

3 Erste Schritte mit dem Visual C++-Compiler

Wie Ihnen aus den Ausführungen in Kapitel 1 »Was ist Programmieren?« bekannt ist, ist der erste Schritt in der Programmentwicklung das Anlegen einer Quelltextdatei. Es erhebt sich also die Frage, welcher Editor zu verwenden ist. Wenn Sie im letzten Kapitel den Visual C++-Compiler auf Ihrem Computer installiert haben, dann werden Sie den in dieser IDE integrierten Editor verwenden. Die dazu erforderliche Vorgehensweise gestaltet sich für den Neuling scheinbar etwas kompliziert. Der Grund: Visual C++ verwaltet Quellcodedateien in Projekten, diese wiederum in Arbeitsbereichen. Dies ist halb so schlimm, wie es sich zunächst anhört. Sie werden sich bald daran gewöhnt haben und später die damit verbundenen Vorteile zu schätzen wissen (wenn es darum geht, den Quellcode auf mehrere Dateien zu verteilen).

Wir werden Ihnen in diesem Kapitel zeigen, was Sie tun müssen, um in der Visual C++-Umgebung eine Quellcodedatei anzulegen (Abschnitt 3.2 »Quellcodedatei anlegen unter Visual C++«). Der folgende Abschnitt 3.1 »Was für einen Editor sollten Sie verwenden?« ist für diejenigen Leser gedacht, die mit einem Kommandozeilen-Compiler arbeiten.

3.1 Was für einen Editor sollten Sie verwenden?

Grundsätzlich steht es Ihnen frei, welchen Editor Sie verwenden. Sie müssen nur darauf achten, den Quellcode als reinen Text, im so genannten ASCII-Format, zu speichern. Andernfalls werden vom Textverarbeitungsprogramm Steuerzeichen eingefügt, die vom Compiler bei der anschließenden Übersetzung nicht interpretiert werden können.

Wie gesagt, stellt sich die Frage, was für einen Editor Sie benutzen sollten, für Sie nur dann, wenn Sie mit einem Kommandozeilen-Compiler arbeiten. Wenn Sie den Visual C++-Compiler oder eine andere IDE einsetzen, greifen Sie auf den integrierten Editor dieser Entwicklungsumgebung zurück.

Sofern Sie mit einem Windows-Betriebssystem arbeiten (und einen Kommandozeilen-Compiler verwenden), können Sie für die Eingabe Ihrer Quelltexte auch den Windows-Editor benutzen. Fürs Erste ist dieser völlig ausreichend.



TIPP

Den Editor des Windows-Betriebssystems erreichen Sie mit der Befehlsfolge `START/PROGRAMME/ZUBEHÖR/EDITOR`.

Für Linux-Anwender bietet sich der Editor *vi* oder *emacs* an.

Unter Umständen stellt sich die Frage, unter welchem Namen die Quellcodedatei zu speichern ist, gleich beim Aufruf des Editors (so z.B. wenn

Sie den *vi* verwenden). Dazu ist zu sagen: Den Namen (exklusive Dateierweiterung) können Sie – unter Beachtung der Vorgaben Ihres Betriebssystems – grundsätzlich frei wählen. Entscheidend ist allein die Dateierweiterung. Diese richtet sich nach dem Compiler, der den Quellcode später übersetzen soll.



Zum besseren Verständnis noch einmal der Hinweis: Wenn im Sinne der Kapitelüberschrift (»Erste Schritte ...«) vom Visual C++-Compiler gesprochen wird, ist damit die gesamte IDE (einschließlich des integrierten Compilers) gemeint. In einem anderen Zusammenhang kann dagegen mit »Visual C++-Compiler« allein die zum Übersetzungsvorgang benötigte Software (also der in der IDE integrierte Compiler) gemeint sein. Wenn der Compiler extern aufgerufen wird, wie das bei einem Kommandozeilen-Compiler der Fall ist, bezeichnet das Wort »Compiler« ausschließlich die Software zum Übersetzen des Quellcodes.

Schauen Sie daher in Ihrer Compilerdokumentation nach. In der Regel ist die Endung für C++-Quellcodedateien *.cpp* (für »C plus plus«). Dies trifft auf den erwähnten Borland C++-Compiler 5.5 und ebenso auf den GCC-Compiler unter Linux zu. Falls Sie mit einer integrierten Entwicklungsumgebung arbeiten, brauchen Sie sich darum nicht zu kümmern, da diese automatisch dafür sorgt, dass die Quellcodedateien mit der richtigen Endung gespeichert werden.



Lesern mit Kommandozeilen-Compiler sei empfohlen: Speichern Sie Ihre Quellcodedateien in einem separaten Verzeichnis und rufen Sie später aus diesem Verzeichnis heraus den Compiler auf.

3.2 Quellcodedatei anlegen unter Visual C++

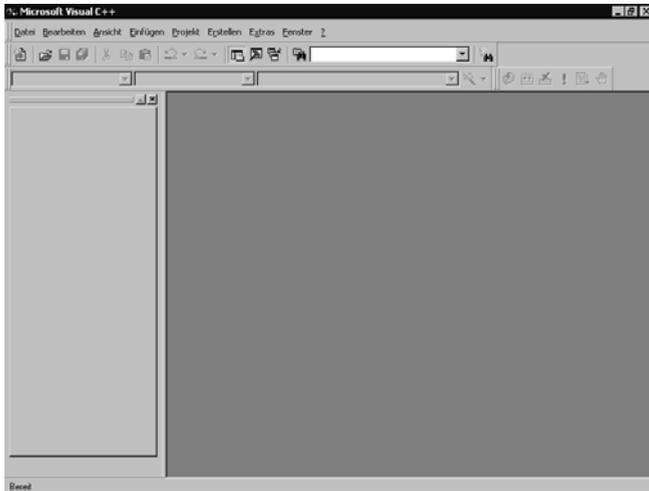
Wie bereits erwähnt, verwaltet Visual C++ Quellcodedateien in Projekten und diese wiederum in Arbeitsbereichen. Das heißt, Sie müssen als Erstes einen Arbeitsbereich anlegen.

1. Starten Sie Visual C++: Wählen Sie in der Taskleiste unter START/PROGRAMME die Programmgruppe MICROSOFT VISUAL C++ 6.0 und hier den gleich lautenden Eintrag MICROSOFT VISUAL C++ 6.0.

Nach dem Aufruf sollte sich Visual C++ wie in Abbildung 3.1 präsentieren (nach eventuellem Schließen eines Fensters mit dem Titel TIPPS UND TRICKS).

Das linke Fenster, genannt Arbeitsbereichsfenster, dient der Verwaltung der Projektdateien. Es ist leer, da noch keine Arbeitsbereiche, Projekte und Quellcodedateien vorhanden sind. Zunächst werden wir uns um einen Arbeitsbereich kümmern.

Bild 3.1:
Visual C++ nach
dem ersten Aufruf



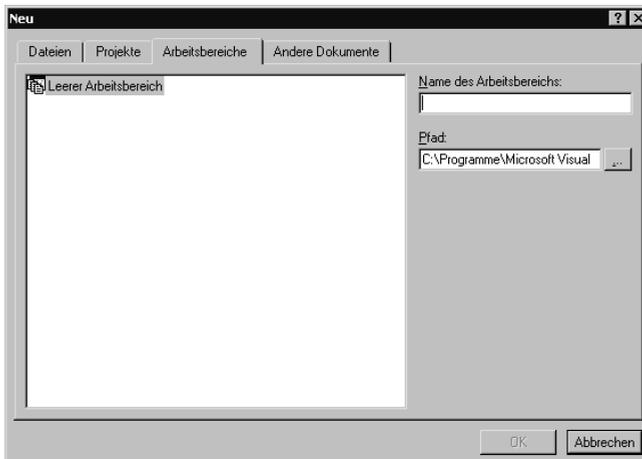
HINWEIS

Falls das Arbeitsbereichsfenster aus irgendeinem Grund nicht sichtbar ist, holen Sie es mit dem Menübefehl ANSICHT/ARBEITSBEREICH oder der Tastenkombination **Alt + O** ins Bild.

3.2.1 Arbeitsbereich anlegen

2. Wählen Sie in der Menüleiste DATEI/NEU... und klicken Sie im folgenden Dialogfenster auf das Register ARBEITSBEREICHE.

Bild 3.2:
Anlegen eines
Arbeitsbereichs



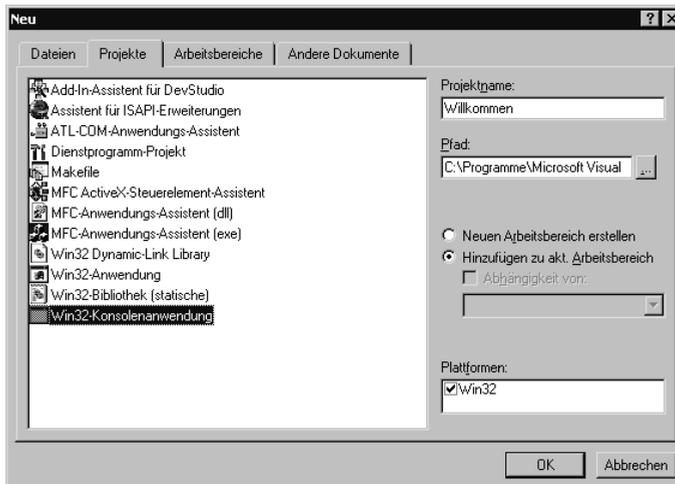
3. Geben Sie in das dafür vorgesehene Textfeld einen Namen für den neuen Arbeitsbereich ein und bestätigen Sie mit OK.

Im Register DATEIEN des Arbeitsbereichsfensters erscheint nun ein Eintrag für den erstellten Arbeitsbereich. In diesen müssen wir nun ein Projekt integrieren.

3.2.2 Ein Projekt in den Arbeitsbereich einfügen

4. Um ein Projekt anzulegen, wählen Sie im Menü erneut die Befehlsfolge DATEI/NEU... und klicken auf das Register PROJEKTE.
5. Wählen Sie in der Liste WIN32-KONSOLENANWENDUNG, geben Sie in das Textfeld rechts oben einen Namen für Ihr Projekt ein und aktivieren Sie die Option HINZUFÜGEN ZU AKT. ARBEITSBEREICH.

Bild 3.3:
Anlegen eines Projekts



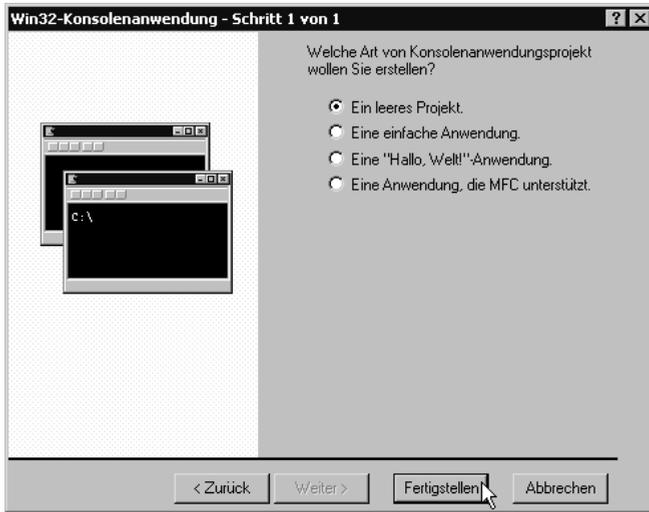
Wir haben unseren Arbeitsbereich sinnigerweise »Erster« und das Projekt im Hinblick auf unser erstes C++-Programm »Willkommen« genannt.



Die Wahl der Option HINZUFÜGEN ZU AKT. ARBEITSBEREICH ist wichtig, da ansonsten ein neuer Arbeitsbereich zusammen mit einem Projekt angelegt wird. Natürlich hätten wir dies zu Beginn ausnutzen können, um einen Schritt zu sparen. Vermutlich werden Sie im Weiteren jedoch mehrere Projekte in ein und denselben Arbeitsbereich einbinden, also fürs Erste keinen weiteren Arbeitsbereich benötigen. Deshalb wird davon ausgegangen, dass die Darstellung in sich wiederholenden Einzelschritten für Sie leichter nachzuvollziehen ist.

6. Bestätigen Sie den oben genannten Dialog mit OK.
7. Anschließend belassen Sie es bei der Option EIN LEERES PROJEKT und klicken auf FERTIGSTELLEN.

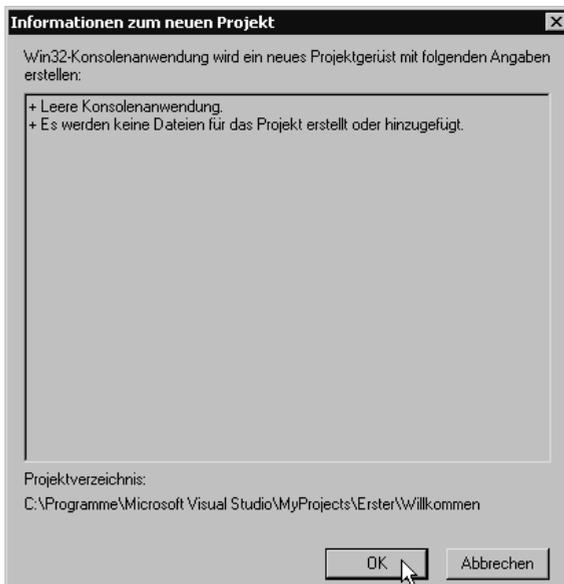
Bild 3.4:
Leeres Projekt



Leeres Projekt bedeutet, dass Sie Visual C++ nicht beauftragen, Code für Sie zu generieren. Was Visual C++ in diesem Fall für Sie täte (zu Ihrer Entlastung), wollen Sie ja gerade erlernen.

8. Dem folgenden Infofenster (Abbildung 3.5) können Sie entnehmen, in welchem Verzeichnis Ihre zukünftigen Projektdateien abgelegt werden. Klicken Sie auf OK.

Bild 3.5:
Infofenster



Im Arbeitsbereichsfenster (Registerblatt DATEIEN) hat sich inzwischen einiges getan. Wenn Sie den Eintrag für den Projektordner aufklappen (Klick auf das Symbol +), sollte sich in Abhängigkeit von Ihrer Namenswahl folgendes Bild ergeben.

Bild 3.6:
Projekt *Willkommen* mit Unterordnern



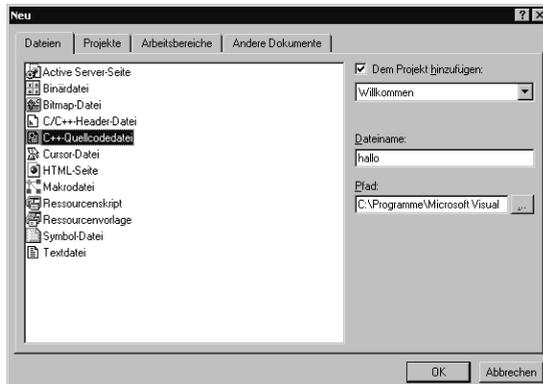
Die Struktur dürfte Ihnen vom Windows-Explorer geläufig sein: ein Verzeichnis für den Arbeitsbereich, ein Unterverzeichnis für das darin integrierte Projekt und auf der untersten Ebene drei vorläufig leere Ordner. Wobei Sie sich sicher denken können, wofür der Ordner *Quellcodedateien* steht. Was es mit Headerdateien auf sich hat, werden Sie im weiteren Verlauf des Buches noch erfahren (siehe Abschnitt 21.2 »Funktionsprototypen«). Ressourcendateien beziehen sich auf die GUI-Programmierung, sollen uns also nicht weiter interessieren.

3.2.3 Endlich die Quellcodedatei ...

Nun wollen wir das tun, worauf es uns eigentlich ankommt: eine Quellcodedatei anlegen. Und wie immer, wenn Sie ein neues Projekt, einen neuen Arbeitsbereich oder eben eine neue Quelltextdatei anlegen wollen, benutzen Sie die Befehlsfolge DATEI/NEU... und wählen das entsprechende Register.

9. Wählen Sie also DATEI/NEU..., Register DATEIEN.
10. Selektieren Sie in der Liste C++-QUELLCODEDATEI und geben Sie im Textfeld mit der Überschrift DATEINAME: einen Namen (ohne Erweiterung) für die Quellcodedatei ein (siehe Abbildung 3.7).

Bild 3.7:
Anlegen der Quellcodedatei



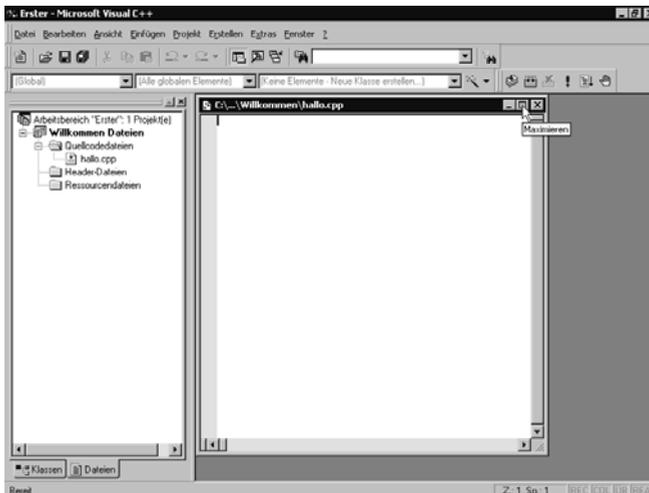


Die Namensgebung richtet sich zwar nach den Vorgaben des Betriebssystems. Von Leer- oder Sonderzeichen ist jedoch allgemein abzuraten. Es sei darauf hingewiesen, dass die einzugebenden Namen von Projekt und Quellcodedatei sich nicht unterscheiden müssen. Der Name *Willkommen* (oder *willkommen*) – anstelle von *hallo* – wäre also auch für unsere Quellcodedatei zulässig gewesen.

11. Klicken Sie auf OK.

Nun ist in Ihrem Projekt eine Quellcodedatei enthalten. Die Endung *.cpp* wurde automatisch angefügt, wie im Arbeitsbereichsfenster nach Aufklappen des Ordners *Quellcodedateien* zu erkennen.

Bild 3.8:
Editieren der
Quellcodedatei
(*hallo.cpp*)



Rechts neben dem Arbeitsbereichsfenster befindet sich Ihr Editor. Dort werden Sie den Quelltext Ihres ersten Programms eingeben. Damit das Editorfenster die maximale Fläche einnimmt, vergrößern Sie es durch einen Klick auf die Maximieren-Schaltfläche (wie in Abbildung 3.8 gezeigt).

Einige Angaben zur aktuellen Quellcodedatei sind in der Titelleiste des Fensters ersichtlich: Pfad und Name der Quellcodedatei sowie Name des zugehörigen Projekts.

Beachten Sie:

- In einem Arbeitsbereich können sich mehrere Projekte befinden.
- Aus einem Projekt entsteht eine ausführbare Datei (*.exe*-Datei, Computerprogramm).
- Bis auf weiteres gilt: eine Quellcodedatei = ein Projekt = eine ausführbare Datei (das soll bedeuten, dass aus jeweils einer Quellcodedatei, die in einem gesonderten Projekt verwaltet wird, genau eine ausführbare Datei entsteht).



In Kapitel 21 »Funktionen« erfahren Sie, wie sich der Quellcode auf mehrere Quellcodedateien (die dann ebenfalls innerhalb eines Projekts verwaltet werden) verteilen lässt. In Präzisierung obiger Gleichung ergibt sich dann:

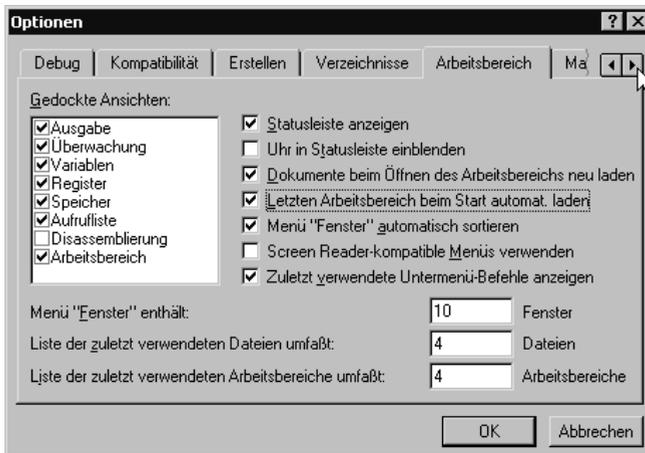
eine oder mehrere Quellcodedateien = ein Projekt = eine ausführbare Datei

3.2.4 Letzten Arbeitsbereich beim Start automatisch laden

Wenn Sie Visual C++ beenden, zeigt sich beim nächsten Start ein leeres Arbeitsbereichsfenster. Dann laden Sie den vorhandenen Arbeitsbereich mit DATEI/ZULETZT GEÖFFNETE ARBEITSBEREICHE ▶ und Auswahl des entsprechenden Arbeitsbereichs in das Arbeitsbereichsfenster.

Es geht jedoch bequemer mit der Einstellung LETZTEN ARBEITSBEREICH BEIM START AUTOMATISCH LADEN. Diese Option steht nach Auswahl von EXTRAS/OPTIONEN auf der Registerkarte ARBEITSBEREICH zur Verfügung. Möglicherweise müssen Sie etwas nach rechts scrollen, um zu diesem Register zu gelangen (siehe Mauszeiger in Abbildung 3.9).

Bild 3.9:
Letzten Arbeitsbereich beim Start automatisch laden



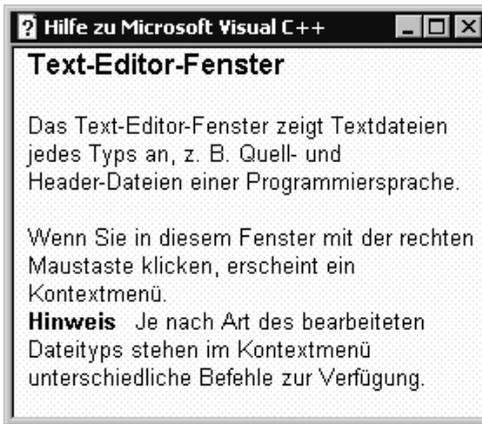
Klicken Sie also auf das Registerblatt ARBEITSBEREICH und dann in das leere Kontrollkästchen neben LETZTEN ARBEITSBEREICH BEIM START AUTOMAT. LADEN. Bestätigen Sie diese Einstellung anschließend mit OK. Nun präsentiert sich Visual C++ bei jedem Start genauso wie beim Ende der letzten Sitzung.

3.3 Mit der Online-Hilfe arbeiten

Vermutlich wird Ihnen die Online-Hilfe zum jetzigen Zeitpunkt noch nicht viel nützen, was die Sprache C++ angeht (ein bisschen Schmökern rentiert sich allemal). Bezüglich der Bedienung Ihres Compilers (gemeint ist die gesamte IDE) kann sie Ihnen aber jetzt schon sehr hilfreich sein.

Kontextbezogene Hilfe erhalten Sie mit der Taste **[F1]**. Sie reagiert auf verschiedene Bereiche der Benutzeroberfläche wie Dialogfenster und Auswahloptionen. So etwa, wenn sich der Cursor im leeren Editorfenster befindet.

Bild 3.10:
Kontextbezogene Hilfe

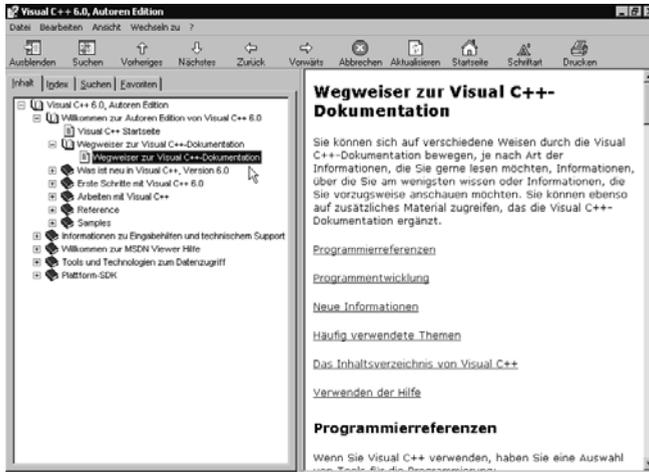


Außerdem bekommen Sie mit **[F1]** alle Hilfedokumente zu einem bestimmten Begriff angezeigt (soweit vorhanden), nachdem Sie diesen an beliebiger Stelle in den Editor eingetippt und den Cursor in das Wort (oder unmittelbar dahinter) gesetzt haben.

Um den vollen Umfang der Dokumentation zu nutzen, klicken Sie in der Menüleiste auf das Fragezeichen und dann auf INHALT, SUCHEN... oder INDEX... (alle drei Möglichkeiten stehen im anschließenden Dialog als Registerkarte zur Verfügung).

- Auf der Registerkarte SUCHEN können Sie in das Textfeld ebenfalls einen Suchbegriff eingeben. Nach Klicken auf THEMEN AUFLISTEN erhalten Sie alle zu diesem Begriff existierenden Hilfedokumente. Um sich ein bestimmtes anzeigen zu lassen, doppelklicken Sie auf den Listeneintrag oder bedienen Sie sich der Schaltfläche ANZEIGEN.
- INDEX listet eine Vielzahl von Begriffen auf. Entweder scrollen Sie in der Liste zu dem gewünschten Begriff oder geben Sie in das Textfeld einen oder mehrere Anfangsbuchstaben ein. Das zugehörige Hilfedokument erhalten Sie wiederum durch Doppelklicken auf den Listeneintrag oder mit der Schaltfläche ANZEIGEN (ganz unten).
- Die Registerkarte INHALT zeigt sämtliche Hilfedokumente nach Sachthemen geordnet in einer Verzeichnisstruktur. Das entsprechende Dokument erscheint nach Klicken auf den Ordneintrag.

Bild 3.11:
Online-Hilfe von
Visual C++



Obige Ausführungen sowie weitere Anmerkungen zur Visual C++-Entwicklungsumgebung (und auch zu anderen Compilern) auf den folgenden Seiten sind gewissermaßen als kleine Zugabe zu Ihrer Unterstützung gedacht. Wie schon erwähnt, sind Sie jedoch für die Bedienung Ihres Compilers letzten Endes selbst verantwortlich. Das Suchen nach Begriffen wie »Benutzeroberfläche«, »Editor« und »Compiler« kann sich daher schon jetzt oder sehr bald als hilfreich erweisen. Ansonsten empfehlen wir Ihnen, gegebenenfalls Begriffe, die Ihnen im Laufe der Lektüre dieses Buches begegnen, als Basis für die Suche in der Online-Hilfe zu verwenden.