



Autorenversion

VC++ installieren und verwalten

Das **Besondere** an dieser Ausgabe: Eine **Entwicklungsumgebung** wird mitgeliefert. Hier die **Hinweise** zur Installation.

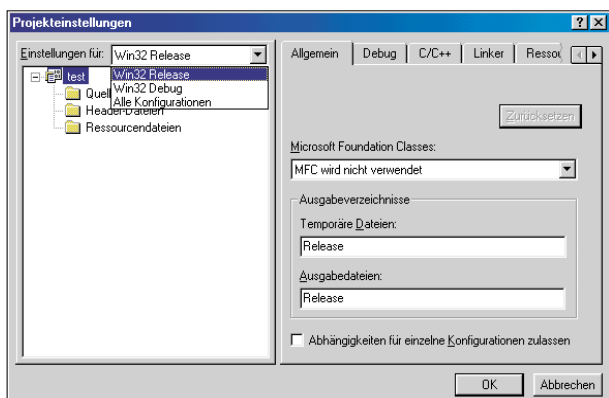
THOMAS WÖLFER

Auf der CD zu diesem Sonderheft finden Sie neben den Beispielquellcodes und Programmen auch die Autorenversion von VC++ 6.0 aus dem Hause Microsoft. Der wesentliche Unterschied zwischen der Autorenversion und der Vollversion dieser Entwicklungsumgebung ist, dass Sie mit der Autorenversion keine kommerziellen Programme entwickeln dürfen. Aufs Lernen hat das aber keinerlei Einfluss.

Viel falsch machen kann man bei der Installation von VC++ nicht. Jedoch hat die C++-Entwicklungsumgebung von Microsoft eine ganze Menge an Optionen und Features – und die im Einzelnen zu besprechen, was für eine benutzerdefinierte Installation eigentlich notwendig wäre, das würde den Umfang dieses Beitrags bei weitem sprengen. Trotzdem sollen Sie in die Lage versetzt werden, alle Beispiele aus diesem Sonderheft direkt und ohne viel Aufwand laden, übersetzen und ausprobieren zu können. Darum macht es Sinn, wenn Sie die gleichen Optionen bei der Installation auswählen, die auch die Auto-

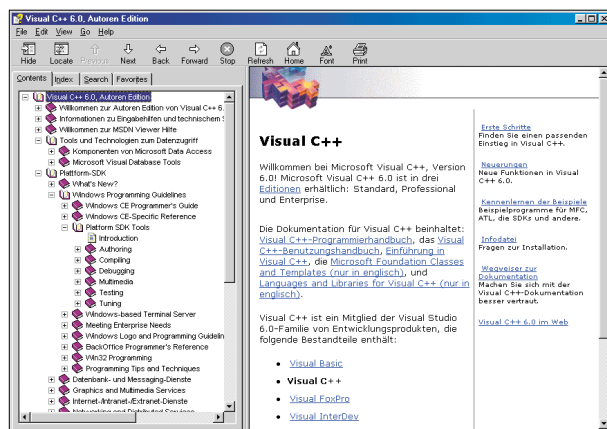
ren der einzelnen Beispiele verwendet haben.

Um das am einfachsten zu gestalten, gehen Sie wie folgt vor: Nachdem Sie Setup gestartet haben, akzeptieren Sie einfach alle vorgeschlagenen Installationsparameter. Der wirklich wichtige Punkt kommt nach einigen Dialogboxen, denn dann fragt Setup an, welche Installationsvariante gewählt



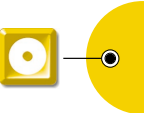
VC++ unterscheidet zwischen "Debug"- und "Release"-Versionen des Programms.

werden soll. Hier sollten Sie auf jeden Fall die Option *Standard* wählen – das stellt sicher, dass auch auf Ihrem System alle für die Beispielprojekte benötigten Header-Dateien und Libraries installiert

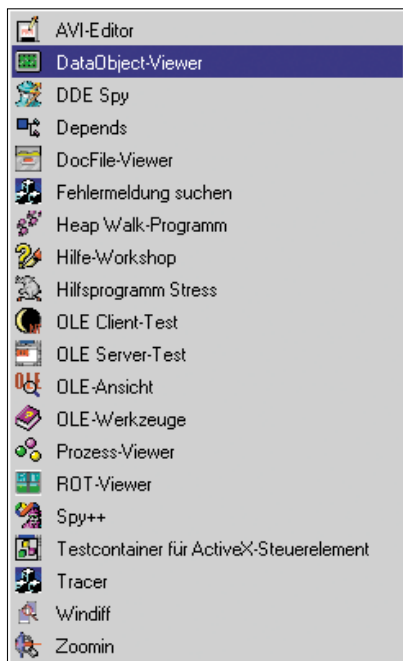


werden. Die Installation selbst läuft je nach Geschwindigkeit Ihres CD-ROM-Laufwerkes in wenigen Minuten ab. Die Entwicklungsumgebung von VC++ selbst benötigt dabei circa 40 MByte Platz auf Ihrer Festplatte. Kurz vor dem Ende der Installation zeigt Setup dabei noch eine Dialogbox an, die Sie über die Position einer bestimmten Batch-Datei aufklärt. Das ist aber zunächst nicht wichtig, sodass Sie diese Dialogbox einfach schließen können. Ist Setup abgelaufen, sind Sie aber noch nicht fertig, denn nun wird automatisch die Installation der Dokumentation gestartet, und die hat es in sich, denn es werden circa 140 MByte auf Ihrer Festplatte benötigt.

Im Wesentlichen handelt es sich bei dieser Dokumentation um den für VC++ gültigen Auszug aus der MSDN CD. Dazu zählt nicht nur die reine Beschreibung der Entwicklungsumgebung, sondern auch ein C- und C++-Referenzteil, Referenzen und Tutorials für die unterschiedlichen Bibliotheken und Projekttypen, die Sie



mit VC++ anlegen können sowie die grundlegende Dokumentation des Win32 Platform SDK. Diese Platform-SDK-Dokumentation ist unverzichtbar, denn die enthält so un-



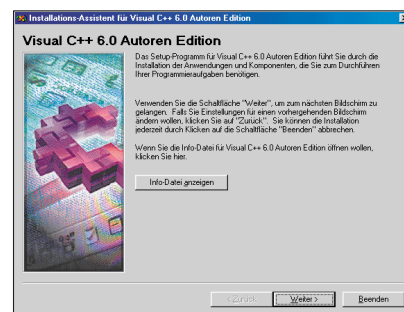
ZUSÄTZLICH ZU DEN TOOLS aus der IDE werden bei VC++ eine ganze Reihe externer Tools mitgeliefert.

gefähr alles, was man über die Programmierung von Anwendungen unter 32bit Windows wissen muss. Sie werden feststellen, dass der Umfang an Dokumentation etwas erschlagend wirkt – wem das aber trotzdem nicht reicht, der kann die komplette MSDN-Dokumentation zu Rate ziehen, denn die steht unter msdn.microsoft.com kostenlos zum Nachschlagen bereit.

Ist auch die Dokumentation installiert, steht der Arbeit nichts mehr im

Wege: Setup hat für VC++ einen extra Eintrag im Startmenü angelegt, von wo aus Sie nicht nur die IDE selbst, sondern auch noch verschiedene zugehörige Tools starten können. Die hauptsächliche Arbeit wird aber mit der VC++ IDE geleistet. Deshalb zunächst einmal ein paar Worte über die grundlegenden Features der IDE (Integrated Development Environment = Integrierte Entwicklungsumgebung).

Bei der Arbeit mit VC++ verbringen Sie die meiste Zeit direkt innerhalb der IDE. Hier bearbeiten Sie Ihren Quelltext und Ressourcen wie Bilder und Dialogboxen. Ferner werden Ihre Programme hier auch übersetzt, im Debugger getestet oder ohne Debugger betrieben. Die Dokumentation steht ebenfalls direkt aus der IDE heraus zur Verfügung – Sie können entweder das Hilfe-Menü bemühen, es ist aber auch möglich direkt im Quelltext-Editor die **[F1]**-Taste zu




DIE AUTOREN-EDITION VON VC++ unterscheidet sich hauptsächlich dadurch von der Vollversion, dass Sie damit keine kommerziellen Programme entwickeln dürfen.

drücken – das Hilfesystem zeigt Informationen zu dem Wort an, auf dem sich der Cursor befindet. Handelt es sich dabei etwa um einen API-Aufruf, wird die

Dokumentation dieser API angezeigt, ist es ein **"C"**-Befehl zeigt VC++ die Dokumentation zu diesem Befehl an. Das Gleiche gilt auch für Fehlermeldungen des Compilers und des Linkers: **[F1]** hilft weiter.

Bei VC++ können Sie die verschiedensten Projekte erstellen: So ist es möglich eine 32bit-Anwendung für die Windows Konsole zu entwickeln, aber natürlich sind auch Programme, die eigene Fenster verwenden, möglich. Ferner können Sie auch Win32-DLLs erzeugen, COM-Objekte programmieren sowie Erweiterungen für bestimmte Pakete (etwa das Developer Studio selbst, aber auch für den IIS) entwickeln. Dabei können Sie bei den einzelnen Projekttypen noch zwischen einer ganzen Reihe von Optionen wählen: Eine Win32-Anwendung kann etwa mit oder ohne MFC, ein ActiveX-Objekt kann mit oder ohne ATL erzeugt werden. Damit sind auch zwei weitere wichtige Komponenten genannt: VC++ kommt nicht nur mit der standardisierten Funktionsbibliothek für C und der standardisierten C++-Bibliothek, sondern auch mit einigen weiteren Bibliotheken – im Wesentlichen sind das die MFC (Microsoft Foundation Classes) und die ATL (Active Template Library). MFC eignet sich besonders fürs Entwickeln von Windows-Anwendungen mit eigenen Fenstern, ATL ist eher für COM und ActiveX-Objekte gedacht: Mischen kann man aber trotzdem.

Wie sie sehen: Alles ist nicht so einfach und obendrein noch mit Fremdworten gespickt – aber die Fremdworte werden Ihnen in Kürze gar nicht mehr so fremd vorkommen, und die verschiedenen verwirrenden Option gehören schnell zum Alltag – alles was man braucht, ist ein wenig Übung, und die werden Sie im Zuge dieses Sonderheftes erhalten.  U R

DIE VC++-EDITOREN UND TOOLS

Wie die Bezeichnung IDE schon sagt, sind bei VC++ die allermeisten benötigten Werkzeuge in einer Oberfläche zusammengefasst. Sie finden dort natürlich einen Quelltexteditor, aber auch Editoren für die unterschiedlichsten anderen Dinge: so zum Beispiel einen speziellen Editor für Bitmaps, Icons und Cursor (ein Malprogramm, wenn man so will), einen Editor für das Bearbeiten von String-Tabellen (sehr praktisch für das Übersetzen eines Programms in andere Sprachen) und auch einen Editor für Dialogboxen.

Zusätzlich zu den Editoren steuern Sie mit VC++ auch den Compiler und den Linker. Alle Optionen für diese Programme – und das können recht viele werden – können direkt

innerhalb der VC++-IDE eingestellt werden. Für den Anfang müssen Sie sich aber nicht mit den verschiedenen Compiler-Optionen beschäftigen, denn VC++ enthält auch Ratgeber, die sich beim Erstellen eines neuen Projekts darum kümmern, dass zumindest recht sinnvolle Optionen von vornherein eingestellt sind.

Auch der Debugger ist in die IDE integriert. Beim Debugger handelt es sich um ein Programm, mit dem Sie Ihr eigenes Programm ausführen können, wobei Sie dabei die Möglichkeit haben, den Inhalt von Variablen während des Programmlaufs einzusehen und zu verändern – Sie können sogar den Fluss des Programms steuern. Der Debugger dient natürlich der Fehlersuche: Als Navi-

gationsmittel dient dabei Ihr eigener Quellcode – der Debugger zeigt Ihnen dazu in einem Editorfenster immer die Position im Quellcode an, an der sich das Programm momentan befindet. Finden Sie dort einen Fehler, können Sie diesen direkt ausbessern und dann entweder das Programm weiter laufen lassen, oder aber selbiges zunächst übersetzen und dann erneut starten.

Der Debugger ist dabei das wichtigste Werkzeug, und Sie werden diesen im Laufe der einzelnen Übungen in diesem Sonderheft oft einsetzen – nur ein Programm, das von vornherein ausreichend debugged wurde, hat eine Chance, auch im "echten Leben" fehlerfrei zu laufen.