

Tauchen Sie in Visual Basic ein



In diesem Kapitel

- ▶ Lernen Sie, wie sich VB in .NET einfügt
- ▶ Schreiben Sie Ihr erstes Visual-Basic-2005-Programm
- ▶ Machen Sie sich die neue Leistungsfähigkeit von VB zunutze

Sind Sie für Visual Basic 2005 bereit? Na, dann würde ich vorschlagen, dass Sie direkt ins kalte Wasser springen und Software schreiben! Und damit Sie bei diesem hohen Tempo nicht die Orientierung verlieren, soll Sie dieses Kapitel mit allem versorgen, was Sie brauchen, um Ihre Zehenspitze in das Wasser des Visual-Basic-Pools zu halten und zu schauen, welchen Platz VB in der großen Visual-Studio-Umgebung einnimmt.

Und dann können Sie sich endlich ins Wasser stürzen und *Hallo Welt* schreiben – Ihre erste VB-2005-Windows-Forms-Anwendung – und das schon auf den ersten paar Seiten! Sie lernen, wie Sie Visual Studio 2005 (das Entwicklungswerkzeug für Ihre VB-Anwendungen) starten, wie Sie ein neues Projekt anlegen und wie Sie ein Formular grafisch zusammenstellen und es mit Code zum Leben erwecken.

In diesem Kapitel dürfen Sie auch einen flüchtigen Blick darauf werfen, was alles Tolles in Visual Basic schlummert. Insbesondere stelle ich Ihnen vor, wie sich VB 2005 in das Microsoft .NET Framework integriert, und Sie bekommen einen Einblick, was das für Sie als Programmierer bedeutet.

Visual Basics Rolle im Framework

Microsoft hat das .NET Framework entworfen, um die Entwicklung von Software für die verschiedenen Windows-Betriebssysteme einfacher zu gestalten. Aber wegen der Unterschiede zwischen VB 6.0 und VB 7.0 (der ersten .NET-Version) empfanden die meisten VB-Programmierer die Entwicklung als sehr viel schwieriger. Zum Beispiel sind in VB 7.0 alle Variablen vom Typ `Object` abgeleitet, was den Programmierer der Möglichkeit beraubte, einfach eine Variable ohne Deklaration zu verwenden.

Aber die Entwicklung von Anwendungen unter .NET muss nicht schwieriger als die Entwicklung mit VB 6.0 sein. Das .NET Framework und Visual Studio 2005 sind mächtige Werkzeuge, und der Trick besteht darin, zu begreifen, wie diese über die integrierte Entwicklungsumgebung (IDE) von Visual Studio zusammenarbeiten.

Eine Schwierigkeit, der Programmierer beim Wechsel zum .NET Framework gegenüber stehen, stellt die teilweise sehr verwirrende Terminologie dar. Ich möchte dieses Problem mit der Terminologie erst gar nicht aufkommen lassen und habe Ihnen zu diesem Zweck die folgende Liste mit potenziell verwirrenden Begriffen aus der .NET-Entwicklung zusammengestellt:

- ✓ **Visual Basic 2005:** Die Programmiersprache, die durchweg in diesem Buch beschrieben wird. Sie können Visual Basic nicht mehr länger als eigenständige Instanz ausführen. Es ist lediglich eine Programmiersprache, die mit dem nächsten Begriff auf unserer Liste, dem .NET Framework, spricht.
- ✓ **.NET Framework:** Die Schicht, die zwischen der Sprache (in diesem Fall Visual Basic) und dem Betriebssystem sitzt. Bei dem Betriebssystem kann es sich um Windows 98, Windows Me, Windows 2000, Windows XP, Windows Server 2003 oder um eine der speziellen Editionen (z.B. die Tablet PC Edition) der Genannten handeln. Die .NET Framework-Schicht dient dazu, sowohl Funktionalität, die auf der Funktionsweise des zugrunde liegenden Betriebssystems basiert, als auch Bibliotheken für andere Zwecke (z.B. mathematische Berechnungen oder den Datenbankzugriff) zur Verfügung zu stellen. Abbildung 1.1 stellt die einzelnen Schichten des .NET Frameworks vor.

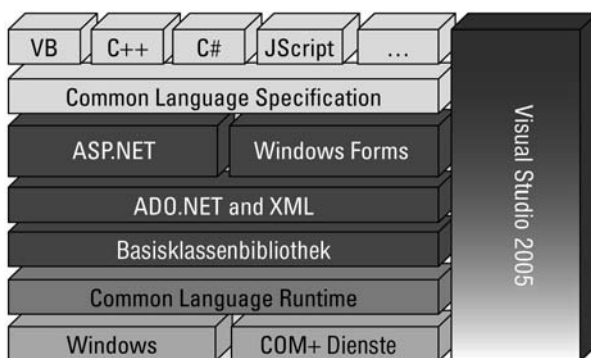


Abbildung 1.1: Die .NET Framework-Hierarchie

- ✓ **Visual Studio 2005:** Die Software, mit der Sie jegliche Art von Anwendung in einer kompatiblen Programmiersprache erstellen können. Visual Studio ersetzt das Programm Visual Basic 6.0, das einst Teil der Visual Studio Suite war (alle Komponenten der Suite wurden mit Version 6.0 gekennzeichnet). Wenn Sie ein neues Programm in der .NET-Umgebung schreiben, starten Sie Visual Studio 2005 und wählen einen Projekttyp und die Sprache aus, in der Sie das Programm schreiben wollen. Beispielweise könnten Sie ja, so wie in alten Zeiten, eine Windows-Anwendung in der Sprache Visual Basic schreiben. Vielleicht möchten Sie aber auch eine Anwendung für ein intelligentes Gerät (oder auch Smart Device genannt) in C# programmieren.
- ✓ **Windows-Forms-Anwendungen:** Der neue Begriff für eine althergebrachte Visual-Basic-Anwendung. Der Begriff bezieht sich auf eine Anwendung, die unter Verwendung des .NET Frameworks geschrieben wurde und eine Windows-Benutzeroberfläche hat.
- ✓ **Web-Forms-Anwendungen:** Der Begriff für eine Anwendung mit einer Webseitenoberfläche, die unter Verwendung des .NET Frameworks geschrieben wurde. Das Erstellen einer Web-Forms-Anwendung funktioniert ähnlich wie das Schreiben einer Windows-Forms-Anwendung.

- ✓ **Web Services:** Klassenbibliotheken, die unter Verwendung eines Standards geschrieben werden, der von denselben Leuten definiert wurde, die auch die Standards für das World Wide Web definiert haben. Web Services werden für die Zusammenarbeit zwischen unterschiedlichen Systemen verwendet.



Das .NET Framework ist das, was Sie vielleicht als Win32-API im alten Windows-DNA-System kennen. Ebenso wie das .NET Framework hat Ihnen die Win32-API ermöglicht, bei der Entwicklung einer Anwendung für eine Windows-Plattform auf die Funktionen des Betriebssystems zuzugreifen. Das .NET Framework enthält außerdem sehr viel zusätzliche Funktionalität, etwa Bibliotheken für mathematische Funktionen oder Grafikprogrammierung, die bei der Entwicklung keine Wünsche offen lassen.

Im Grunde wird alles, was man mit Windows tun kann, durch das .NET Framework offen gelegt. Insbesondere gibt das .NET Framework jedem Objekt und jedem Ereignis, das Windows beeinflussen kann, einen programmatischen Namen. Ein Programmierer kann dann mit diesem Namen auf alles zugreifen, was Code im Betriebssystem besitzt. Wollen Sie dem Drucker mitteilen, dass er zwei Exemplare Ihres Dokuments drucken soll? Mit `My.Computer.Printers.DefaultPrinter.PrinterSettings.Copies = 2` kein Problem. Wollen Sie ein Rechteck in Ihrer Benutzeroberfläche blau ausmalen? `System.Drawing.Brushes.Blue` löst Ihr Problem.

In dieser .NET-Welt dient die Programmiersprache VB nur dazu, mit dem Framework, und damit mit Windows, zu kommunizieren. Alle Programme brauchen eine Menge festgelegter Regeln, mit denen man den Programmfluss (Entscheidungen, Schleifen und Ähnliches) steuern kann. Visual Basic stellt so eine Regelmenge zur Verfügung und das Framework liefert die Objekte und Ereignisse, die mit den Regeln zur Funktionalität des Programms verarbeitet werden.



Wie sich Visual Basic 2005 von Visual Basic 6 unterscheidet

Visual Basic 6 ist ein eigenständiges Programm, während Visual Basic 2005 nur eine Sprache in einem größeren Entwicklungssystem ist. Ursprünglich stammt Visual Basic von der Sprache BASIC ab, die 20 Jahre zuvor als Teil von MS DOS verwendet wurde. 1991 wurde aus BASIC Visual Basic und damit ein Programm zur Erstellung von Windows-Anwendungen mit Benutzeroberfläche. Visual Basic 6 umfasst sehr viel mehr als nur die Sprache – die Software zur Formularerstellung heißt z.B. *Ruby*. In Visual Basic 2005 gibt es einen neuen Formulargenerator und mit ihm eine neue Art, mit Windows zu kommunizieren. Der wahre Grund, warum Sie den Umfang dieses nun größeren Entwicklungssystems – und die Komplexität des VB 2005 umgebenden .NET Frameworks – verstehen sollen, ist, dass es Ihnen das Lesen von Büchern und Dokumentationen zu diesem Thema vereinfacht.

Sagen Sie zu VB 2005 Hallo!

In diesem Abschnitt beginnen Sie mit dem klassischen Hallo-Welt-Programm. Obwohl das nicht die aufregendste Anwendung ist, die Sie je schreiben werden, kann man mit ihr doch gut feststellen, ob Ihre Entwicklungsumgebung richtig eingerichtet ist.

Visual Studio installieren

Damit Sie diesem Beispiel folgen können, müssen Sie zunächst Visual Studio ausführen – die Entwicklungsumgebung, mit der im gesamten Buch Anwendungen in Visual Basic erstellt werden. Aber bevor Sie Visual Studio ausführen können, müssen Sie es installieren!

Visual Studio gibt es in verschiedenen Editionen:

- ✓ **Team System:** Diese Edition, entworfen für einen ganzen Stab an Programmierern in großen Firmen, enthält Entwurfswerkzeuge für große Anwendungssysteme, wie testgetriebene Entwicklung und den Team Foundation Server.
- ✓ **Professional Edition:** Für die Erstellung professioneller, vielschichtiger Anwendungen geeignet. Diese Edition ist mehr für den einzelnen Entwickler oder für die Entwicklung in kleinen bis mittelgroßen Teams gedacht. Die Professional Edition verwende ich in diesem Buch.
- ✓ **Standard Edition:** Diese Version ist perfekt für 80 Prozent aller Anwendungen geeignet und wurde für die Entwicklung kleiner Einzel- oder Client-Server-Anwendungen entworfen. Aber wenn Sie vorhaben, große, geschäftskritische Systeme zu erstellen, auf die vielleicht mal viele Benutzer zugreifen werden, verwenden Sie lieber die Professional Edition.
- ✓ **Express Edition:** Für Studenten und Hobbyprogrammierer entworfen. Bei dieser Version fehlen einige Features, die in den anderen Versionen enthalten sind, so z.B. die Unterstützung mobiler Geräte oder der Klassen-Designer.

Die Professional Edition und Visual Studio Team Suite können Sie auch als Testversionen von der Microsoft-Webseite herunterladen, ebenso wie die Express-Versionen, die aber völlig kostenlos sind und unbegrenzt verwendet werden können.



Sollten Sie keinen Zugriff auf die MSDN-Bibliothek (Microsofts praktisches, technisches Archiv) haben, empfehle ich Ihnen sehr, sie sich irgendwo zu besorgen. Dann stehen Ihnen nämlich jede Menge Beispielcode, Dokumentationen und anderes Referenzmaterial über die einzelnen Visual-Studio-Editionen, über Betriebssysteme und Server-Software zur Verfügung. Sie können sich über die Bibliothek unter <http://www.microsoft.com/germany/msdn> informieren und Abonnements bei verschiedenen Händlern, Ihren Lieblings-Software-Händler eingeschlossen, kaufen.

Die Installation von Visual Studio kann ein schwieriges Unterfangen sein, also empfehle ich Ihnen, sich bei Ihrem ersten Mal zunächst an die Standardeinstellungen zu halten. Zudem braucht der Installationsprozess eine Weile. Auch wenn Sie die DVD verwenden, können Sie

gut und gerne von zwei Stunden Installationszeit ausgehen. Nach der Installation von Visual Studio können Sie es starten, indem Sie in Windows START|ALLE PROGRAMME|MICROSOFT VISUAL STUDIO 2005|MICROSOFT VISUAL STUDIO 2005 auswählen. Die Umgebung wird geladen und Sie können dann mit der Auswahl DATEI|NEU|PROJEKT aus dem Hauptmenü ein neues Projekt erstellen. Als Nächstes müssen Sie sich für einen Projekttyp und eine Programmiersprache entscheiden. Dies wird im nächsten Abschnitt beschrieben.

Ein Windows-Forms-Projekt anlegen

Nachdem Sie DATEI|NEUES PROJEKT aus dem Visual-Studio-Hauptmenü ausgewählt haben, erscheint das Dialogfenster NEUES PROJEKT, das Sie in Abbildung 1.2 sehen. Im Abschnitt PROJEKTTYPEN finden Sie eine Ordnerstruktur, die alle installierten Sprachen und die dafür verfügbaren Projekttypen auflistet. Ich finde, wir sollten mit einer guten, alten Windows-Anwendung beginnen – die Antwort von Visual Basic 2005 auf die traditionelle (und Ihnen möglicherweise bekannte) VB-6.0-Anwendung.

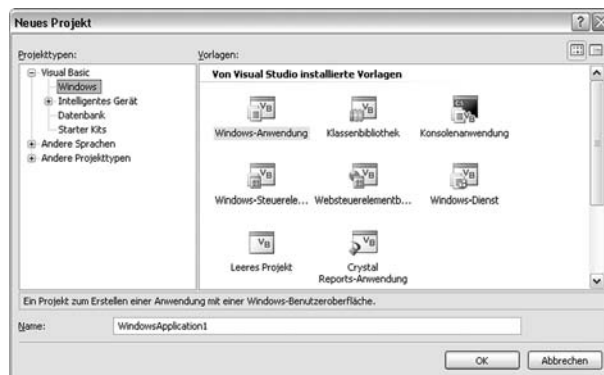


Abbildung 1.2: Das Dialogfenster NEUES PROJEKT

Folgen Sie den nächsten Schritten, um Ihre (vielleicht erste) Hallo-Welt-Anwendung unter VB 2005 zu programmieren:

1. Wählen Sie aus dem Abschnitt VORLAGEN in dem Dialogfenster NEUES PROJEKT den Projekttyp aus.

Wählen Sie für dieses Beispiel WINDOWS-ANWENDUNG aus. Vergewissern Sie sich, dass im Abschnitt PROJEKTTYPEN VISUAL BASIC als Sprache ausgewählt ist. Sind noch andere Sprachen installiert, finden Sie diese auch im Abschnitt PROJEKTTYPEN.

2. Geben Sie den Namen, den Sie Ihrem Projekt geben wollen, statt des Standardnamens in das Textfeld NAME ein.

In diesem Beispiel gebe ich Hallo Welt in das Textfeld ein.

3. Klicken Sie auf OK.

Visual Studio lädt das Standardformular `Form1` und stellt es Ihnen in der Entwurfsansicht dar. Das Standardformular enthält von Anfang an schon einen Arbeitsbereich, eine Titelleiste und die bekannten Windows-Elemente, wie die Schaltflächen zum Schließen und Maximieren bzw. Minimieren des Fensters. Die meisten Änderungen an Ihrem Formular werden Sie in dieser grafischen Ansicht vornehmen.

4. Klicken Sie das Wort **TOOLBOX** auf der linken Seite Ihres Bildschirms an.

Die Toolbox mit den installierten Formular-Steuerelementen erscheint, wie in Abbildung 1.3 gezeigt.

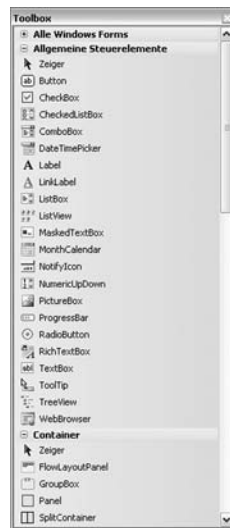


Abbildung 1.3: Auswahl des Schaltflächensteuerelements aus der Toolbox

5. Doppelklicken Sie auf das Steuerelement **BUTTON**.

Visual Studio platziert in der Entwurfsansicht eine Schaltfläche auf dem Standardformular.

6. Klicken Sie im Formular `Form1` auf das **Button**-Steuerelement und positionieren Sie es neu, indem Sie es an die gewünschte Stelle ziehen.

Abbildung 1.4 zeigt das Formular, nachdem die Schaltfläche in die Mitte des Fensters `Form1` gezogen wurde.

Diese Schritte legen den Grundstein für die Windows-Forms-Anwendung, die durch `Form1` in der Entwurfsansicht repräsentiert wird. Aber damit Sie sehen, wo Visual Basic ins Spiel kommt, müssen Sie zunächst den Quelltext finden, der hinter dem Formular steckt. Visual Studio bietet Ihnen (Überraschung!) eine Codeansicht an, sobald Sie an dem Punkt angelangt sind, Funktionalität zu Ihrem Formular hinzuzufügen.

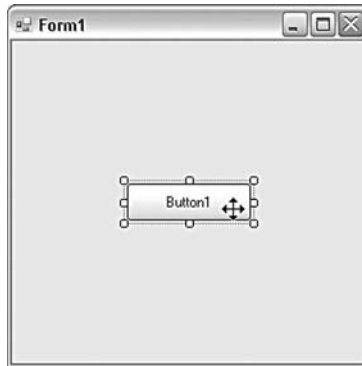


Abbildung 1.4: Verschieben der Schaltfläche über das Formular

Mit VB-Code Funktionalität zu einem Formular hinzufügen

Um etwas Funktionalität zu Ihrem bis hierher erstellten Windows-Formular hinzuzufügen, folgen Sie bitte den nächsten Schritten:

1. Doppelklicken Sie auf die Schaltfläche in Form1, um zur Codeansicht zu gelangen.

Im Fenster der Codeansicht sehen Sie den grundlegenden Code für das Klick-Ereignis der Schaltfläche, der wie folgt aussieht:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
    End Sub
End Class
```

Dieser Programmausschnitt umgibt den Quelltext, der ausgeführt wird, wenn man auf die Schaltfläche klickt. Die schwierigste Aufgabe daran, nämlich den Prozedurkopf richtig zu schreiben, erledigt Visual Studio für Sie!

2. Geben Sie im Fenster der Codeansicht eine Zeile Quellcode ein, um den Text, der auf dem Button-Steuerelement erscheint, in Hallo Welt zu ändern.

Im Detail: Geben Sie in der Zeile vor der End Sub-Zeile den folgenden Programmtext ein:

```
Button1.Text = "Hallo Welt"
```

Der Code für Ihre Schaltfläche sieht nun so aus:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
    Button1.Text = "Hallo Welt"
    End Sub
End Class
```

```
Button1.Text = "Hallo Welt"  
  
End Sub  
End Class
```

Ausführen und Verwenden Ihrer Windows-Anwendung

Na, war das nicht ein Wahnsinnserlebnis? Programmieren mit Visual Basic ist so einfach, dass Sie schon hier, in Kapitel 1, eine Windows-Forms-Anwendung schreiben können. Aber was können Sie nun damit anfangen? Probieren Sie mal das Folgende:

- ✓ **Führen Sie die Windows-Forms-Anwendung innerhalb der Visual-Studio-Umgebung aus.** Drücken Sie **[F5]** auf Ihrer Tastatur und Visual Studio öffnet Ihr aktives Projekt als Windows-Programm. Es erscheint als Fenster und, wie üblich, in der Taskleiste. Klicken Sie auf die Schaltfläche auf Ihrem Formular und schon ändert sich der Text des Knopfes in »Hallo Welt« (oder welchen Text Sie auch immer im Code angegeben haben). Der Wahnsinn! Ihre Anwendung sollte ungefähr wie in Abbildung 1.5 dargestellt aussehen.



Abbildung 1.5: Ihre Hallo-Welt-Anwendung

- ✓ **Führen Sie die Windows-Forms-Anwendung außerhalb der Visual-Studio-Umgebung aus.** Wenn Sie sich immer noch im Unterbrechungsmodus befinden, müssen Sie Ihr Programm erst anhalten, indem Sie auf die Schaltfläche **DEBUGGEN BEENDEN** auf der Werkzeugleiste klicken oder einfach das Fenster Ihrer Anwendung schließen. Dann können Sie Ihr Projekt speichern und fortfahren.

Der einfachste Weg, eine Anwendung außerhalb von Visual Studio auszuführen, ist der folgende:

1. Wählen Sie DATEI|ALLE SPEICHERN aus dem Visual-Studio-Hauptmenü aus.

Das Dialogfenster **PROJEKT SPEICHERN** wird geöffnet und Visual Studio fordert Sie auf, einen Speicherort für Ihr Projekt auszuwählen (siehe Abbildung 1.6). Übernehmen Sie in diesem

Fall den Standardordner. Derselbe Name wird auch für die Projektmappe übernommen, aber Sie können diesen auch frei wählen.

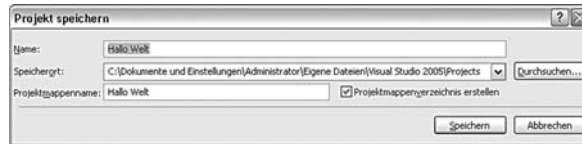


Abbildung 1.6: Das Dialogfenster PROJEKT SPEICHERN

2. Wählen Sie ERSTELLEN|<PROJEKTNAME> ERSTELLEN aus dem Hauptmenü.

In diesem Beispiel wählen Sie ERSTELLEN|HALLO WELT ERSTELLEN aus, Visual Studio kompiliert Ihre Anwendung dann zu einem ausführbaren Windows-Programm (mit der Dateierdung .exe) und speichert es in dem Standardordner des Projektes.

3. Öffnen Sie den Standardordner mit Ihrer neuen Windows-Anwendung.

Der Pfad zu meiner Anwendung lautet C:\Dokumente und Einstellungen\sempf\Eigene Dateien\Visual Studio 2005\Projects\Hallo Welt\Hallo Welt\bin\Release.

4. Doppelklicken Sie auf den Dateinamen des kompilierten Programms, um es auszuführen.

Sie finden im Standardordner jede Menge Dateien, aber Hallo Welt.exe ist die Datei, die Sie suchen.



Es gibt noch eine schwierigere Methode, Ihre VB-Programme außerhalb der Visual-Studio-Umgebung auszuführen. Sie können ein Setup-Projekt verwenden, was aber über den Rahmen dieses Buches hinausgeht. Suchen Sie in der MSDN-Bibliothek nach dem Begriff »Setup-Projekt«, wenn Sie mehr über eigene Setups und wie diese Sie bei der Verteilung Ihrer Anwendung an die Benutzer unterstützen können, herausfinden wollen.

Mehr Leistungsfähigkeit in Visual Studio entdecken

Nun haben Sie schon die Entwicklungsumgebung für Windows-Forms-Anwendungen und ein wenig vom neuen Visual-Basic-2005-Code kennen gelernt. Wenn Sie mit VB 6.0 vertraut sind, werden Ihnen das Formular und der Quelltext recht bekannt vorkommen. Tatsächlich funktionieren die Hauptentwicklungswerkzeuge für Windows-Anwendungen für jede Programmiersprache ähnlich. Aber wenn Sie einen Blick über das Windows-Formular und die Programmstruktur hinaus werfen, eröffnen sich Ihnen noch viel mehr Möglichkeiten. Visual Studio kann nämlich noch etwas anderes aus Ihrem VB-Code machen als nur Windows-Anwendungen. Die folgenden Abschnitte verschaffen Ihnen einen Überblick über die leistungsstarken Möglichkeiten in der Entwicklung mit Visual Studio.

Visual Studio kann nicht nur Windows!

Die erste offensichtliche Neuerung, die Visual Studio als Entwicklungswerkzeug von anderen abhebt, ist die folgende: Sie können Visual Studio dazu verwenden, Programme zu schreiben, die sowohl im Internet als auch auf einem Computer unter Windows laufen. Wenn Sie auf das Menü DATEI klicken, um ein neues Projekt zu erstellen, beachten Sie bitte einmal die zweite Option in dem Menü. Wie Sie in Abbildung 1.7 sehen können, ist die zweite Projektoption eine neue Website.

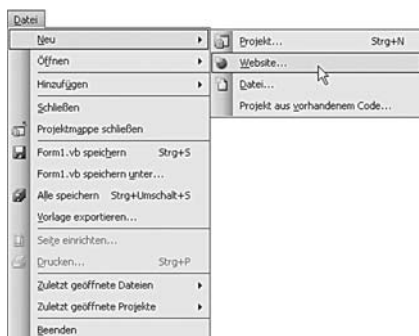


Abbildung 1.7: Das Menü DATEI in Visual Studio

Wählen Sie diese Option aus, um eine neue Webanwendung zu erstellen. Eine Webanwendung umfasst eine ganze Reihe von Technologien – das .NET Framework, ASP.NET, Visual Basic und HTML –, von denen jede einen wichtigen Teil dazu beiträgt, dass eine Anwendung online ausgeführt werden kann.

Visual Basic wird mobil

Mobile Geräte halten Einzug in Visual Basic 2005. Mehrere Projekttypen, die für diese Geräte verwendet werden können, sind direkt in Visual Studio integriert. Dabei kann man zwischen den Plattformen Windows CE 5.0, Pocket PC 2003 und SmartPhone 2003 wählen.

Ich gebe in diesem Buch keine Beispiele zu diesen speziellen Projekttypen an, da Sie eine Anwendung für ein mobiles Gerät genauso erstellen können wie eine Windows-Forms-Anwendung (z.B. das zuvor beschriebene Hallo-Welt-Programm). Wenn Sie sich in diesem Buch mit der Sprache Visual Basic vertraut machen, können Sie im Prinzip auch Anwendungen für Pocket PCs schreiben. Anwendungen für mobile Geräte erfordern allerdings ein paar besondere Programmiermethoden und daher sollten Sie sich ein paar Geräte-spezifische Informationen besorgen, wenn Sie mit diesen Projekttypen arbeiten.



Es ist nun auch einfacher, mit Visual Basic 2005 Programme zu schreiben, die zusammen mit anderer Software verwendet werden. Sie können Add-Ins für Microsoft-Office-Anwendungen schreiben, einschließlich Vorlagen mit ausführbarem VB-Code. Diese Programme verwenden nicht das VBA, wie Sie es vielleicht schon

mal gesehen haben. Ein vollständig neuer Teil von Office 2003 erlaubt Ihnen, Vorlagen mit besonderer, integrierter Funktionalität zu schreiben. Ich habe beispielsweise eine Word-Vorlage erstellt, die einen Berichtsprozess automatisiert. Sie fordert vom Benutzer eine Berichtsnummer an, prüft die Nummer gegen eine Datenbank mit allen archivierten Berichten und füllt einen Teil des gerade bearbeiteten Dokuments mit den entsprechenden Informationen aus der Datenbank aus.

VB - Entwicklung aus einer Hand

Im Allgemeinen sind Visual Studio und das .NET Framework so entworfen, dass mit ihnen jegliche Art der Entwicklung auf Windows-Rechnern möglich ist. Und in dieser Version ist nun auch VB in der Lage, alle gewünschten Funktionen bereitzustellen. Die Sprache kann nun auf alle Teile des .NET Frameworks zugreifen, so wie die anderen Sprachen auch, ohne auf die kryptischen Funktionsaufrufe zurückzugreifen, die in früheren VB-Versionen nötig waren.



Die neuen Eigenschaften, die in diesem Buch beschrieben werden, umfassen:

- ✓ **Sicherheit:** Verschlüsselung, Gültigkeitsprüfung und Berechtigungen. Der Schutz Ihres Programmcodes durch komplexe Verschlüsselung ist nun, neben anderer Funktionalität, in das Framework integriert.
- ✓ **Daten:** Informationssammlungen, Datenbankzugriffe und XML. Es gibt auch neue datengebundene Steuerelemente für Ihre Formulare!
- ✓ **E/A:** Der Programmzugriff auf Dateien, Festplatten und Ordner ist integriert, so dass viel weniger Code geschrieben werden muss.
- ✓ **Netzwerkzugriff und Internet:** VB kennt sich auch im Internet aus. Web-, FTP- und E-Mail-Funktionen sind alle an einer Stelle im Framework zusammengefasst.
- ✓ **Grafik:** Umfassende Bildschirmgrafiken für das Web und Windows, sogar in 3D.
- ✓ **Das My-Objekt:** Sie kommen mit dieser leicht zu verwendenden Abkürzungssammlung einfacher zu den schwer erreichbaren Stellen des Frameworks. Sie können damit schnell bestimmte Funktionen ausführen, etwa den gerade am Computer angemeldeten Benutzer bestimmen.

Diese Auflistung zeigt, dass Visual Basic etwas erwachsener geworden ist. Sollten Sie VB 6 nicht kennen, brauchen Sie sich keine Sorgen zu machen! Es ist besser, sich mit dieser Version (Visual Basic 2005) gut anzufreunden. Wenn Sie VB 6 kennen, dann willkommen zu Hause! Sie sind endlich da angekommen, wo Sie immer sein wollten!

