

In Visual Basic eintauchen



In diesem Kapitel

- ▶ Wie sich Visual Basic in .NET einfügt
- ▶ Ihr erstes Visual Basic 2008-Programm schreiben
- ▶ Die neue Leistungsfähigkeit von Visual Basic nutzen

Sind Sie für Visual Basic 2008 bereit? Ja? Na, dann würde ich vorschlagen, dass Sie direkt ins kalte Wasser springen und Software entwickeln! 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 sehen, welchen Platz Visual Basic in der großen Visual Studio-Umgebung einnimmt.

Und dann können Sie sich endlich richtig ins Wasser stürzen und *Hallo Welt* schreiben – Ihre erste VB 2008 Windows Forms-Anwendung – und das schon auf den ersten paar Seiten! Sie lernen, wie Sie Visual Studio 2008 (das Entwicklungswerkzeug für Ihre VB -Anwendungen) starten, ein neues Projekt anlegen, ein Formular grafisch zusammenstellen und es mit Code zum Leben erwecken.

In diesem Kapitel dürfen Sie auch schon mal einen flüchtigen Blick darauf werfen, welche tollen Sachen in Visual Basic schlummern. Insbesondere stelle ich Ihnen vor, wie VB 2008 und das Microsoft .NET Framework zusammenhängen, 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. Da es aber zwischen VB 6.0 und VB 7.0 (der ersten .NET-Version) einige Unterschiede gibt, empfanden die meisten VB-Programmierer die Entwicklung damit 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 sein als die Entwicklung mit VB 6.0. Das .NET Framework und Visual Studio 2008 sind leistungsstarke Werkzeuge, und der Trick besteht darin, zu begreifen, wie diese über die integrierte Entwicklungsumgebung (IDE) Visual Studio zusammenarbeiten.

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

- ✓ **Visual Basic 2008:** 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 sich zwischen der Sprache (in diesem Fall Visual Basic) und dem Betriebssystem befindet. Bei dem Betriebssystem kann es sich um Windows 98, Windows Me, Windows 2000 (.NET Framework 2.0), Windows XP, Windows Vista, Windows Server 2003, Windows Server 2008 oder um eine der speziellen Editionen (zum Beispiel die Tablet PC Edition) der genannten Betriebssysteme handeln. Die .NET Framework-Schicht dient dazu, sowohl Funktionalität, die auf der Funktionsweise des zugrunde liegenden Betriebssystems basiert, als auch Bibliotheken für andere Zwecke (zum Beispiel mathematische Berechnungen oder Datenbankzugriff) zur Verfügung zu stellen. Abbildung 1.1 stellt die einzelnen Schichten des .NET Frameworks vor.

.NET Framework 3.5



.NET Framework 3.0 + SP1

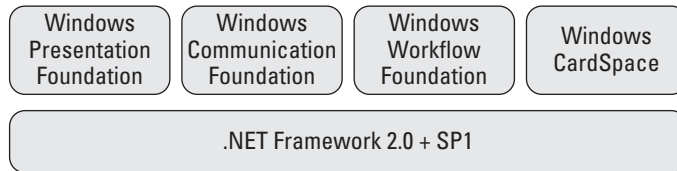


Abbildung 1.1: Die .NET Framework-Hierarchie

- ✓ **Visual Studio 2008:** 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 erstellen wollen, starten Sie Visual Studio 2008 und wählen einen Projekttyp und die Sprache aus, in der Sie das Programm schreiben möchten. Beispielweise könnten Sie ja, wie in alten Zeiten, eine Windows-Anwendung in der Sprache Visual Basic schreiben. Vielleicht möchten Sie aber auch eine Anwendung für ein intelligentes Gerät (auch Smart Device genannt) in C# programmieren. Sie können auch verschiedene Sprachen zusammen verwenden, indem Sie zum Beispiel die Formulare in VB schreiben und die Klassen in C#. In diesem Buch verwenden wir für alle Aufgaben VB – denn dies ist ja schließlich ein Buch über VB!
- ✓ **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 wird und eine Windows-Benutzeroberfläche hat.

- ✓ **Web Forms-Anwendungen:** Der Begriff für eine Anwendung mit einer Webseitenoberfläche, die unter Verwendung des .NET Frameworks geschrieben wird. 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 festgelegt haben. Web Services werden für die Zusammenarbeit zwischen unterschiedlichen Systemen verwendet.



Das .NET Framework ist das, was Sie vielleicht als Win32-Programmierschnittstelle vom 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, wie etwa Bibliotheken für mathematische Berechnungen oder den Zugriff auf Daten, die bei der Entwicklung keine Wünsche offen lassen.

Im Grunde wird alles, was Windows an Funktionalität bietet, durch das .NET Framework offen gelegt. Insbesondere gibt das .NET Framework jedem Objekt und jedem Ereignis, das Windows steuern kann, einen programmatischen Namen. Ein Programmierer kann mit diesen Namen auf alles zugreifen, was zum Betriebssystem gehört. Sie wollen wissen, wie viel physikalischer Speicher Ihrem Computer noch zur Verfügung steht? Mit `My.Computer.Info.AvailablePhysicalMemory` kein Problem. Sie wollen ein Rechteck in Ihrer Benutzeroberfläche blau ausmalen? `System.Drawing.Brushes.Blue` gibt Ihnen das richtige Werkzeug dazu an die Hand.

In dieser .NET-Welt dient die Programmiersprache 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 eine solche Regelmenge zur Verfügung und das Framework liefert die Objekte und Ereignisse, die mit Hilfe der Regeln erzeugt, verwendet und bearbeitet werden können.

Sagen Sie VB 2008 Hallo!

In den folgenden Abschnitten beginnen Sie mit dem klassischen HalloWelt-Programm. Obwohl das nicht die aufregendste Anwendung sein wird, die Sie je schreiben werden, kann man mit ihr trotzdem gut prüfen, ob Ihre Entwicklungsumgebung optimal 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!



Wie sich VB 2008 von VB 6 unterscheidet

Visual Basic 6 ist ein eigenständiges Programm, während Visual Basic 2008 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. Visual Basic 6 umfasst sehr viel mehr als nur die Sprache – die Software zur Formularerstellung heißt zum Beispiel *Ruby*.

Visual Basic hat seit VB 6 mehrere Revisionen durchlaufen. VB 2002 (auch als VB 7 bekannt), VB 2003 (VB 7.1) und VB 2005 (VB 8) sind lediglich Überarbeitungen der Sprache, die das .NET Framework verwenden. Mit VB 2002 wurde die Erstellung von Windows-Anwendungen völlig neu überdacht und mit VB 2005 kehrten einige Eigenschaften zurück, auf die sich VB 6-Programmierer verlassen haben – zum Beispiel die einfache Verwendung.

In Visual Basic 2008 gibt es eine neue Art und Weise, Benutzeranwendungen zu schreiben und damit auch einen neuen Weg, mit Windows zu kommunizieren. Der wahre Grund, warum Sie den Umfang dieses nun größeren Entwicklungssystems – und die Komplexität des VB 2008 umgebenden .NET Frameworks – verstehen sollen, besteht darin, dass es Ihnen das Lesen von Büchern und Artikeln zu diesem Thema vereinfacht.

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 besser die Professional Edition.
- ✓ **Express Edition:** Für Studenten und Hobbyprogrammierer entworfen. Bei dieser Version fehlen einige Features und Projekttypen, die in den anderen Versionen enthalten sind, so zum Beispiel die Unterstützung mobiler Geräte oder der Klassen-Designer.

Die Professional Edition und Visual Studio Team Suite können Sie auch als Testversionen von der Microsoft-Website 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, sich darum zu kümmern. Dann stehen Ihnen nämlich jede Menge Beispielcode, Dokumentationen und anderes Referenzmaterial zu den einzelnen Visual Studio-Editionen, zu Betriebssystemen und Serversoftware zur Verfügung. Sie können sich über die Bibliothek unter <http://www.microsoft.com/germany/msdn> informieren und Abonnements bei verschiedenen Händlern erwerben.

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 2008\MICROSOFT VISUAL STUDIO 2008 auswählen. Die Umgebung wird geladen und Sie können dann über DATEI\NEUES PROJEKT im Hauptmenü ein neues Projekt erstellen. Als Nächstes müssen Sie sich für einen Projekttyp und eine Programmiersprache entscheiden. Das beschreibe ich im nächsten Abschnitt.

Ein Windows Forms-Projekt erstellen

Nachdem Sie DATEI\NEUES PROJEKT im Visual Studio-Hauptmenü gewählt haben, erscheint das Dialogfeld NEUES PROJEKT, das Sie in Abbildung 1.2 sehen. Im Bereich 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 2008 auf die traditionelle (und Ihnen möglicherweise bekannte) VB 6.0-Anwendung.

Folgen Sie diesen Schritten, um mit der Programmierung Ihrer HalloWelt-Anwendung zu starten:

1. Wählen Sie im Dialogfeld NEUES PROJEKT im Bereich VORLAGEN den Projekttyp aus.

Wählen Sie für dieses Beispiel WINDOWS FORMS-ANWENDUNG aus. Vergewissern Sie sich, dass im Bereich PROJEKTTYPEN als Sprache VISUAL BASIC ausgewählt ist. Sind noch andere Sprachen installiert, finden Sie diese und die entsprechenden Projekttypen ebenfalls im Bereich 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 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 beziehungsweise Minimieren des Fensters. Die meisten Änderungen an Ihrem Formular werden Sie in dieser grafischen Ansicht vornehmen.

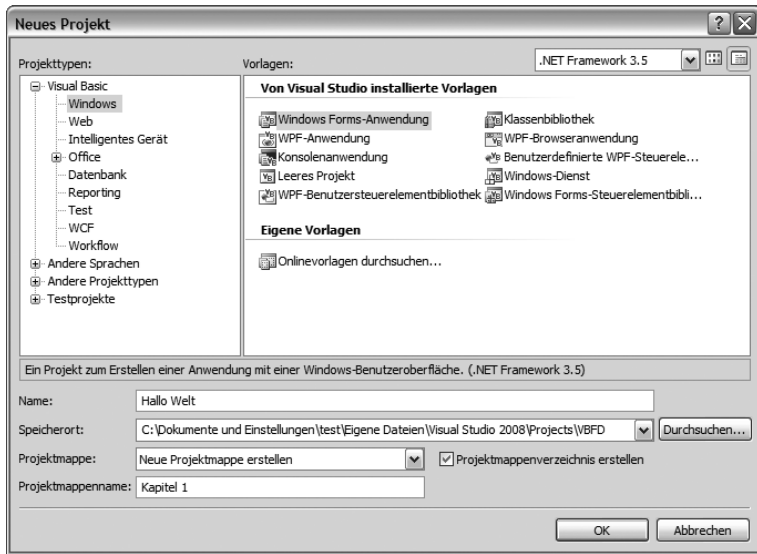


Abbildung 1.2: Das Dialogfeld NEUES PROJEKT

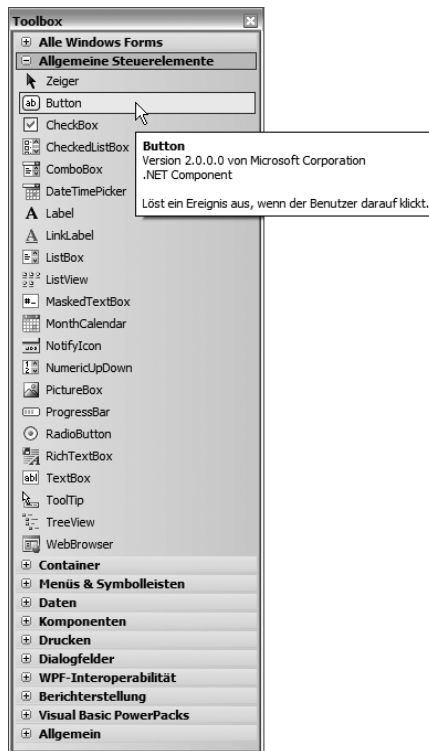


Abbildung 1.3: Auswahl des Schaltflächen-Steuerelements in der Toolbox

4. Klicken Sie auf das Wort TOOLBOX auf der linken Seite Ihres Bildschirms und erweitern Sie den Knoten ALLGEMEINE STEUERELEMENTE.

Die Toolbox mit den Windows Forms-Steuerelementen erscheint, wie in Abbildung 1.3 gezeigt.

5. Doppelklicken Sie auf das Steuerelement BUTTON.

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

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 Form1-Fensters 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 soweit sind, Funktionalität zu Ihrem Formular hinzuzufügen.

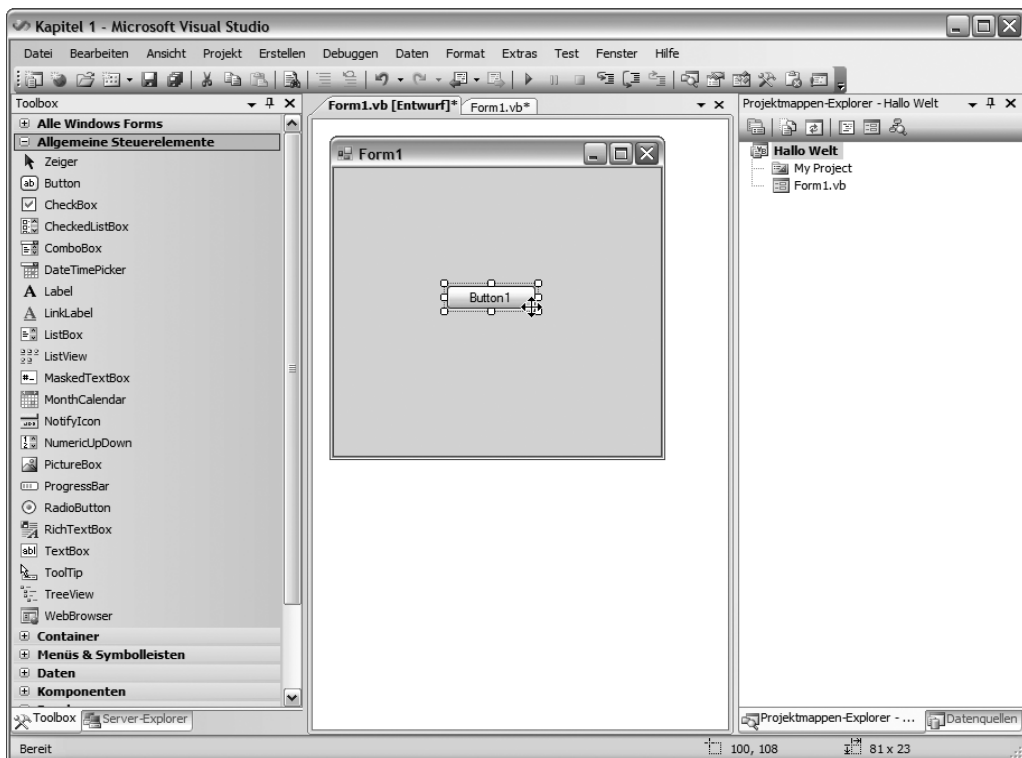


Abbildung 1.4: Verschieben der Schaltfläche auf dem Formular

Mit VB-Code Funktionalität zu einem Formular hinzufügen

Um etwas Funktionalität zu Ihrem bislang erstellten Windows-Formular hinzuzufügen, folgen Sie diesen 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
```

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 einmal Folgendes:

- ✓ 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, als Anwendungssymbol in der Taskleiste. Klicken Sie auf die Schaltfläche auf Ihrem Formular und schon ändert sich der Text der Schaltfläche in *Hallo Welt* (oder welchen Text Sie auch immer im Code angegeben haben). Der Wahnsinn! Ihre Anwendung sollte etwa wie in Abbildung 1.5 aussehen.

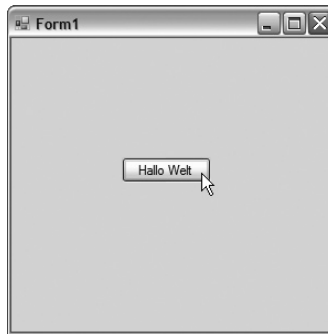


Abbildung 1.5: Ihre HalloWelt-Anwendung

- ✓ **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 **DEBUGGING BEENDEN** in der Symbolleiste 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 folgender:

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

Visual Studio speichert Ihr Projekt an dem Speicherort, den Sie beim Anlegen des Projekts im Dialogfeld **NEUES PROJEKT** angegeben haben.

2. Wählen Sie ERSTELLEN|<PROJEKTNAME> ERSTELLEN im Hauptmenü.

In diesem Beispiel wählen Sie **ERSTELLEN|HALLO WELT ERSTELLEN**. Visual Studio kompiliert Ihre Anwendung zu einem ausführbaren Windows-Programm (mit der Dateierweiterung `.exe`) und speichert es in dem Standardordner des Projekts.

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

Der Pfad zu meiner Anwendung lautet `C:\Dokumente und Einstellungen\sempf\Eigene Dateien\Visual Studio 2008\Projects\VBFD\Kapitel 1\Hallo Welt\bin\Debug`.



Sollte Ihr Projekt mit der Release-Konfiguration kompiliert werden (Für dieses Buch nicht empfohlen; Sie können die Konfigurationseinstellungen in den Projekteigenschaften auf der Seite **KOMPILIEREN** prüfen), finden Sie die ausführbare Datei in `C:\Dokumente und Einstellungen\sempf\Eigene Dateien\Visual Studio 2008\Projects\VBFD\Kapitel 1\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 komplexere Methode, Ihre VB-Programme außerhalb der Visual Studio-Umgebung auszuführen und Benutzern zur Verfügung zu stellen. Sie können ein Setup-Projekt verwenden. Setup-Projekte sind sehr praktisch, gehen aber über den Rahmen dieses Buchs hinaus. Suchen Sie in der MSDN-Bibliothek nach dem Begriff »Setup-Projekt«, wenn Sie mehr über eigene Setups herausfinden wollen und wie diese Sie bei der Verteilung Ihrer Anwendung an die Benutzer unterstützen können.

Mehr Leistungsfähigkeit in Visual Studio entdecken

Nun haben Sie schon die Entwicklungsumgebung für Windows Forms-Anwendungen und ein wenig vom neuen Visual Basic 2008-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 der Entwicklung mit Visual Studio.

Visual Studio kann nicht nur Windows!

Die erste offensichtliche Neuerung, die Visual Studio als Entwicklungswerkzeug von anderen abhebt, ist 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 einmal die zweite Option in dem Menü. Wie Sie in Abbildung 1.6 sehen können, ist die zweite Projektoption eine neue Website.

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 einzelne einen wichtigen Teil dazu beiträgt, dass eine Anwendung online ausgeführt werden kann.

Visual Basic wird mobil

Mobile Geräte traten in Visual Basic 2005 erstmals auf den Plan und glänzen ebenfalls in Visual Studio 2008. Wenn Sie die Entwicklung mobiler Geräte verfolgt haben, ist Ihnen vielleicht aufgefallen, dass das Mobile Team in den letzten Jahren sehr viele Releases herausgebracht hat. Diese wurden alle in Visual Studio 2008 berücksichtigt. Pocket PC 2003, Mobile 5.0 und Mobile 6.0 tauchen alle in Visual Studio 2008 auf und Sie finden für jede Version in VB 2008 einen entsprechenden Projekttyp.

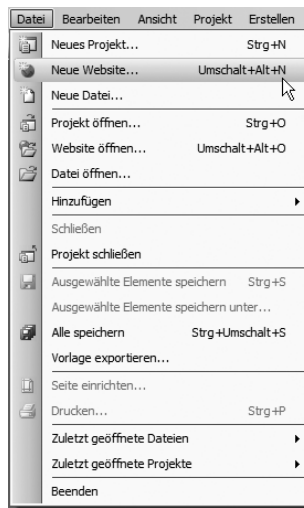


Abbildung 1.6: Das Menü DATEI in Visual Studio

Ich zeige in diesem Buch keine Beispiele zu diesen speziellen Projekttypen, da Sie eine Anwendung für ein mobiles Gerät genauso erstellen können wie eine Windows Forms-Anwendung (zum Beispiel das zuvor beschriebene HalloWelt-Programm). Wenn Sie sich in diesem Buch mit der Sprache Visual Basic vertraut machen, können Sie auch Anwendungen für Pocket PCs schreiben. Anwendungen für mobile Geräte erfordern allerdings ein paar besondere Programmierkniffe; daher sollten Sie sich ein paar gerätespezifische Informationen besorgen, wenn Sie mit diesen Projekttypen arbeiten.



Es ist auch einfach, mit Visual Basic 2008 Programme zu schreiben, die zusammen mit anderer Software verwendet werden. Sie können Add-Ins für Microsoft Office-Anwendungen schreiben, einschließlich Word- und Excel-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 2007 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. Sie können sogar die Multifunktionsleiste anpassen und Add-Ins ganz einfach erstellen und verteilen.

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 Visual Basic 2008 nun auch 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 Neuerungen von Visual Basic 2008, die in diesem Buch beschrieben werden, umfassen:

- ✓ **Die Windows Presentation Foundation:** Microsoft hat die Möglichkeiten zur Darstellung von Benutzeroberflächen und Multimedia-Inhalten noch einmal verbessert und umfangreicher gestaltet.
- ✓ **Die Windows Communication Foundation:** Die Möglichkeiten der Kommunikation verteilter Anwendungen untereinander sind durch dieses Feature noch umfangreicher geworden.
- ✓ **Language Integrated Query (LINQ):** LINQ integriert Abfragesprachenkonstrukte direkt in Ihren Programmcode und führt neue Abfragemechanismen für Objektauflistungen ein.
- ✓ **Microsoft Silverlight:** Microsoft Silverlight eröffnet neue und schicke Möglichkeiten zur Programmierung von Webanwendungen.