Programm

C-Programme werden als Textdateien geschrieben mit der Endung `.c` am Ende. Code::Blocks stellt sich so ein, dass `.c` immer mit ihm verbunden wird.

Innerhalb von Code::Blocks wählen Sie den Menüpunkt `SETTINGS | COMPILER AND DEBUGGER` und vergewissern sich, dass alle Flags so wie in Abbildung 1.8 und Abbildung 1.9 gesetzt sind.

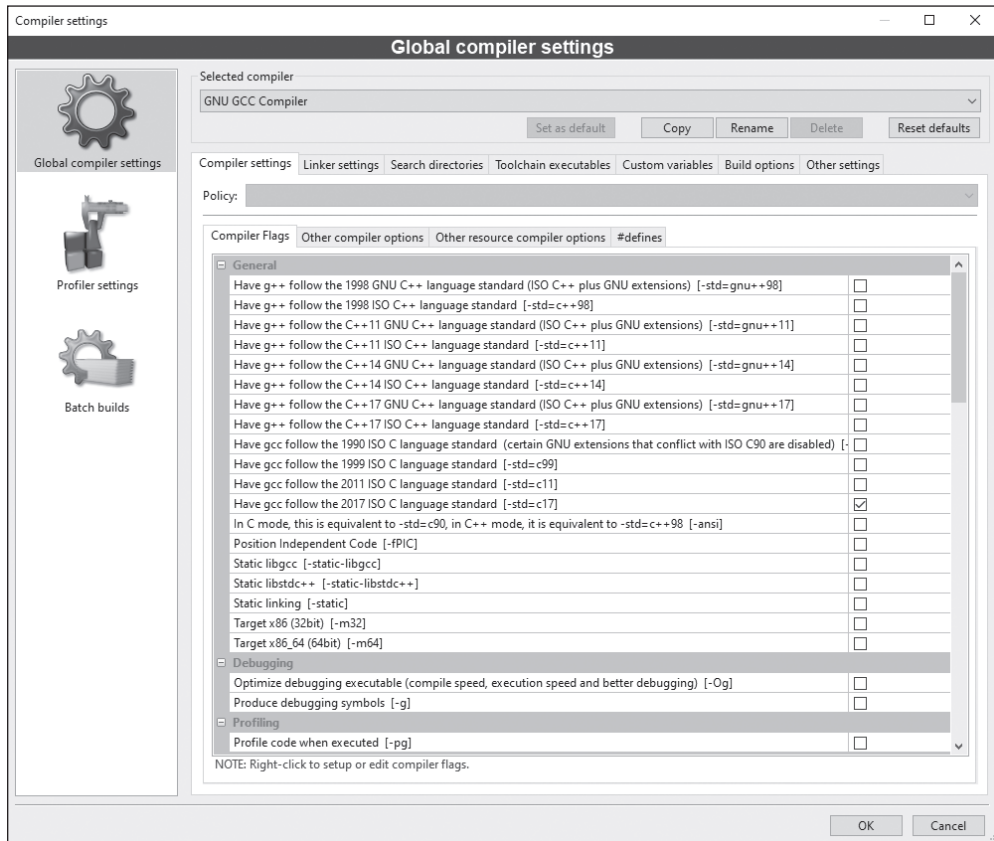


Abbildung 1.8: Der erste Teil der Compiler-Einstellungen



Passen Sie auf, dass der richtige Compiler-Standard ausgewählt ist – 2017 ISO C, wie in Abbildung 1.8.

Im gleichen Dialogfeld muss als Compiler außerdem unter **SELECTED COMPILER** der **GNU GCC COMPILER** ausgewählt sein.

- ✓ Klicken Sie auf OK.
- ✓ Es geht gleich los.
- ✓ 2017 ISO C? Das erkläre ich in Kürze.

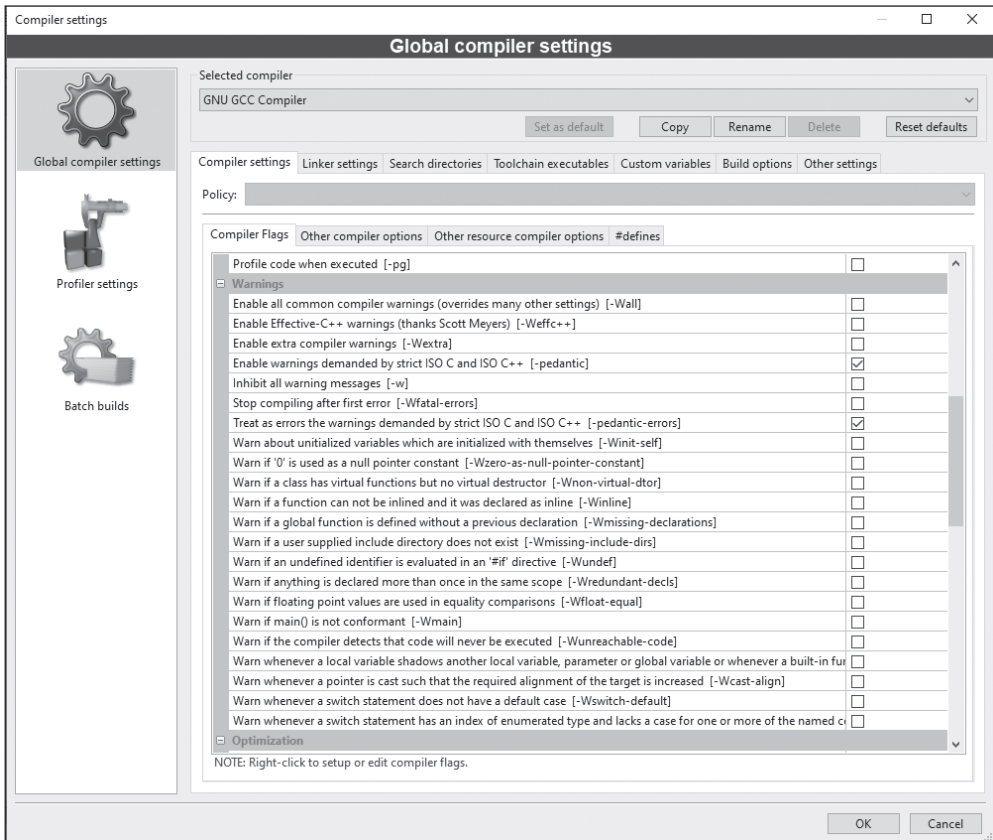


Abbildung 1.9: Der zweite Teil der Compiler-Einstellungen

Ihr erstes Programm

Der erste Schritt, um ein C-Programm zu erstellen, besteht darin, C-Anweisungen mit einem Texteditor einzugeben. Die Code::Blocks-Benutzerschnittstelle besteht hauptsächlich aus einem Programmeditor, der speziell für das Schreiben von C-Programmen entworfen wurde.

Welches C hätten Sie gern?

Wie Sie bei den Compilereinstellungen gesehen haben, gibt es mehrere Varianten oder Einstellungen zum C-Compiler. C hat einige Jahre auf dem Buckel, alle Jahre wird der Standard neu überarbeitet und herausgegeben. Mit der Compilereinstellung können Sie wählen, welchen Standard aus welchem Jahr Sie als Basis für Ihre Programme verwenden möchten.

Alle Beispiele im Buch funktionieren im Modus C17, aber Sie sollten alles auch im Modus C90 übersetzen können. Keines der Beispiele im Buch taucht so tief ab, dass Sie hier in Konflikte mit den Standards kommen.

Standard	Jahr	Anmerkungen
K&R-C	1978	Ursprünglich von den Erfindern Brian W. Kernighan und Dennis Ritchie verwendet, und noch kein ISO-Standard
C89	1989	Die erste offiziell normierte Sprachversion, auch erstmalig als ANSI-C bezeichnet
C90	1990	ANSI C, C89 und ISO C90 werden identisch benutzt. C90 ist die erste ISO-Standardisierung, jeder Compiler heute sollte mit C90 klarkommen.
C95	1995	Eine Erweiterung, vor allem in Bezug auf bessere Unterstützung von Zeichensätzen
C99	1999	Die CPUs wurden immer größer, und C lernte viele neue Datentypen. Mehr dazu in Kapitel 3.
C11	2011	Multi-Threading wurde immer wichtiger in der Entwicklung, C11 bot verbesserte Unterstützung.
C17	2017	Einige Korrekturen von C11

Tabelle 1.1: Wesentliche C-Standards im Laufe der Zeit

An die Tastatur!

Wählen Sie **FILE|NEW|PROJECT**. Suchen Sie das Icon **CONSOLE APPLICATION**, wählen Sie es aus und klicken Sie auf **GO**, wie in Abbildung 1.10 zu sehen.

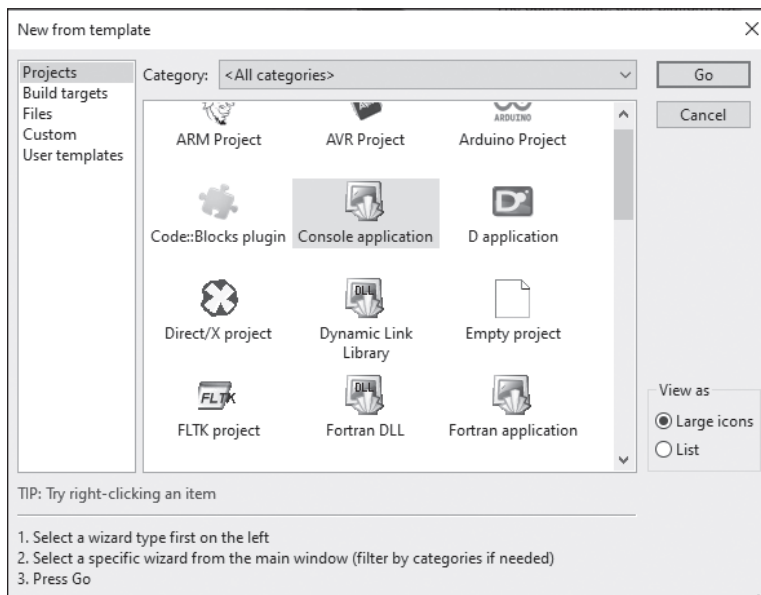


Abbildung 1.10: Eine Konsolenanwendung soll es sein.

Im nächsten Fenster mit der Assistentenseite klicken Sie auf **NEXT**.

Wählen Sie nun die Programmiersprache aus, indem Sie die Zeile **C** markieren und auf **NEXT** klicken. Code::Blocks unterstützt außer C auch C++.



Abbildung 1.11: C und nicht C++

Im Dialogfeld für den Projektnamen wählen Sie in der Zeile FOLDER TO CREATE PROJECT IN den Pfad aus, wo das Projekt liegen soll. In der Zeile PROJECT TITLE geben Sie den Namen des Projekts an, nennen wir es mal Goodbye.

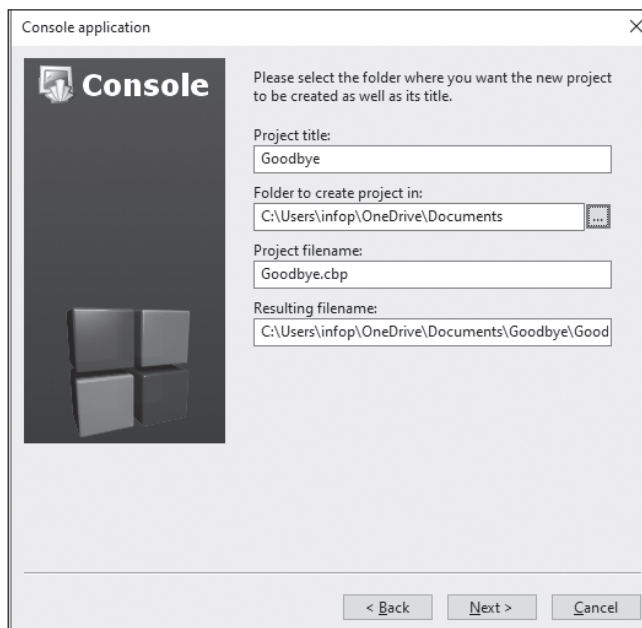


Abbildung 1.12: Unser erstes Programm Goodbye.c

Erneut klicken Sie auf NEXT.

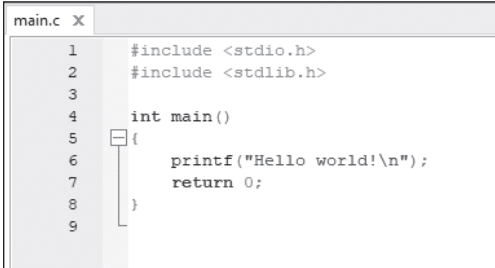
Im nächsten Dialogfeld werden einige Optionen angezeigt, wie das Projekt erstellt werden soll. Die Vorgaben sind bereits in Ordnung und können beibehalten werden.

Mit einem Klick auf FINISH wird das Projekt erstellt.

Den C-Code eingeben

Das Projekt Goodbye besteht nun aus einer einzigen `main.c`-Datei, die nichts tut. Im nächsten Schritt muss also das Programm eingegeben werden.

Klicken Sie auf `main.c`, das sich im MANAGEMENT-Fenster unter CONVERSION|SOURCES befindet. Code::Blocks öffnet ein Fenster mit einem elementaren Programm, wie in Abbildung 1.13 zu sehen ist.



```
main.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      printf("Hello world!\n");
7      return 0;
8  }
9
```

Abbildung 1.13: Das vorgegebene Programmgerüst

Geben Sie in `main.c` das folgende Programm ein, und zwar exakt so, wie es geschrieben ist:

```
#include <stdio.h>

int main()
{
    printf("Ade, grausame Welt!\n");
    return 0;
}
```


Listing 1.1: Name: Goodbye.c


Wählen Sie FILE|SAVE, um die Datei zu speichern.

Spickzettel für die Eingabe des Programms Goodbye.c


Beginnend mit der ersten Zeile geben Sie Folgendes ein:

```
#include <stdio.h>
```

Geben Sie das Gatter # (auch als Nummernzeichen bekannt), das Wort `include`, ein Leerzeichen, eine öffnende spitze Klammer `<`, das Wort `stdio`, einen Punkt, ein `h` und eine schließende spitze Klammer `>` ein. Verwenden Sie keine Großbuchstaben. Drücken Sie zweimal , um die Zeile zu beenden und eine leere Folgezeile zu erhalten.

Geben Sie das Wort `int`, ein Leerzeichen und dann gleich `main`, gefolgt von einer öffnenden runden Klammer `(` und einer schließenden runden Klammer `)` ein. Drücken Sie , um die Zeile zu beenden.


```
int main()
```


Die nächste Zeile enthält nur ein Zeichen: eine öffnende geschweifte Klammer `{`. Drücken Sie .

```
{
```



Drücken Sie die -Taste, um die nächste Zeile einzurücken, falls Ihr Editor dies nicht ohnehin automatisch macht:

```
    printf("Ade, grausame Welt!\n");
```

Geben Sie `printf` (als ein Wort), gefolgt von einer öffnenden runden Klammer `(` und einem Anführungszeichen `"` ein. Und dann tippen Sie `Ade, grausame Welt!`. Geben Sie einen Backslash `\` und ein kleines `n`, dann wieder ein Anführungszeichen `"` und eine schließende runde Klammer `)` ein. Danach geben Sie ein Semikolon `;` ein und drücken .

Jetzt fehlt etwas: Tippen Sie `return`, gefolgt von einer `0` und einem Semikolon `;` – kein Komma, sondern ein Semikolon. Danach drücken Sie wieder die Taste .

```
    return 0;
```

Beginnen Sie die nächste Zeile, indem Sie die Taste  drücken und damit den Cursor wieder an den Anfang der Zeile zurücksetzen. Geben Sie eine schließende geschweifte Klammer `}` ein und drücken Sie danach .

```
}
```

Ich weiß, dass dies vielleicht nicht besonders aufregend aussieht, aber Sie haben gerade den Code für Ihr erstes C-Programm eingegeben!

Ihr Programm erstellen

Nachdem Sie Ihre C-Quelldatei `main.c` auf die Festplatte gespeichert haben, können Sie daraus den ausführbaren Maschinencode erstellen.

Wählen Sie zu diesem Zweck den Menübefehl `BUILD|BUILD` oder drücken Sie `Strg+F9`. Sofort legt Code::Blocks los und übersetzt Ihr Programm. Falls alles geklappt hat, steht unten im Fenster folgende Meldung:

```
Process terminated with status 0 (0 minutes, 0 seconds)
0 errors, 0 warnings
```

- ✓ Wenn Ihre Programme größer werden, wird irgendwann auch eine größere Zahl als 0 Sekunden angezeigt werden. Aber so kurz – da wird schnell übersetzt.
- ✓ Sollte bei der Eingabe ein Fehler aufgetreten sein und das Programm nicht startbar sein – im nächsten Kapitel erfahren Sie, wie Sie Fehler beseitigen können.

Ihr Programm ausführen

Jetzt können wir Ihre neue Kreation ausführen ... das heißt Ihr Programm laufen lassen. Sie werden die Programmdatei ausführen, um zu sehen, wie gut es funktioniert.

Um das Programm auszuführen, wählen Sie `BUILD|RUN` oder drücken Sie `Strg+F10`.

Sofort öffnet sich ein Fenster und Sie sehen die wenig spektakuläre Bildschirmausgabe (Abbildung 1.14).

```
C:\Users\infop\OneDrive\Documents\Goodbye\bin\Debug\Goodbye.exe
Ade, grausame Welt!
Process returned 0 (0x0) execution time : 0.009 s
Press any key to continue.
```

Abbildung 1.14: Die Ausgabe des ersten Programms

Die Meldung `Press any key to continue.` gibt Ihnen Gelegenheit, die Ausgaben in Ruhe zu studieren, bevor das Programm beendet wird. Wenn Sie `↵` drücken, wird das Fenster geschlossen und sein Inhalt verschwindet. Glückwunsch! Sie haben gerade Ihr erstes C-Programm eingegeben, erstellt und ausgeführt.



Beachten Sie, dass Code::Blocks nicht wirklich dafür bestimmt ist, Windows-Programme zu entwickeln. Theoretisch können Sie eine Windows-Anwendung mit Code::Blocks schreiben, aber es ist nicht einfach. (Mit Microsofts Visual Studio geht das sehr viel leichter.)

Die C-Programme, die Sie in diesem Buch schreiben, haben eine sogenannte Kommandozeilen- oder Befehlszeilenschnittstelle (englisch *command line interface*), die in einem Konsolen-Fenster ausgeführt wird.

Ein Wort zu Code::Blocks und Spenden

Obwohl Code::Blocks freie Software unter der GNU-Lizenz ist, sollten Sie einen kleinen Beitrag bezahlen, damit die Entwicklung des Werkzeugs weiterhin finanziert werden kann. Sie müssen nicht zahlen, aber Sie können es tun, falls Sie wollen. Und Sie sollten es tun, wenn Sie können. Die Webseite enthält dazu mehr Details.

Schauen Sie dazu auf der Website unter *DONATIONS*.