Ihr System einrichten

1

In diesem Kapitel

- Beschaffen Sie sich C++
- Erhalten Sie eine Kopie von Code::Blocks
- Richten Sie auf Ihrem System eine Arbeitsumgebung für Code::Blocks ein
- Erkennen Sie, wie Code::Blocks Ihnen beim Erledigen von Aufgaben hilft
- Arbeiten Sie mit anderen IDEs

Vielleicht haben Sie Ihre flammneue Kopie von C++ bereits auf Ihrer Maschine installiert. Sie können C++ alleinstehend oder als Teil einer IDE erhalten. Wie Sie mit C++ arbeiten, hängt davon ab, was Sie benötigen, aber wenn Sie bereits eine Kopie besitzen und damit glücklich sind, können Sie eigentlich auf dieses Kapitel verzichten – holen Sie sich eine neue Tasse Kaffee und machen Sie in der beruhigenden Erkenntnis mit dem nächsten Kapitel weiter, dass Sie bereits ein ganzes Kapitel dieses Buches ohne jegliche Anstrengung hinter sich gebracht haben. Für alle anderen Leser gilt, dass sie ohne ein installiertes C++ nicht viel anfangen können. Dies bedeutet natürlich, dass Sie online gehen, sich eine passende Softwareversion suchen und diese herunterladen müssen. Wenn Sie eine integrierte Entwicklungsumgebung (Integrated Development Environment – IDE) wie Code::Blocks verwenden, auf die wir hier in diesem Buch setzen, erhalten Sie zusammen mit Ihrer Installation eine Kopie von C++. Damit müssen Sie sich keine Gedanken mehr über den ersten Teil dieses Kapitels machen. Dieses Buch geht davon aus, dass Sie C++ 14 besitzen, die Version der Software, die zum Zeitpunkt des Schreibens dieses Buches zur Verfügung stand. Aber viele Beispiele funktionieren auch mit älteren Versionen von C++, falls Sie so etwas installiert haben.

Auch wenn sich dieses Buch auf das Arbeiten mit C++ auf den Plattformen Mac, Windows und Linux konzentriert, können Sie die hier vorgestellten Techniken auf vielen anderen Plattformen wie zum Beispiel einigen Smartphones einsetzen. Sie sollten aber beachten, dass je esoterischer Ihre Plattform wird, desto weniger Beispiele darauf funktionieren werden, weil Ihre Plattform mit ziemlicher Sicherheit besondere Programmiertechniken benötigt. Am besten klappt alles mit einer Kopie von Code::Blocks 13.12, die C++ 14 enthält. Beides wird dann in einem Rutsch auf dem Mac oder unter Windows oder Linux installiert.

An C++ 14 gelangen

Ein Produkt mit dem Namen C++ 14 gibt es nicht. Der C++-Standard 14 besagt einfach nur, was die Sprache enthält und wie sie implementiert werden sollte. Oder mit anderen Worten, Sie können nicht einfach online gehen und sich eine Kopie von C++ 14 besorgen; stattdessen müssen Sie die Implementierung eines Compilers auftreiben, die den Standard von C++ 14 enthält. Sie können sich zum Beispiel unter `http://gcc.gnu.org/releases.html` die C++-Version der Gnu Compiler Collection (GCC) herunterladen.

Jeder Anbieter interpretiert den Standard etwas anders und erweitert ihn ab und an sogar. Kurz, jeder Compiler kennt eine eigene Version von C++. Natürlich sind Sie nicht gezwungen, die besonderen Funktionen dieser Versionen zu verwenden, was wiederum bedeutet, dass Ihr Quellcode weniger anfällig für Probleme ist, zu denen es kommen kann, wenn Sie mehrere Compiler einsetzen. Die Beispiele in diesem Buch halten sich genau an den Standard C++ 14, weshalb Sie sie überall verwenden können.

Es ist wichtig, dass Sie verstehen, dass ein Compiler nicht dasselbe wie eine integrierte Entwicklungsumgebung (IDE) ist. Der Compiler ist häufig unabhängig von der IDE und wird oft auch von zwei unabhängigen Seiten gepflegt. So unterstützt zum Beispiel die IDE Code::Blocks mehrere Compiler, und der Compiler GCC arbeitet mit mehreren IDEs zusammen. Der Compiler ist ein sehr wichtiges Stück Software, weil es aus Ihrem Quellcode eine ausführbare Datei macht, die Ihr Betriebssystem ablaufen lassen kann.

Der Compiler, den Sie gewählt haben, muss die Plattformen unterstützen, mit denen Sie arbeiten wollen. So unterstützt zum Beispiel GCC nicht nur die Entwicklung auf Mac, Windows und Linux, sondern auch die auf einigen Acorn-RISC-Machine-Prozessoren (ARM-Prozessoren). Eventuell werden, wenn Sie dieses Kapitel lesen, noch weitere Plattformen unterstützt. Da der Compiler GCC fast universell einsetzbar ist, verwenden auch wir ihn in diesem Buch, wobei aber gilt, dass die Beispiele in der Regel auch mit anderen Compilern laufen.

Sich Code::Blocks besorgen

Die IDE Code::Blocks sorgt für eine Umgebung, in der Sie Quellcode schreiben, kompilieren, testen und bei Bedarf debuggen können. Die IDE selbst kompiliert den Quellcode nicht, aber sie unterstützt einen Compiler, der diesen Job dann übernimmt. (Dies geschieht dadurch, dass der Compiler Teil der IDE wird.) Sie können in Code::Blocks aus einer Reihe von Compilern auswählen, aber dieses Buch konzentriert sich auf den Einsatz von GCC, um sicherzustellen, dass die Beispiele auf so vielen Plattformen wie möglich ablaufen können. Wenn Sie mit Windows arbeiten, erhalten Sie GCC zusammen mit Ihrer Kopie von Code::Blocks – Sie müssen also nichts Besonderes unternehmen, sondern GCC nur bei der Installation auswählen. (Wenn Sie mit einem Mac- oder einem Linux-System arbeiten, müssen Sie GCC eigenständig installieren, da es in diesen Fällen den Compiler nicht zusammen mit Code::Blocks gibt.)

Wir haben beim Schreiben dieses Buches Code::Blocks Version 13.12 verwendet. Das heißt nicht, dass Sie es nicht auch mit früheren oder späteren Versionen von Code::Blocks verwenden können. Wenn Sie mit anderen Versionen von Code::Blocks arbeiten, kann es natürlich passieren, dass Sie den Code ein wenig ändern müssen. Diese Änderungen sind notwendig, um den Compiler zu unterstützen, der dann zusammen mit Ihrer Version von Code::Blocks ausgeliefert wird.

Code::Blocks gibt es sowohl in binärer Form als auch als Quellcode. Sie können beides unter `http://www.codeblocks.org/downloads` herunterladen. Wenn Sie mit Mac, Windows oder Linux arbeiten, sollten Sie sich von der Webseite `http://www.codeblocks.org/`

downloads/26 die binäre Version herunterladen. Dies ist die Version, die in diesem Buch verwendet wird. Der Abschnitt *Code::Blocks installieren* in diesem Kapitel erzählt Ihnen mehr darüber, wie Code::Blocks auf Ihrem System installiert wird.

Wenn Sie mit einer Windows-Installation arbeiten, müssen Sie dafür sorgen, dass Sie den »Imstaller« `codeblocks-13.12mingw-setup.exe` verwenden, um zusammen mit Code::Blocks auch eine Kopie von GCC zu erhalten. Wenn Sie auf Ihrer Maschine nicht über administrative Rechte verfügen, laden Sie stattdessen die Datei `codeblocks-13.12mingw-setup_user.exe` herunter; achten Sie unbedingt darauf, sie auf Ihrem System nicht im Ordner `Programme` beziehungsweise `Programme (x86)` zu installieren, weil diese Anwendung dort nicht funktioniert. Legen Sie einen Ordner an, auf den Sie Schreibrechte haben, und installieren Sie Code::Blocks dort.

Wir können es nicht oft genug erwähnen: Achten Sie darauf, von der Code::Blocks-Site die Datei `codeblocks-13.12mingw-setup.exe` herunterzuladen. Das Herunterladen und Installieren der Datei `codeblocks-13.12mingw-setup-TDM-GCC-481.exe` erzeugt eine Installation, die instabil und fehlerbehaftet ist. Und mit ziemlicher Sicherheit werden einige der Beispiele dieses Buches nicht mit dieser Version von Code::Blocks funktionieren. Die Herausgeber dieser Version wissen von diesen Stabilitätsproblemen und arbeiten daran. Schauen Sie im Blog von John Mueller (`http://blog.johnmuellerbooks.com`) nach, ob es inzwischen Aktualisierungen von Code::Blocks gibt.

Code::Blocks installieren

Bevor Sie Code::Blocks als IDE verwenden können, müssen Sie es installieren. Die nächsten Abschnitte beschreiben, wie Code::Blocks auf den Plattformen installiert wird, die dieses Buch unterstützt. Die Installationen in diesen Abschnitten gehen davon aus, dass Sie die binäre Version von Code::Blocks heruntergeladen haben und dass Sie keine benutzerdefiniert kompilierte Version des Produkts verwenden.

Mit Windows arbeiten

Code::Blocks kommt mit einem Windows-Installer, der die Aufgabe, die IDE zu installieren, stark vereinfacht. Die folgenden Schritte helfen Ihnen, mit der Installationsdatei `codeblocks-13.12mingw-setup.exe` oder `codeblocks-13.12mingw-setup_user.exe` klarzukommen:

1. **Führen Sie auf der Datei, die Sie von der Code::Blocks-Site heruntergeladen haben, einen Doppelklick aus.**

   Es startet der Code::Blocks Setup Wizard.

2. **Klicken Sie auf Next.**

   Es erscheint die (englischsprachige) Lizenzvereinbarung, die Sie gegebenenfalls durchlesen sollten, damit Sie die Nutzungsbedingungen von Code::Blocks kennen.

3. Klicken Sie auf I Agree.

Damit stimmen Sie der Lizenzvereinbarung zu, und der Installationsassistent zeigt eine Reihe von Konfigurationsmöglichkeiten an (siehe Abbildung 1.1). Wir gehen in diesem Buch davon aus, dass Sie die standardmäßige vollständige Installation vornehmen.

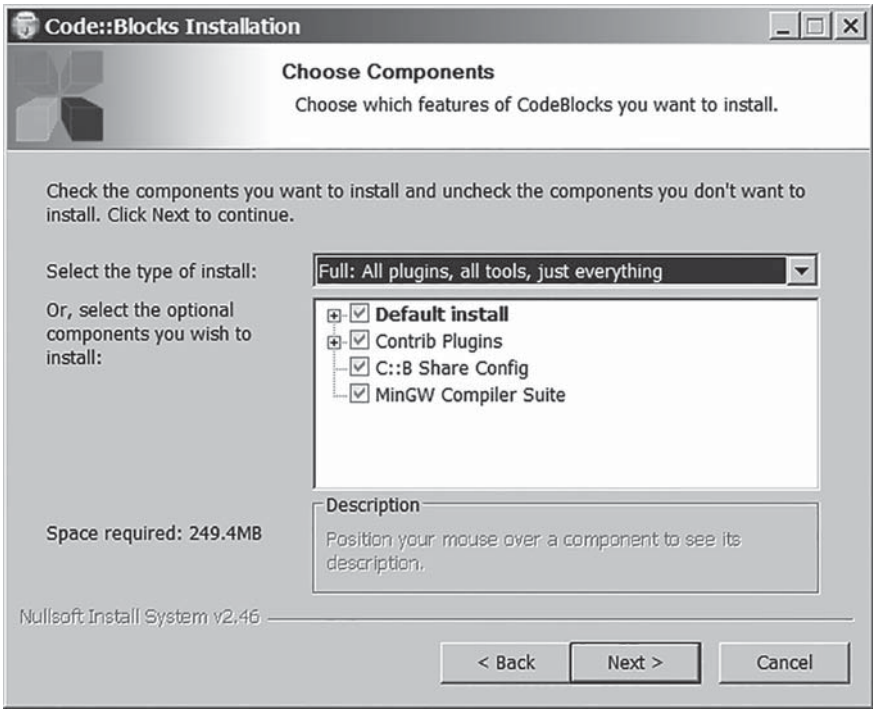

Abbildung 1.1: Der Assistent fordert Sie auf, die gewünschten Konfigurationsoptionen auszuwählen.

4. Klicken Sie auf Next.

Das Installationsprogramm fragt nach, wo Code::Blocks auf Ihrem Computer installiert werden soll. Code::Blocks schreibt, anders als andere Programme, von Zeit zu Zeit Daten in diesen Ordner. Wenn Sie keinen administrativen Zugriff auf den Ordner `Programme` beziehungsweise auf einem 64-Bit-System auf `Programme (x86)` haben, sollten Sie am besten einen Ordner verwenden, auf den Sie Schreibrechte haben. Wir verwenden hier zum Beispiel den Ordner `C:\CodeBlocks`.

5. Geben Sie in das Feld für den Zielordner (Destination Folder) `C:\CodeBlocks` ein und klicken Sie auf Install.

Das Installationsprogramm erstellt automatisch den Ordner `C:\CodeBlocks`, wenn dieser noch nicht existiert. Sie können zusehen, wie die Dateien auf Ihrem System im Zielordner installiert werden.

Der Installationsassistent zeigt nun ein Dialogfeld an, das Sie fragt, ob Code::Blocks ausgeführt werden soll. Klicken Sie in diesem Fall auf Nein.

6. **Klicken Sie auf NEXT.**

   Sie sehen ein Dialogfeld, das die Fertigstellung der Installation anzeigt.

7. **Klicken Sie auf FINISH.**

   Der Installationsassistent beendet sich.

Mit Mac OS/X arbeiten

Die Installation von Code::Blocks auf einem Mac verlangt ein paar kleinere Dinge mehr als die Windows-Installation. Code::Blocks benötigt für die Installation Mac OS/X 10.4 oder später. Die folgenden Schritte sagen Ihnen, wie Sie auf Ihrem Mac-System an eine funktionierende Code::Blocks-Installation gelangen:

1. **Laden Sie aus dem App Store Xcode herunter und installieren Sie es, um an eine Kopie von GCC zu gelangen.**

   Sie können überprüfen, ob Sie den GNU GCC Compiler installiert haben, indem Sie ein Terminal öffnen, `gcc –v` eingeben und ↵ drücken. Wenn Sie GCC installiert haben, sollten Sie Versionsinformationen und Compileranweisungen sehen.

   Wenn Sie mit einer älteren Version von Mac OS/X arbeiten, wird beim Herunterladen von Xcode eine Gebühr fällig. (Ab Version 10.6 ist der Zugriff kostenfrei.) Da die Datei, die Sie herunterladen, sehr groß ist (ungefähr 1 GB), sollten Sie die entsprechende Zeit dafür einplanen.

2. **Entpacken Sie die Code::Blocks-Dateien in einen Ordner.**

   Sie sehen eine Reihe von Dateien, zu denen auch die Code::Blocks-Anwendung, eine Readme-Datei mit den neuesten Update-Informationen und eine PDF-Datei gehören, die die Dokumentation enthält.

3. **Öffnen Sie den Ordner `Programme`.**

   Sie sehen, welche Anwendungen auf Ihrem System installiert sind.

4. **Ziehen Sie die Datei `CodeBlocks.app` aus dem Ordner, den Sie für das Entpacken verwendet haben, in den Ordner `Programme`.**

   Das Betriebssystem fügt Code::Blocks der Liste der verwendbaren Anwendungen hinzu.

5. **Gehen Sie zu `https://developer.apple.com/downloads`.**

   Die Site verlangt, dass Sie sich anmelden, um eine kostenlose Apple-ID zu erhalten. Folgen Sie einfach den Aufforderungen auf dem Bildschirm, um eine Apple-ID zu bekommen. Dieser Anmeldevorgang ist kostenfrei.

6. **Klicken Sie für die Xcode-Verknüpfung auf die Command Line Tools (die manchmal auch Xcode-Befehlszeilen-Tools genannt werden).**

   Das Betriebssystem lädt die Datei herunter und zeigt einen Paketordner an.

7. **Führen Sie auf dem Command-Line-Tools-Paket einen Doppelklick aus.**

   Das Betriebssystem installiert das Paket, über das aus Code::Blocks heraus der Zugriff auf GCC ermöglicht wird.

Eine Standardinstallation unter Linux

Leider gibt es für die Installation von Code::Blocks unter Linux keine einheitlichen Vorgehensweise, weil jede Linux-Variante ihre eigene Vorgehensweise verlangt. Code::Blocks unterstützt

- ✔ Blag
- ✔ Debian
- ✔ Fedora
- ✔ Gentoo
- ✔ Playpux
- ✔ Red-Hat-Distributionen, die auf dem Package Manager (RPM) basieren (wie SUSE, Red Hat, Yellow Dog, Fedora Core und CentOS)
- ✔ Ubuntu

Jede dieser Distributionen hat ihre eigene Vorgehensweise, die Sie unter `http://wiki.codeblocks.org/index.php?title=Installing_Code::Blocks` nachlesen können. Achten Sie darauf, den Compiler, den Debugger und gegebenenfalls die IDE herunterzuladen und zu installieren, indem Sie sorgfältig den Anleitungen folgen (die im Terminal angezeigt werden). Die Datei, die Sie von `http://www.codeblocks.org/downloads/26` heruntergeladen haben, enthält alle Pakete für die Installation von Code::Blocks, weshalb Sie sich nicht um jedes einzelne Paket separat kümmern müssen.

Einige Linux-Installationen kennen für das Arbeiten mit Code::Blocks bestimmte Anforderungen oder Einschränkungen. Davon gibt es eine, die offensichtlich dieses Buch betrifft: Boost wird weder für Red Hat noch für CentOS unterstützt. Aus diesem Grund können Sie bei diesen Systemen die Beispiele in Kapitel 4 und 5 nicht nutzen. Außerdem verwendet Code::Blocks beim Starten (Run) der Programme standardmäßig XTERM. Das Programm ist meist installiert. Sollte es nicht vorhanden sein, muss es gegebenenfalls nachinstalliert werden. Sollten Sie auf weitere Einschränkungen stoßen, lassen Sie es John Mueller unter `John@JohnMuellerBooks.com` wissen; er wird sie im Blog dieses Buches posten.

Die grafische Linux-Installation verwenden

Nun gibt es aber auch Linux-Distributionen, die auf Debian-Versionen basieren (wie Ubuntu 12.x und neuer), die eine grafische Installationsroutine bereitstellen. Sie benötigen das Kennwort (`sudo`) der administrativen Gruppe, um diese Möglichkeit, die Ihnen viel Zeit spart, nutzen zu können. Die folgenden Schritte skizzieren die grafische Installation für Ubuntu, wobei diese Technik der bei anderen Linux-Installationen ähnelt.

1. **Starten Sie unter UBUNTU das Programm SOFTWARE-CENTER (auf anderen Plattformen können Sie SYNAPTICS oder MUON verwenden).**

   Sie sehen eine Liste mit den beliebtesten Programmen, die heruntergeladen und installiert werden können (siehe Abbildung 1.2). Die Liste auf Ihrem Bildschirm unterscheidet sich mit ziemlicher Sicherheit von der, die die Abbildung zeigt.

Abbildung 1.2: Das Ubuntu Software-Center enthält die beliebtesten Programme.

Abbildung 1.3: Die Kategorie ENTWICKLUNGSWERKZEUGE enthält einen Code::Blocks-Eintrag.

2. Wählen Sie in der Dropdownliste ALLE ANWENDUNGEN die Option ENTWICKLUNGSWERKZEUGE.

Sie sehen eine Liste mit Entwicklerwerkzeugen, zu denen auch Code::Blocks gehört, wie Abbildung 1.3 zeigt.

3. **Führen Sie auf Code::Blocks einen Doppelklick aus.**

   Das Ubuntu Software-Center liefert Einzelheiten zu Code::Blocks und bietet an, die Anwendung für Sie zu installieren (siehe Abbildung 1.4).

Abbildung 1.4: Gegebenenfalls können Sie weitere Informationen zu Code::Blocks erhalten.

4. **Klicken Sie auf Installieren.**

   Ubuntu beginnt mit der Installation von Code::Blocks. Ein Fortschrittsbalken zeigt den Status des Herunterladens und der Installation an. Wenn die Installation fertig ist, wird aus der Schaltfläche Installieren die Schaltfläche Entfernen.

5. **Schließen Sie den Ordner Ubuntu Software-Center.**

   Sie sehen, dass dem Desktop ein Code::Blocks-Symbol hinzugefügt worden ist. Die IDE wartet auf ihren Einsatz.

Die wesentlichen Programmfunktionen von Code::Blocks kennenlernen

Unabhängig davon, auf welcher Plattform Sie Code::Blocks installiert haben, erhalten Sie immer eine IDE mit Standardmerkmalen. Dies ist einer der besten Gründe dafür, eine IDE wie Code::Blocks zu nutzen – Sie können dieselbe IDE auf jeder beliebigen Plattform einsetzen.

Ihr Bildschirm kann anders aussehen als das, was in diesem Buch gezeigt wird. Auch wenn wir hier Screenshots der Windows-Version von Code::Blocks verwenden, stellt Code::Blocks auch in anderen Installationsumgebungen dieselben Programmfunktionen zur Verfügung, wobei die IDE dort vielleicht etwas anders aussieht. Die folgenden Abschnitte beschreiben die wesentlichen Funktionen, die Sie kennen müssen, wenn Sie mit Code::Blocks arbeiten wollen.

Code::Blocks zum ersten Mal starten

Öffnen Sie das Programm Code::Blocks. Verwenden Sie dafür die Technik, die Sie normalerweise in Ihrer Betriebssystemumgebung einsetzen. Führen Sie zum Beispiel unter Windows oder auf dem Mac einen Doppelklick auf dem Code::Blocks-Symbol aus. Wenn Sie Code::Blocks zum ersten Mal starten, sehen Sie das Dialogfeld Compiler Auto-detection für das automatische Erkennen des Compilers. Wählen Sie den Eintrag GNU GCC Compiler aus (bei dem es sich wohl um den einzigen Eintrag handeln wird) und klicken Sie auf OK.

Nun zeigt Code::Blocks das Dialogfeld File Associations an (siehe Abbildung 1.5), in dem Sie Code::Blocks Dateien zuordnen können. Dies erleichtert Ihre Arbeit, weil das Öffnen einer auf diese Weise zugeordneten Datei gleichzeitig auch die IDE öffnet.

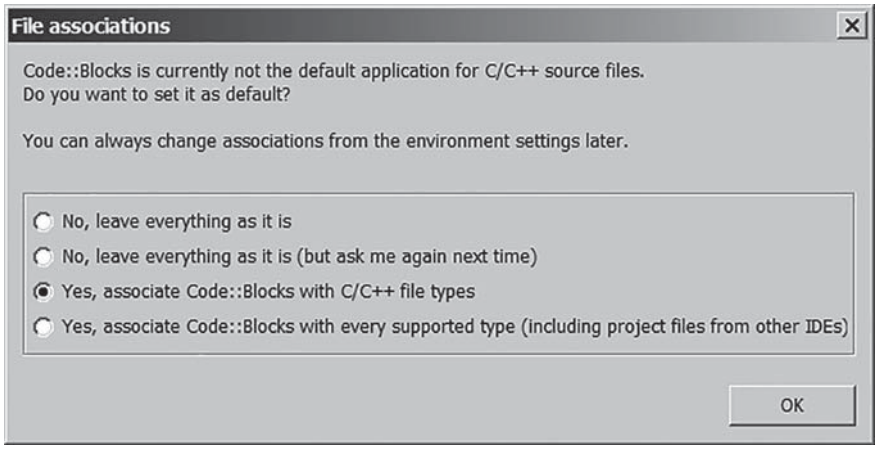

Abbildung 1.5: Verbinden Sie Code::Blocks mit Ihren C++-Dateien, um deren Verwaltung zu erleichtern.

Wählen Sie eine der Yes-Optionen dieser Liste aus. Es ist möglich, Code::Blocks auch mit anderen Quellcodetypen zu verknüpfen, aber in diesem Buch benötigen Sie nur eine Verknüpfung mit C++-Dateien. Klicken Sie auf OK, um die Aktion abzuschließen. Sie sehen nun die IDE.

Damit Ihre Änderungen auf Dauer gelten, wählen Sie File|Quit, um das Programm zu verlassen. Die IDE zeigt eine Meldung wie die in Abbildung 1.6. Diese Meldung besagt, dass Sie Änderungen an der Konfiguration von Code::Blocks vorgenommen haben. Wenn Sie auf Ja beziehungsweise Yes klicken, werden Ihre Änderungen gespeichert.

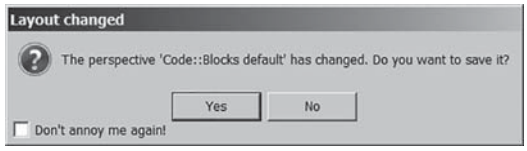

Abbildung 1.6: Speichern Sie Ihre Änderungen auf der Festplatte.

An dieser Stelle können Windows-Benutzer auf ein Problem stoßen. Wenn Sie Code::Blocks im Verzeichnis `C:\Programme` beziehungsweise auf einem 64-Bit-System im Verzeichnis `C:\Programme (x86)` installieren und dort keine administrativen Rechte besitzen (oder die Anwendung als normaler Benutzer gestartet haben), bemerken Sie vielleicht, dass Sie Code::Blocks-Einstellungen nicht speichern können. Damit Sie Code::Blocks nutzen können, ohne dass es zu Schwierigkeiten kommt, müssen Sie darauf achten, dass Sie auf dem Ordner über Schreibrechte verfügen, in dem Sie das Programm installiert haben. Am besten installieren Sie Code::Blocks in einem Verzeichnis wie `C:\CodeBlocks`. Sie können das Code::Blocks-Verzeichnis aber auch mit der rechten Maustaste anklicken und im Kontextmenü Als Administrator ausführen wählen, um Code::Blocks mit den entsprechenden Rechten auszuführen.

Code::Blocks startet von nun an immer damit, dass es die IDE öffnet und gegebenenfalls ein Dialogfeld mit Tipps anzeigt (siehe Abbildung 1.7). Bei dem Tipp handelt es sich um eine zufällig ausgewählte Information darüber, wie Sie Code::Blocks sinnvoll verwenden können. Sie lassen sich den nächsten Tipp anzeigen, indem Sie auf Next Tip klicken, oder Sie deaktivieren die Anzeige von Tipps dadurch, dass Sie das Kontrollkästchen vor Show Tips at Startup deaktivieren. Klicken Sie auf Close, wenn Sie den Tipp gelesen haben.

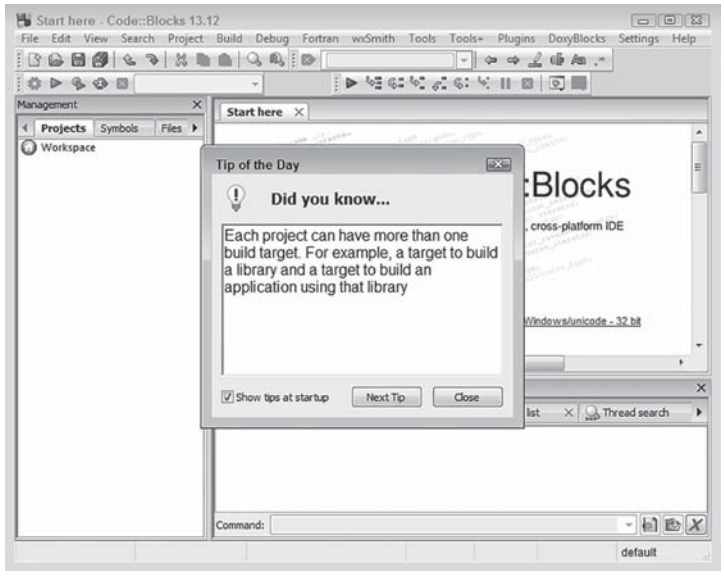

Abbildung 1.7: Code::Blocks stellt ein Dialogfeld mit hilfreichen Tipps zur Verfügung.

Die Beispielprojekte öffnen

Sie erhalten den Quellcode der Beispiele auf der Webseite des Verlags: `www.wiley-vch.de/publish/dt/books/ISBN3-527-71170-8`. Wenn Sie die `.zip`-Datei heruntergeladen haben, müssen Sie sie nur noch auf Ihrer Festplatte entpacken, um an den Quellcode zu gelangen.

Dieser Quellcode ist in Kapitel aufgeteilt und in den Kapiteln nach Projekten gegliedert worden. Wenn Sie zum Beispiel die Zip-Datei im Ordner `C:\Buch\CPP` entpackt haben, finden Sie das erste Beispielprojekt, das zu Kapitel 2 dieses Buches gehört, im Ordner `c:\Buch\CPP\CPP_Beispiele\Kapitel02\HalloSagen` (oder den für Ihre Plattform geltenden Speicherort). In diesem Verzeichnis gibt es die Datei `SayHello.cbp`. Bei dieser Datei handelt es sich um eine Code::Blocks-Projektdatei(`.cbp`), die alles enthält, was Code::Blocks benötigt, um das Projekt zu öffnen. Wenn Sie das erste Projekt gefunden haben, führen Sie auf der Datei `SayHello.cbp` einen Doppelklick aus, und Code::Blocks öffnet automatisch das Projekt für Sie, wie Abbildung 1.8 zeigt.

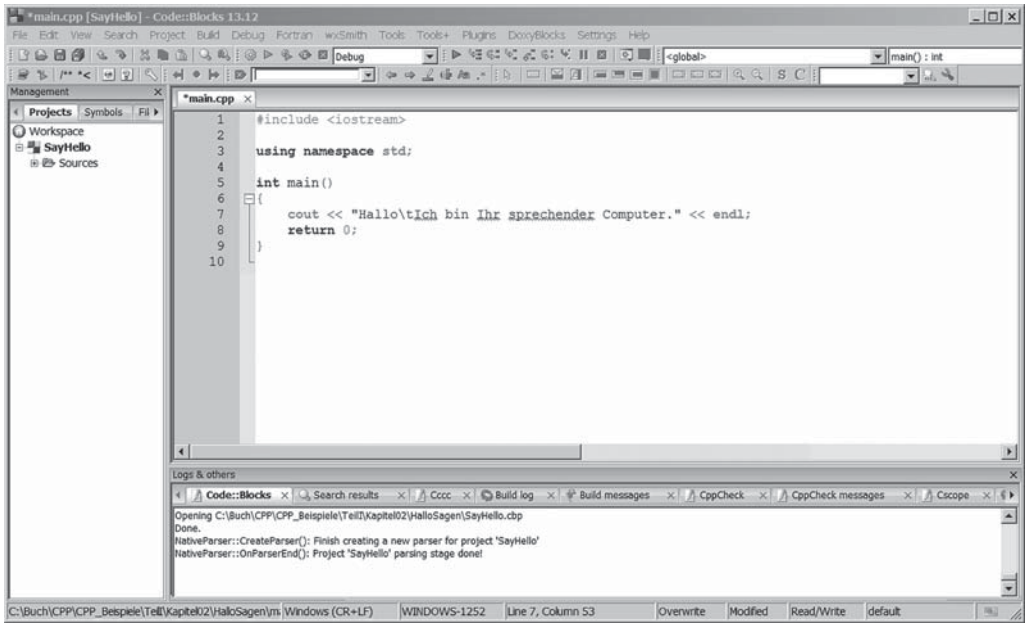

Abbildung 1.8: Zu jedem Beispiel gehört eine `.cbp`-Datei, die das Beispiel in Code::Blocks öffnet.

Wenn Sie es zulassen, dass Tipps angezeigt werden, sehen Sie nun zuerst das Dialogfeld Tip of the Day (das dem in Abbildung 1.7 ähnelt). Klicken Sie, nachdem Sie den Tipp durchgelesen haben, auf Close, und Sie sehen das Projekt. Machen Sie sich über den Inhalt dieses Beispiels noch keine Gedanken. Sie erfahren in Kapitel 2, was es macht und wie es funktioniert. Das Einzige, was Sie im Moment wissen müssen, ist, wie ein Projektbeispiel geöffnet wird, damit Sie den Beispielen in diesem Buch folgen können.

Wenn Sie mit einer anderen IDE als Code::Blocks arbeiten, können Sie statt der `.cbp`-Datei die C++-Datei (`.cpp`) öffnen. Dadurch wird das Codebeispiel angezeigt. C++ speichert den Quellcode in `.cpp`-Dateien.

Die wichtigsten Fenster

Es gibt einige Fenster, die Sie ständig mit den Beispielen dieses Buches nutzen. Diese Fenster werden in den folgenden Abschnitten beschrieben, und Sie müssen sie einfach kennen, wenn Sie mit Code::Blocks loslegen wollen, und im Verlauf dieses Buches werden Sie sehen, dass Code::Blocks noch ein paar Fenster mehr enthält.

Das Fenster »Start here«

Das Fenster Start here, das Abbildung 1.9 zeigt, macht genau das, was sein Name bedeutet – hier starten Sie Code::Blocks erst richtig. Dieses Fenster wird automatisch angezeigt, wenn Sie Code::Blocks direkt und ohne ein Projekt öffnen. Es taucht unmittelbar nach dem Schließen des Dialogfelds Tip of the Day auf.

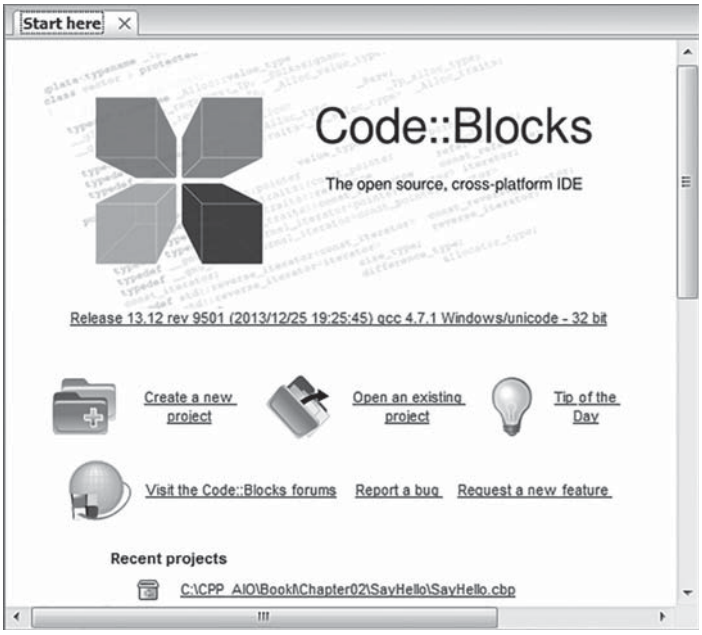

Abbildung 1.9: Benutzen Sie das Fenster »Start here«, um eine neue Sitzung zu beginnen.

Dieses Fenster ist wichtig, weil Sie darüber auch auf verschiedene Code::Blocks-Funktionen zugreifen können und die Möglichkeit erhalten, Änderungen vorzunehmen. Dies sind die Dinge, auf die Sie mithilfe dieses Fensters zugreifen können:

- ✔ **Create a new project:** Bevor Sie Code::Blocks effektiv nutzen können, müssen Sie ein Projekt erstellen. Ein Projekt ist wie ein Container, der die Dateien aufnimmt, die verwendet werden, um die Anwendung zu erstellen. Außerdem speichert das Projekt die Einstellungen, die verwendet werden, um die Entwicklungsumgebung einzurichten und auf eine bestimmte Weise darzustellen.
- ✔ **Open an existing project:** Jedes Mal, wenn Sie die Umgebung wiederherstellen wollen, die Sie während einer früheren Codierungssitzung verwendet haben, öffnen Sie ein Projekt. Das Projekt öffnet dann automatisch alle Quellcodedateien, die zum Projekt gehören, und führt weitere Aufgaben durch, damit Sie genau an der Stelle weitermachen können, an der Sie das Projekt verlassen haben.
- ✔ **Tip of the Day:** Wenn Sie den Tipp des Tages *(Tip of the Day)* vermissen oder diese Funktion einfach nur reaktivieren wollen, klicken Sie auf diese Verknüpfung. Code::Blocks zeigt dann das Dialogfeld Tip oft he Day an, das Abbildung 1.7 zeigt.
- ✔ **Visit the Code::Blocks forums:** Sie können mit den Machern von Code::Blocks nicht direkt kommunizieren. Aber Sie haben die Möglichkeit, direkt mit anderen Benutzern in Kontakt zu treten und von vielen Seiten Unterstützung zu erhalten. Die Macher von Code::Blocks beobachten die Foren, und sie kümmern sich um die Themen, die von anderen nicht geklärt werden können.
- ✔ **Report a bug:** Jedes Programm auf dieser Erde enthält Programmierfehler *(Bugs)*, und die Code::Blocks-IDE bildet davon keine Ausnahme. Es ist wichtig, dass Sie Fehler melden, wenn Sie sie finden, damit sie behoben werden können.
- ✔ **Request a new feature:** Jeder, der lange genug mit einer Anwendung arbeitet, hat irgendwann eine Idee, um diese Anwendung zu verbessern. Die Macher von Code::Blocks sind immer an Ihren phänomenalen Ideen interessiert, weshalb Sie diese Leute lieber früher als später kontaktieren sollten.
- ✔ **Recent projects:** Während Sie mit Code::Blocks arbeiten, legen Sie mehr als ein Projekt an. Statt nun Ihre gesamte Festplatte immer wieder nach einem bestimmten Projekt durchsuchen zu müssen, können Sie diese Funktion verwenden, um das Projekt sehr schnell zu finden. Wenn Sie es dann öffnen wollen, klicken Sie in der Liste Recent Projects, die Ihre letzten Projekte enthält, einfach auf die Verknüpfung.

Selbst wenn Sie das Fenster Start here nicht sehen können, nachdem Sie ein Projekt geöffnet haben, können Sie es immer anzeigen, indem Sie es über Settings|Environment|View|Show »Start here« page aktivieren. Denken Sie daran, dass Ihnen dieses Fenster die Möglichkeit bietet, schnell auf häufig genutzte Funktionen von Code::Blocks zuzugreifen. Natürlich können Sie diese Funktionen auch über die Menüs erreichen. Um zum Beispiel ein neues Projekt anzulegen, können Sie auch File|New|Project wählen.

Das Fenster »Management«

Sinn einer IDE ist es, Ihre Codierungsprojekte auf verschiedene Weise zu verwalten. Aus diesem Grund ist es keine Überraschung, dass Code::Blocks ein Fenster MANAGEMENT kennt, das Abbildung 1.10 zeigt. Dieses Fenster befindet sich normalerweise an der linken Seite des Hauptfensters der IDE, aber Sie können es an eine beliebige Stelle auf dem Bildschirm verschieben, indem Sie die Titelleiste verwenden, um es an seine neue Position zu ziehen.

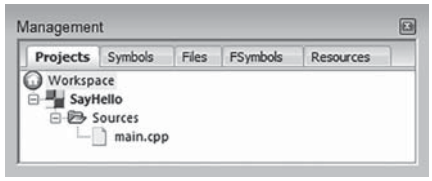

Abbildung 1.10: Das Fenster MANAGEMENT hilft Ihnen bei der Verwaltung Ihrer Code::Blocks-Projekte.

Das Fenster MANAGEMENT hat vier Registerkarten. Die folgende Liste beschreibt, welchem Zweck die einzelnen Registerkarten dienen:

- ✔ **Projects:** Das Gruppieren der Dateien, die benötigt werden, um eine Anwendung zu erstellen, hilft bei der Verwaltung dieser Dateien. Die Gruppierung von Anwendungsdateien wird *Projekt* genannt, und die Hilfe von Code::Blocks beim Erstellen und Pflegen von Projekten ist einer der Wege, auf denen die Anwendungsentwicklung erleichtert wird.
- ✔ **Symbols:** Anwendungen enthalten eine Reihe von Symbolen, zum Beispiel die Namen von Funktionen. Sie verwenden die Registerkarte SYMBOLS, um die Symbole zu finden, die Sie im Rahmen einer Anwendung benötigen. Machen Sie sich im Moment noch keine Gedanken über Symbole, aber vielleicht finden auch Sie heraus, dass diese Registerkarte dabei helfen kann, Zeit und Mühen zu sparen, indem es einfacher wird, bestimmte Puzzleteile Ihrer Software aufzufinden.
- ✔ **Files:** Es kann ziemlich zeitaufwendig sein, den Code und die Ressourcen zu lokalisieren, die Sie dem aktuellen Projekt hinzufügen müssen. Die Registerkarte FILES sorgt für eine Methode, sich im Dateisystem zurechtzufinden. Sie können dann Dateien, die Sie benötigen, mit der rechten Maustaste anklicken und die Einträge des Kontextmenüs nutzen, um Aufgaben wie das Hinzufügen der Datei zum aktuellen Projekt zu erledigen.
- ✔ **Resources:** Grafische Anwendungen verlangen Dialogfelder und andere visuelle Elemente, die C++ als Ressourcen behandelt. Die Registerkarte RESOURCES enthält eine Liste dieser Ressourcen, wodurch Sie sie leicht finden können und mehrere Möglichkeiten haben, sie zu verwalten.

Bei der Registerkarte RESOURCES handelt es sich um ein sogenanntes *Feature*, das von erfahrenen Entwicklern benutzt wird. Sie beschäftigen sich normalerweise erst dann mit dieser Programmfunktion, wenn Sie sich entschließen, grafische Anwendungen zu erstellen, die eine Kombination aus C++ und dem Plug-in wxWidgets verwenden. (Dieses Plug-in wird unter Windows automatisch installiert, während sich sowohl Mac- als auch Linux-Entwickler eigenhändig um die Installation kümmern müssen.) Wie Sie solche Anwendungen erstellen, über-

schreitet den Rahmen dieses Buches, aber Sie finden unter `http://wiki.codeblocks.org/index.php?title=WxSmith_tutorial:_Hello_world` ein einfaches Beispiel eines solchen Projekts.

Das Fenster »Logs & others«

Code::Blocks hilft Ihnen dabei, allen möglichen Aktivitäten auf der Spur zu bleiben. Wenn Sie zum Beispiel aus Ihrem Quellcode eine Anwendung erstellen (was in Code::Blocks *Building* genannt wird), sehen Sie Meldungen, die Sie darüber informieren, wie dies abgelaufen ist (siehe Abbildung 1.11). Die Beispiele in diesem Buch werden Ihnen dabei helfen zu verstehen, wann Sie die verschiedenen Registerkarten mit Protokollen und anderen Informationen (wie die Registerkarte DEBUGGER) verwenden sollten, um festzustellen, wie Ihre Anwendung funktioniert.

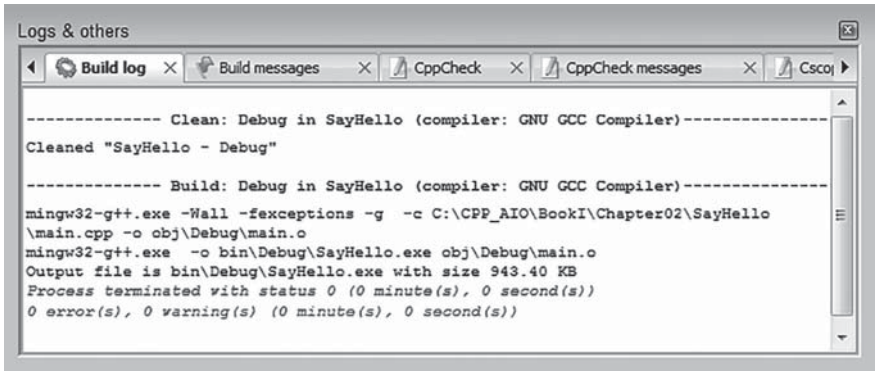

Abbildung 1.11: Nutzen Sie das Fenster LOGS & OTHERS*, um zu verstehen, wie die Anwendung funktioniert.*

Die Registerkarten, die Sie in diesem Fenster sehen, hängen von den Optionen ab, die Sie in Code::Blocks eingeschaltet haben, und von den Aufgaben, die Sie erledigen. Code::Blocks sucht normalerweise die Registerkarten, die Sie benötigen, automatisch aus. Wenn Sie eine bestimmte Registerkarte schließen wollen, klicken Sie oben auf der Karte auf das X. Um eine Registerkarte anzuzeigen, die Sie nicht sehen, klicken Sie mit der rechten Maustaste auf eine beliebige Registerkarte und wählen im Kontextmenü in der Liste der Option TOGGLE den gewünschten Eintrag aus.

Einen Compiler auswählen

Code::Blocks unterstützt eine Reihe von Compilern. Dieses Buch verwendet GCC, weil dieser Compiler auf allen Zielplattformen eingesetzt werden kann und über eine erstklassige Unterstützung von C++ 14 verfügt. Die meisten Code::Blocks-Installationen wählen diesen Compiler automatisch aus. Es gibt sehr viele Gründe, sich für GCC zu entscheiden. Sollten Sie aber auf Ihrer Plattform einen anderen Compiler gewählt haben, kann dies beim Kompilieren der Beispiele zu Problemen führen. Nicht jeder Hersteller eines Compilers unterstützt das gesamte Spektrum von C++ 14, oder es sind Komponenten auf andere Weise implementiert worden

als im GCC. Die folgenden Schritte helfen Ihnen dabei, herauszufinden, ob auf Ihrem System GCC läuft, und falls dies nicht der Fall ist, Ihre Konfiguration zu ändern:

1. **Starten Sie Code::Blocks.**

   Hier interessiert jetzt nicht, ob Sie ein Projekt ausgewählt haben oder nicht. Das Einrichten eines Compilers ist in beiden Fällen identisch.

2. **Wählen Sie Settings|Compiler.**

   Sie sehen das Dialogfeld Compiler Settings, das Abbildung 1.12 zeigt.

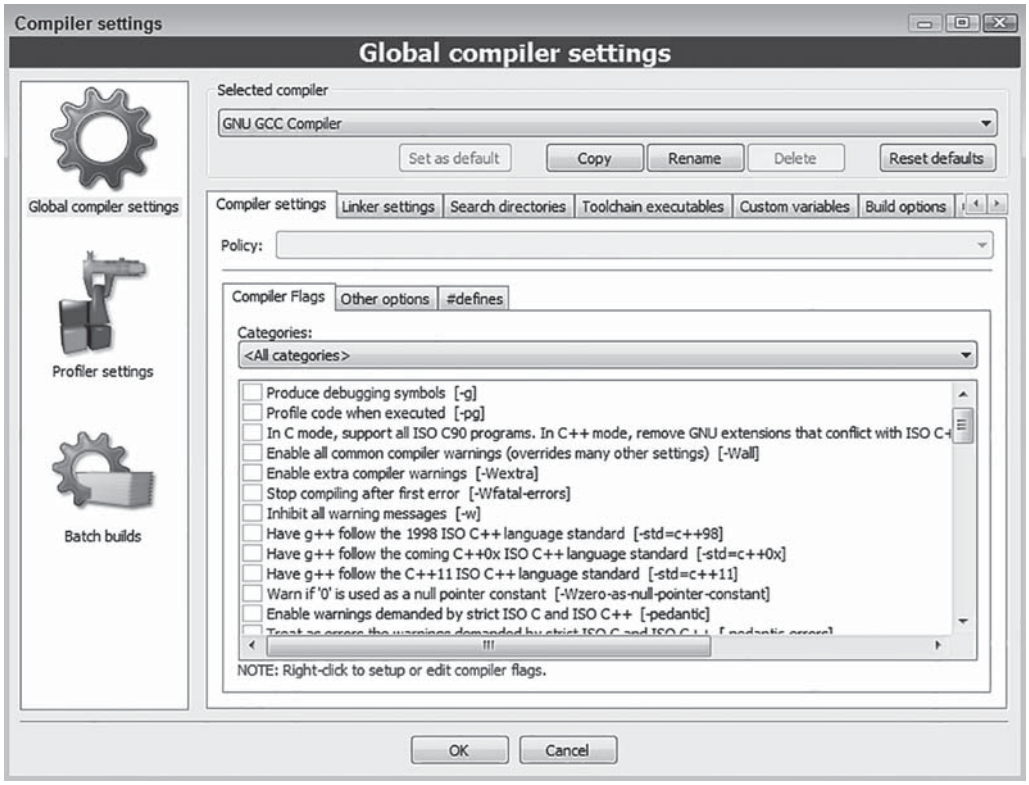

Abbildung 1.12: Richten Sie Code::Blocks ein, damit es den GCC-Compiler verwendet, um die Beispiele dieses Buches ablaufen zu lassen.

3. **Klicken Sie links im Fenster auf Global compiler settings, um die globalen Compiler-Einstellungen angezeigt zu bekommen.**

4. **Prüfen Sie, ob der GNU GCC Compiler (oder seine Entsprechung für Ihre Plattform) in der Liste Selected compiler ausgewählt ist.**

   Die Liste kann mehrere GCC-Compiler enthalten. Am besten geeignet ist der GNU GCC Compiler, weil er die größtmögliche Kompatibilität mit den Beispielen des Buches bietet.

Wenn dieser Compiler (oder seine Entsprechung für Ihre Plattform) ausgewählt ist, machen Sie mit Schritt 7 weiter.

5. **Wählen Sie gegebenenfalls den GNU GCC Compiler (oder seine Entsprechung für Ihre Plattform) in der Liste Selected compiler aus.**

   Die Schaltfläche Set as default wird aktiviert, mit der Sie Ihre Auswahl zum Standard machen können.

6. **Klicken Sie auf Set as default.**

   Dieser Schritt sorgt dafür, dass der GNU GCC Compiler selbst dann für Ihre Projekte genommen wird, wenn Sie nur den heruntergeladenen Quellcode öffnen wollen.

7. **Klicken Sie auf OK.**

8. **Beenden Sie Code::Blocks.**

   Sie sehen das Dialogfeld Layout changed, das auf das geänderte Layout von Code::Blocks hinweist und Sie fragt, ob die Änderungen dauerhaft gespeichert werden sollen.

9. **Klicken Sie auf Yes.**

   Ihre Änderungen werden gespeichert, und Code::Blocks schließt sich.

Andere Entwicklungsumgebungen verwenden

Auch wenn sich dieses Buch auf die Kombination aus der Entwicklungsumgebung (IDE) Code::Blocks und dem Compiler GCC konzentriert, können Sie das, was Sie hier kennenlernen, auf jede andere Kombination aus IDE und Compiler anwenden. Eigentlich benötigen Sie als Einziges den Compiler. Die meisten Entwickler setzen eine IDE nur deshalb ein, weil sie vieles einfacher macht (und wir lieben einfache Dinge). Aber vielleicht enthält Code::Blocks nicht die Funktionen, die Sie benötigen, oder es ist für Sie zu aufwendig, das Programm zu verwenden.

Die Entscheidung für oder gegen eine IDE ist eine persönliche Angelegenheit, und die meisten Entwickler haben bestimmte Gründe, warum sie sich für eine Entwicklungsumgebung entscheiden. So verwende ich zum Beispiel mehrere IDEs und entscheide mich jedes Mal anhand dessen, was für ein bestimmtes Projekt benötigt wird. Es ist also nicht zwingend notwendig, die ganze Zeit über immer dieselbe IDE zu verwenden. IDEs stellen Werkzeuge für die Verwaltung bereit, während sich Compiler darum kümmern, wie der Quellcode interpretiert und in eine ausführbare Datei umgewandelt wird. Beide Anwendungen erledigen also vollständig unterschiedliche Aufgaben.

GCC ist als Compiler erste Wahl, weil er von vielen Entwicklungsumgebungen unterstützt wird. Wenn Sie sich für eine andere als die hier im Buch vorgestellte IDE entscheiden, haben wir nichts dagegen. Wir gratulieren Ihnen sogar zu Ihrem Vorhaben, einen anderen Weg einzuschlagen!

Hier ein paar alternative IDEs, die Sie in Ihre Überlegungen einbeziehen sollten:

- ✔ **CodeLite:** `http://codelite.org/`
- ✔ **Dev-C++:** `http://dev-c.soft32.com/free-download/`
- ✔ **Eclipse:** `http://www.eclipse.org/downloads/` unter der Voraussetzung, dass das Programm zusammen mit C/C++ Development Tooling (CDT) (`http://www.eclipse.org/cdt/`) verwendet wird
- ✔ **Emacs:** `http://www.gnu.org/software/emacs/`, wenn die Anwendung zusammen mit dem Emacs Code Browser (ECB) (`http://ecb.sourceforge.net/`) verwendet wird
- ✔ **Netbeans:** `https://netbeans.org/downloads/`
- ✔ **Qt Creator:** `http://sourceforge.net/projects/qtcreator.mirror/`