

HANSER



Vorwort

Jan Tittel

Office 2010 Programmierung mit VSTO und .NET 4.0

Word, Excel und Outlook erweitern und anpassen

Herausgegeben von Holger Schwichtenberg

ISBN: 978-3-446-42411-1

Weitere Informationen oder Bestellungen unter

<http://www.hanser.de/978-3-446-42411-1>

sowie im Buchhandel.

1

Einleitung

Vielen Dank, dass Sie sich für dieses Buch zur Office 2010-Programmierung mit VSTO und .NET 4.0 entschieden haben. Bevor es in den folgenden Kapiteln zur Sache geht, erfahren Sie in dieser Einleitung, für wen dieses Buch gedacht ist, was in diesem Buch behandelt wird, wie es aufgebaut ist und wie die Anforderungen an eine Arbeitsumgebung für die Entwicklung mit VSTO aussehen.

Meine Motivation, dieses Buch zu schreiben, ist aus meinen Erfahrungen mit den Visual Studio Tools für Office (VSTO) entstanden sowie aus den Vorteilen, die VSTO mit sich bringt. So werden immer noch viele auf Office basierende Lösungen ohne VSTO umgesetzt. Dabei lassen sich viele Aufgaben mit VSTO wesentlich eleganter als mit anderen Möglichkeiten lösen. Weitere Vorteile von VSTO sind, dass sich die Anwender innerhalb ihrer gewohnten Arbeitsumgebung (Microsoft Office) bewegen können, um selbst an solchen Aufgaben zu arbeiten, die mit Office direkt gar nichts zu tun haben. Die Akzeptanz von eigenen Anwendungen wird dadurch in der Regel deutlich erhöht, und die beste Anwendung taugt nichts, wenn sie vom Anwender nicht angenommen wird. Zum anderen setzt die Entwicklung mit VSTO auf .NET und Visual Studio auf, sodass .NET-Entwickler auf ihre vorhandenen Kenntnisse aufbauen können und keine große Einarbeitungszeit in VSTO benötigen.

■ 1.1 An wen richtet sich dieses Buch?

Das Buch richtet sich an Entwickler, die bereits Erfahrungen mit .NET und Visual Studio gesammelt und die keine Schwierigkeiten haben, eine eigene Anwendung mit Visual C# oder Visual Basic zu erstellen. Grundlagen von C# oder Visual Basic sind für das Verständnis der Inhalte in diesem Buch Voraussetzung. Da Visual C++ auch mit den CLR-Erweiterungen nicht als Sprache für VSTO-Projekte zur Verfügung steht, sind reine C++-Entwickler etwas benachteiligt. Aufgrund der Ähnlichkeit zwischen C# und C++ sollten jedoch auch C++-Entwickler in der Lage sein, die in C# gehaltenen Beispiele ohne große Schwierigkeiten zu verstehen und an die eigenen Bedürfnisse anzupassen sowie zu erweitern. Kenntnisse in VBA werden nicht vorausgesetzt, reichen für den ambitionierten Entwickler aber durchaus aus, um mit diesem Buch zu arbeiten.

Visual Basic-Entwickler sollten sich nicht davon abschrecken lassen, dass die Beispiele durchgehend in C# gehalten sind. Da beide Sprachen das .NET Framework bedienen, sind die Unterschiede bis auf wenige Besonderheiten doch deutlich geringer als oftmals angenommen. So sollte eigentlich jeder C#-Entwickler Code lesen und verstehen können, der in Visual Basic gehalten ist und umgekehrt. Die Unterschiede zwischen C# und Visual Basic kommen in diesem Buch ohnehin nicht zur Geltung, da der Schwerpunkt der Inhalte die Objektmodelle der einzelnen Office-Anwendungen sind und nicht die Programmiersprachen an sich. Auf den Abdruck der Beispiele sowohl in C# als auch in Visual Basic wurde lediglich aus Platzgründen verzichtet.

■ 1.2 Was wird in diesem Buch behandelt

Mit dem Schwerpunkt auf die Objektmodelle von Office spielt bei der reinen .NET-Programmierung ohne VSTO die verwendete Office-Version nur eine untergeordnete Rolle. Dies liegt darin begründet, dass sich die Objektmodelle der einzelnen Office-Anwendungen seit den letzten Versionen kaum geändert haben. Die größten Änderungen gibt es im Zusammenhang mit Features, die erst mit neueren Versionen eingeführt wurden. Code, der auf diese Funktionen zugreift, funktioniert selbstverständlich auch nur in den Office-Versionen, in denen diese Funktionen enthalten sind. Für allgemeine Aufgaben wie das Auslesen von Werten aus einem Excel-Arbeitsblatt gibt es zwischen den verschiedenen Office-Versionen keine Unterschiede, sodass sich viele Beispiele auch auf Office 2003 oder Office 2007 übertragen lassen.

Nichtsdestotrotz wurden die Beispiele in diesem Buch ausschließlich unter Office 2010 entwickelt und getestet, und es wird auch ausschließlich auf die aktuelle Version von Office und VSTO Bezug genommen. Dabei liegt der Schwerpunkt auf den Anwendungen

- Word
- Excel
- Outlook

Die Codeerstellung für VSTO und .NET ist bis auf wenige Ausnahmen identisch. VSTO stellt in Form von Add-Ins oder Dokumentvorlagen einen eigenen Rahmen für Erweiterungen zur Verfügung. Mit VSTO werden also selbstständige Anwendungen entwickelt. Eine andere Möglichkeit ist es, die Objektmodelle aus eigenen Anwendungen heraus zu verwenden, um beispielsweise eine Berichtsausgabe in Excel zu automatisieren. Sowohl in VSTO-Anwendungen als auch in eigenen .NET-Anwendungen werden dabei die gleichen Objektmodelle verwendet, sodass ab einem gewissen Punkt auch der Code identisch ist. Lediglich der Rahmen, in dem der Code abläuft, ist ein anderer. In diesem Buch wird daher zunächst auf die Besonderheiten von VSTO eingegangen, worauf die Beispiele zu den einzelnen Office-Anwendungen sich dann sowohl aus VSTO als auch aus eigenen .NET-Anwendungen heraus anwenden lassen.

VSTO bietet ebenso Unterstützung für die Erweiterung von PowerPoint, Visio, Project und InfoPath, worauf in diesem Buch nur teilweise und einführend eingegangen wird.

Während in früheren Versionen von Visual Studio beziehungsweise von VSTO auch die Entwicklung von Formularvorlagen für InfoPath unterstützt wurde, besteht diese Möglichkeit in

VSTO 4.0 respektive Visual Studio 2010 nicht mehr. Auf die Entwicklung von Formularvorlagen und andere Möglichkeiten aus vergangenen Versionen wird daher nicht eingegangen.

Ebenso nicht zu den Themen gehört die Entwicklung von klassischen COM-Add-Ins für Office. Im Zeitalter von VSTO ist die Entwicklung solcher Add-Ins ohnehin unnötig aufwendig und umständlich. So gibt es für COM-Add-Ins beispielsweise keine visuellen Designer. Auch wurde der Funktionsumfang für COM-Add-Ins nicht mehr erweitert, wohingegen VSTO kontinuierlich weiterentwickelt wird. Die Entwicklung klassischer COM-Add-Ins ist eigentlich nur noch notwendig und sinnvoll, wenn Add-Ins eine Abwärtskompatibilität zu früheren Office-Versionen besitzen müssen und sich der Aufwand für die Erstellung von separaten versionspezifischen Add-Ins nicht lohnt. Dies ist in der Regel nur bei einfachen Add-Ins mit geringem Funktionsumfang der Fall. Bei komplexeren Add-Ins profitieren die auf .NET basierenden Add-Ins von einem deutlich höheren Funktionsumfang und einer besseren Wartbarkeit des Codes, zumal sich gemeinsamer Code ganz einfach als DLL auslagern lässt.

In der frühen Planungsphase dieses Buchs war zunächst vorgesehen, dass auch die in VSTO 3.0 eingeführten Workflow-Vorlagen für SharePoint Server sowie die Grundlagen der SharePoint-Entwicklung mit .NET behandelt werden sollten. Mit Visual Studio 2010 wurden die Entwicklungsmöglichkeiten für SharePoint Server jedoch dermaßen erweitert, dass dieses Thema nun ein eigenes Buch füllen würde und ein Einstieg in das Thema auf circa 100 Seiten innerhalb dieses Buchs nicht mehr angemessen ist.

■ 1.3 Aufbau dieses Buchs

Dieses Buch liefert nicht nur praktische Beispiele mit Schritt-für-Schritt-Anleitungen, sondern vermittelt Ihnen auch die notwendigen theoretischen Kenntnisse für die Office-Entwicklung mit VSTO und .NET. Das Buch ist in insgesamt neun Kapitel mit unterschiedlichen Schwerpunkten unterteilt. Mit der folgenden Zusammenfassung können Sie sich einen ersten Überblick verschaffen, was Sie in den einzelnen Kapiteln erwartet:

- Nach der Einleitung, die Sie gerade lesen, liefert das zweite Kapitel eine Einführung in die Office-Entwicklung mit VSTO. Sie erfahren hier unter anderem, welche Änderungen und Neuerungen es in der aktuellen Version von VSTO gibt, und es wird auch auf die Unterschiede zwischen VSTO und VBA eingegangen. Anschließend folgt eine Beschreibung der unterschiedlichen Projekttypen und Features, die VSTO bereithält. Am Ende des Kapitels werden Sie wissen, was mit VSTO alles möglich ist.
- Kapitel drei vermittelt die Möglichkeiten von VSTO, die zum Teil auf eine bestimmte Office-Anwendung beschränkt sind und teilweise auch für mehrere Office-Anwendungen eine Rolle spielen. Anhand von kleinen Office-Lösungen werden Anpassungen auf Dokumentenebene und Add-Ins auf Anwendungsebene vorgestellt. Die Beispiele umfassen die Verwendung von Hostelementen und Hoststeuerelementen, den Zugriff auf Daten, die Verwendung von Inhaltssteuerelementen und Lesezeichen, die Anpassung von Menübändern und vieles mehr.
- Die Kapitel vier bis sechs widmen sich jeweils einer der am meisten verwendeten Office-Anwendungen: Word, Excel und Outlook. Nach einer Einführung in die wichtigsten Klassen

und Methoden der einzelnen Objektmodelle werden die Grundlagen in einer praktischen Übung vertieft. Dabei wird die Entwicklung mit VSTO ebenso berücksichtigt wie die Automatisierung von Office aus eigenen .NET-Anwendungen heraus.

- Nach der ausführlichen Behandlung von Word, Excel und Outlook in den vorausgegangenen Kapiteln wird im siebten Kapitel kurz auf die Entwicklung von Lösungen für weitere Office-Anwendungen eingegangen.
- Das achte Kapitel wendet sich insbesondere an Office-Entwickler, die in verschiedenen Welten zu Hause sind und sowohl mit VSTO als auch mit VBA entwickeln. Bei dem parallelen Einsatz beider Techniken ist oftmals eine Interaktion zwischen den verschiedenen Lösungen erforderlich. Wie dies geht, erfahren Sie in diesem Kapitel. Darüber hinaus wird auch auf die Bearbeitung von Office-Dateien mit Office Open XML eingegangen. Mit Office Open XML ist es möglich, Office-Dateien ohne Verwendung der Objektmodelle zu bearbeiten.
- Zum Abschluss wird im letzten Kapitel auf Überlegungen zur Migration, Bereitstellung und Sicherheit von Office-Lösungen eingegangen.

Das Buch kann dabei vom Anfang bis zum Ende durchgearbeitet oder auch als Nachschlagewerk für gezielte Themen verwendet werden. Wer noch gar keine Erfahrungen mit der Office-Entwicklung unter VSTO und .NET gesammelt hat, sollte das Buch am besten von vorne bis hinten durchlesen und die Beispiele nachprogrammieren, um einen vollständigen Überblick der Möglichkeiten zu erhalten. Wer sich gezielt für die Entwicklung für eine bestimmte Office-Anwendung interessiert, sollte die Kapitel zwei und drei sowie anschließend das passende Kapitel vier bis sechs lesen.

Während ein Buch über beispielsweise ASP.NET dafür prädestiniert ist, ein großes Beispiel durch das gesamte Buch zu ziehen und dieses nach und nach aufzubauen beziehungsweise zu erweitern, bietet sich dies bei VSTO aus verschiedenen Gründen nicht an. Der offensichtlichste Grund ist, dass mit VSTO vollkommen unterschiedliche Anwendungen bedient werden, die jeweils für verschiedene Einsatzzwecke gedacht sind. Ein großes Beispiel zu konstruieren, das alle Anwendungen aus der Office Suite berücksichtigt, wäre praxisfremd und an den Haaren herbeigezogen. Daher werden in diesem Buch kleinere und in sich abgeschlossene Beispiele verwendet, die sich dafür aber auch leichter nachvollziehen und vor allem wiederverwenden lassen.



HINWEIS: Sämtliche Beispiele finden Sie unter <http://downloads.hanser.de>

■ 1.4 Voraussetzungen an die Arbeitsumgebung

Das Buch orientiert sich an der Entwicklung mit Visual Studio 2010 für Office 2010. Die Beispiele in diesem Buch wurden daher in der folgenden Umgebung entwickelt und getestet:

- Windows 7 x64 mit Service Pack 1
- Office 2010 x86
- Visual Studio 2010 mit Service Pack 1
- SQL Server 2008 inkl. AdventureWorks-Beispieldatenbank

Mit zum Teil nur geringen Einschränkungen lassen sich auch andere Versionen von Visual Studio und Office verwenden. Die nachfolgenden Abschnitte geben dazu einen kurzen Überblick.

1.4.1 Visual Studio

Für die vollständige Umsetzung aller Beispiele benötigen Sie Visual Studio 2010 Professional oder höher sowie Office 2010. Die kostenlosen Express Editionen von Visual Studio bieten keine Unterstützung für VSTO. Die meisten der in diesem Buch beschriebenen Funktionen von VSTO stehen bereits ab VSTO 3.0 und somit ab Visual Studio 2008 und Office 2007 zur Verfügung. Sie können dieses Buch also durchaus auch verwenden, um sich in die VSTO-Entwicklung für Office 2007 mit Visual Studio 2008 einzuarbeiten.

Bei der reinen .NET-Entwicklung für Office sehen die Voraussetzungen an die Version von Visual Studio hingegen ganz anders aus: Da in diesem Fall lediglich mit einem Verweis auf die Objektmodelle gearbeitet wird, können Sie auch auf Visual Studio 2005 und sogar auf die kostenlosen Express-Editionen zurückgreifen.

1.4.2 Office

Auch wenn in Visual Studio 2010 sowohl Projektvorlagen für Office 2007 als auch für Office 2010 pauschal zur Verfügung stehen, können jedoch nur für die Office-Version Lösungen entwickelt werden, die auch lokal auf dem Entwicklungsrechner installiert ist. Eine parallele Installation von Office 2007 und Office 2010 wird von VSTO nicht unterstützt, sodass Sie sich für eine der beiden Versionen entscheiden beziehungsweise gegebenenfalls zwei unterschiedliche Entwicklungsumgebungen aufsetzen müssen.



HINWEIS: Das Entwickeln von Office-Lösungen wird von Visual Studio 2010 nicht unterstützt, wenn sich auf dem Entwicklungsrechner eine Version von Office 2010 befindet, die über Click-to-Run installiert wurde.

1.4.3 SQL Server

In einigen Beispielen werden Daten aus der Beispieldatenbank AdventureWorks herangezogen. Wenn Sie diese Beispiele nachvollziehen möchten, benötigen Sie Zugriff auf einen SQL Server, auf dem die unter <http://msftdbprodsamples.codeplex.com/releases/view/37109> erhältliche Beispieldatenbank installiert ist. Die Version von SQL Server (2005 oder höher) sowie die damit verbundene Version von AdventureWorks spielt hierfür keine Rolle, auch wenn es zwischen den Versionen der Beispieldatenbank kleine Unterschiede gibt. Es reicht vollkommen aus, wenn Sie die standardmäßig zusammen mit Visual Studio 2010 installierte Version von SQL Server 2008 Express Edition verwenden. Die gleiche Datenbank wurde auch für dieses Buch verwendet.