Kurzeinführung zu Code::Blocks

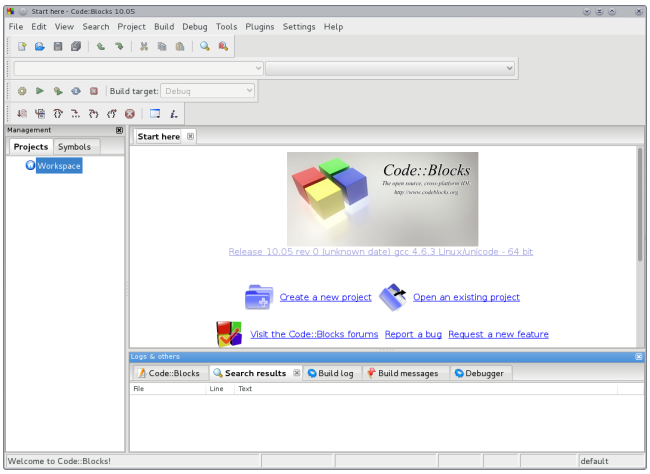
1 Kurzeinführung in Code::Blocks

Im Folgenden erklären wir die grundlegende Bedienung der Entwicklungsumgebung `Code::Blocks`, die für Linux, Windows und Mac frei verfügbar ist. Sie beinhaltet u.a. einen Editor zum Schreiben des Programmcodes, eine Einbindung eines C++-Compilers, um den Code per Knopfdruck übersetzen und linken zu können, Unterstützung beim Beheben der durch den Compiler gefundenen Syntaxfehler sowie eine grafische Oberfläche zur Bedienung eines Debuggers, um auch semantische Fehler aufspüren zu können.

Dazu betrachten wir eine Beispielsitzung mit `Code::Blocks`, bei der wir ein neues Projekt erstellen, übersetzen und ausführen.

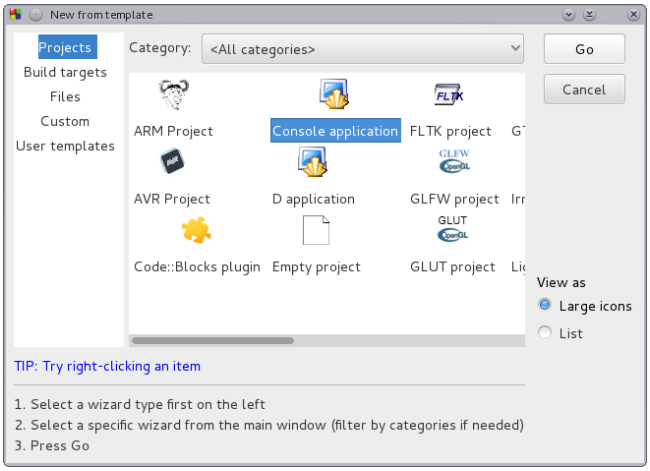
1. Starten Sie `Code::Blocks`, z.B. durch Klick auf das entsprechende Icon oder durch Eingabe von `codeblocks` in der Linux-Shell bzw. -Konsole.

2. Das Hauptfenster von `Code::Blocks` öffnet sich. Evtl. den Tipp des Tages wegklicken, dann kann es schon losgehen.



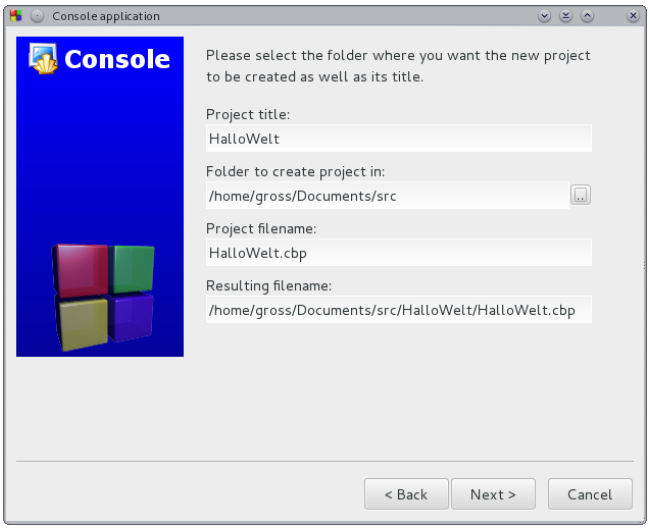
Um ein neues Programm zu schreiben müssen wir zunächst ein neues *Projekt* anlegen. Klicken Sie dazu auf „Create new project“ oder benutzen sie den Menüpunkt „File > New... > Project“.

3. Wählen Sie als Projekttyp „Console application“, da wir unser Programm in der Konsole starten und bedienen möchten und z.B. keine grafische Benutzeroberfläche programmieren wollen.



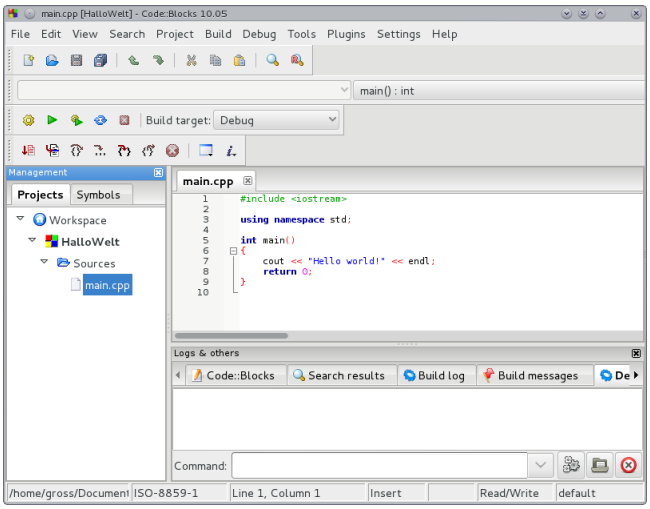
Der Button „Go“ startet einen Assistenten, der die notwendigen Informationen in mehreren aufeinanderfolgenden Dialogfenstern abfragt.

4. Wir wählen also C++als Sprache, danach legen wir z.B. „HalloWelt“ als Projekttitel und als Verzeichnis „Documents/src/“fest.

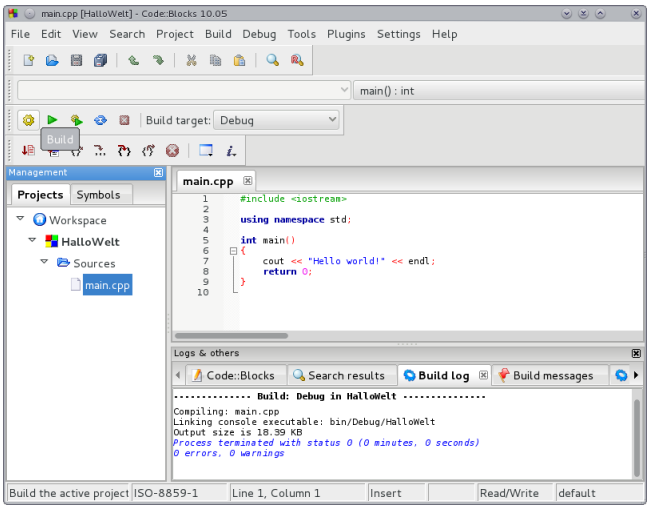


Die folgenden Angaben zum Compiler lassen wir unverändert, da wir den GCC-Compiler benutzen wollen. Ein Klick auf „Finish“ erzeugt dann das Projekt.

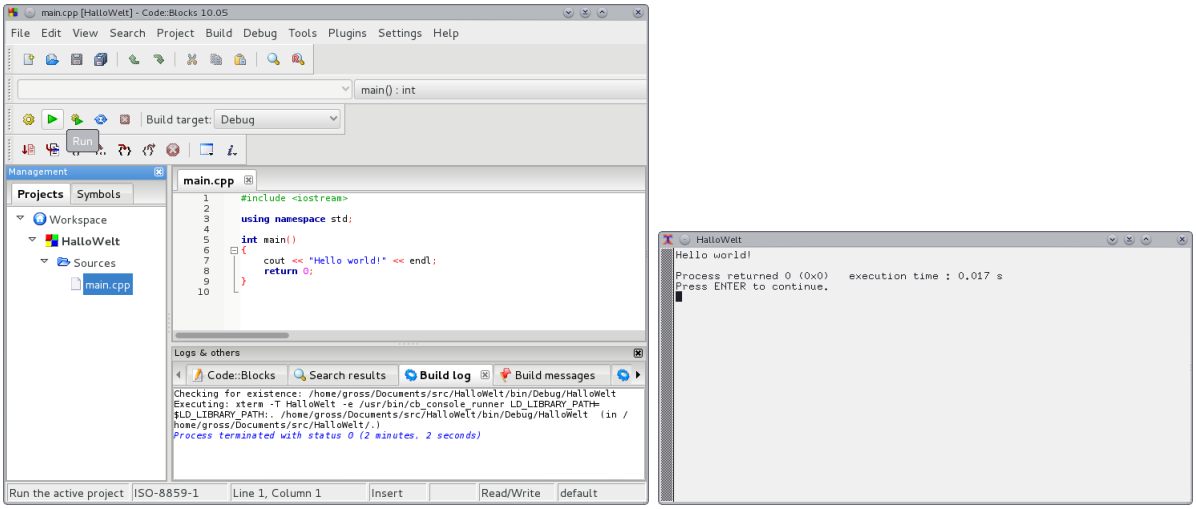
5. Im Projekt-Manager (links) sehen wir das neu erzeugte Projekt `HalloWelt`, wo wir auch die automatisch angelegte Hauptdatei `main.cpp` finden, die wir mit einem Doppelklick im Editor öffnen. Diese enthält bereits ein kleines Programm.



6. In der Werkzeugleiste (oben) finden wir einen Button mit einem Zahnrad, der das Übersetzen („Build“) des Projektes ausführt. Alternativ können Sie auch das Menü („Build > Build“) oder die Tastenkombination Strg+F9 benutzen. Dass das Programm fehlerfrei übersetzt wurde, erkennen Sie an der Meldung „0 errors“ im Build Log (unten).

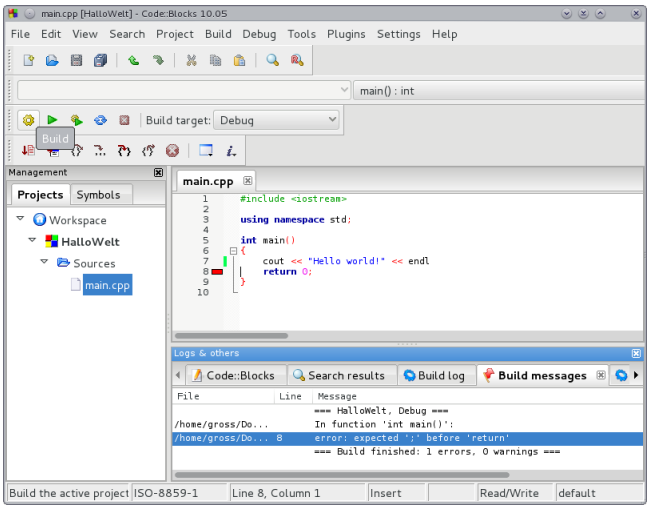


7. In der Werkzeugleiste (oben) finden wir einen Button mit einem grünen Play-Dreieck (Play-Symbol), der das Programm in einer Shell ausführt („Run“). Alternativ können Sie auch das Menü („Build > Run“) oder die Tastenkombination Strg+F10 benutzen. Wie erwartet, grüßt uns das Programm freundlich in der Konsole und ist dann auch schon wieder beendet.



Mit der Enter-Taste schließen wir das Konsolenfenster wieder.

8. Wir erzeugen nun extra einen Syntaxfehler, indem wir das Semikolon hinter `endl` entfernen. Wir speichern die Datei ab (Button mit Disketten-Symbol, Menü „File > Save File“ oder Tastenkombination Strg+S). Dann übersetzen wir erneut wie oben beschrieben (z.B. mit Strg+F9).



Der Compiler meldet in den Build Messages (unten) einen Fehler in Zeile 8: „expected ';' before 'return'". Diese Zeile ist im Editor auch mit einem roten Balken gekennzeichnet. Aufgrund der Fehlermeldung erkennen wir sofort, dass wir das Semikolon am Ende der vorhergehenden Zeile einfügen müssen.

9. Es kann durchaus vorkommen, dass der Compiler sehr viele Fehlermeldungen ausspuckt, z.B. bei einer vergessenen schließenden Klammer. Davon sollte man sich nicht irritieren lassen, denn die meisten Meldungen sind nur Folgefehler, die verschwinden, sobald Sie den ersten monierten Fehler behoben haben. Sie sollten in so einem Falle also immer den zuerst gefundenen Fehler beseitigen und dann erneut übersetzen.

10. Sie können nun `Code::Blocks` schließen (Menü „File > Quit" oder die Tastenkombination Strg+Q). Beim nächsten Start von `Code::Blocks` finden Sie Ihr Projekt im Projekt-Manager (links) wieder und können es dort weiter bearbeiten.