

PROGRAMMER'S CHOICE

Michael Kofler
Michael Kofler

Ralf Nebelo
Ralf Nebelo

Excel 2007 programmieren

Anwendungen entwickeln und
Abläufe automatisieren mit VBA

7. Auflage

 ADDISON-WESLEY

Sofort einsetzbares Add-in
zur Nachbildung der
klassischen Menüleiste





Intuitiver Einstieg



1	Das erste Makro	21
2	Neuerungen in Excel 2007	61





2 Neuerungen in Excel 2007

Mit der Versionsnummer 2007 hat Excel einen großen Entwicklungssprung gemacht. Augenfälligstes Merkmal ist eine radikal neue Benutzeroberfläche, welche die Menüs und Symbolleisten früherer Excel-Versionen (fast) vollständig ersetzt.

Aber auch »unter dem Blech« hat Excel 2007 funktional zugelegt. Das zeigt sich unter anderem an einer neuen Diagramm-Engine, die nicht mehr nur einfache »Strichzeichnungen« produziert, sondern erstmals auch sehenswerte 3D- und Rendering-Effekte zu bieten hat. Mit den völlig neuen SmartArt-Diagrammen ist jetzt die optisch eindrucksvolle Darstellung von Beziehungen und Prozessen möglich. Und mit der kräftig »aufgebohrten« bedingten Formatierung lassen sich Zahlenwerte nun erstmals innerhalb von Zellen (!) als Diagramme darstellen, was sehr viel Übersicht in Arbeitsblätter bringt. Darüber hinaus gibt es neue XML-basierte Dateiformate für die Speicherung von Excel-Dokumenten, Arbeitsblätter mit drastisch erhöhter Kapazität und eine Vielzahl von kleineren Neuheiten und funktionalen Verbesserungen.

In Sachen Programmierung steht mit Excel 2007 eine Ablösung bevor. Zwar ist VBA nach wie vor das Entwicklungswerkzeug Nummer eins, mit dem Sie auch viele der neuen Excel-Funktionen ansprechen können. Einige Neuerungen aber – darunter so wichtige wie die Multifunktionsleistenoberfläche und die SmartArt-Diagramme – haben keinen Eingang mehr in das Objektmodell von



Excel 2007 gefunden, was den VBA-Einsatz verhindert und eine neue Generation von Entwicklungswerkzeugen erforderlich macht.

2.1 Die neue Benutzeroberfläche

Die neue »RibbonX« oder auch »Fluent« genannte Oberflächenarchitektur von Excel 2007 präsentiert die zahllosen Programmfunktionen auf eine wesentlich übersichtlichere und aufgabenorientierte Art, was insbesondere Einsteigern zugute kommt. Umsteiger dagegen müssen mit einer deutlich längeren Eingewöhnungszeit rechnen, weil sie zuerst einmal vieles vergessen müssen, was sie über die bisherige Excel-Bedienung wussten. Die Mühe lohnt allerdings, weil es sich mit RibbonX tatsächlich effektiver arbeiten lässt.

Die Multifunktionsleiste

Die RibbonX-Oberfläche schiebt Menüs und Symbolleisten weitestgehend in den Ruhestand. Deren Stelle nimmt nun die sogenannte *Multifunktionsleiste* ein, die einen Satz von aufgabenorientierten *Befehlsregisterkarten* enthält. Aufgabenorientiert bedeutet, dass jede Registerkarte nur die Befehle bereitstellt, die für die Erledigung einer bestimmten Aufgabe nützlich sind. So verfügt Excel unter anderem über Registerkarten für die Gestaltung von Arbeitsblättern (SEITENLAYOUT) oder den Umgang mit Rechenfunktionen (FORMELN).



Abbildung 2.1: Neben neun ständig sichtbaren Registerkarten enthält die Multifunktionsleiste auch solche, die nur bei Bedarf angezeigt werden.

Neben einem festen Inventar an Registerkarten zeigt die Multifunktionsleiste auch kontextbezogene Exemplare an. Die erscheinen ganz automatisch, wenn der Anwender bestimmte Objekte – ein Diagramm oder eine Grafik etwa – bearbeitet.

Die Befehle auf den Registerkarten bilden thematisch geordnete *Befehlsgruppen* und stehen zumeist in Form von unterschiedlich großen Schaltflächen oder Listenfeldern zur Verfügung. Das erfordert zwar deutlich mehr Platz als eine schlanke Menüzeile, bringt aber wesentlich mehr Übersicht und spürbar verkürzte Mauswege mit sich, da es nur eine »Klickebene« gibt, in der stets alle Befehle sichtbar sind. Das bislang übliche »Abtauchen« in tief verschachtelte Menüstrukturen entfällt also.

Sollte sich das Platzangebot auf dem Bildschirm durch eine Änderung der Bildschirmauflösung oder durch das Verkleinern des Programmfensters verringern, zeigt Excel 2007 dennoch weiterhin alle Befehle der jeweiligen Registerkarte an. Das gelingt dem Programm durch intelligente »Sparmaßnahmen«, die von einer schrittweisen Reduzierung der Schaltflächengrößen über das Weglassen von Beschriftungen bis hin zum Zusammenfassen von Befehlen zu Listenfeldern reichen.

Kataloge und Livevorschau

Neben Schaltflächen und Listenfeldern findet man in den Registerkarten der Multifunktionsleiste auch ein völlig neues Steuerelement namens *Katalog*. Damit kann der Anwender viel Zeit bei der Bewältigung von Auswahl- und Formatierungsaufgaben sparen, weil er sich nicht mehr im Versuch-und-Irrtum-Prinzip mit Dialogfeldern herumschlagen muss, um ein bestimmtes Resultat zu erzielen. Stattdessen zeigt ihm das Katalog-Control ein Vorschaubild der zu erwartenden Ergebnisse an, sodass er sich nur noch das gewünschte aussuchen muss.

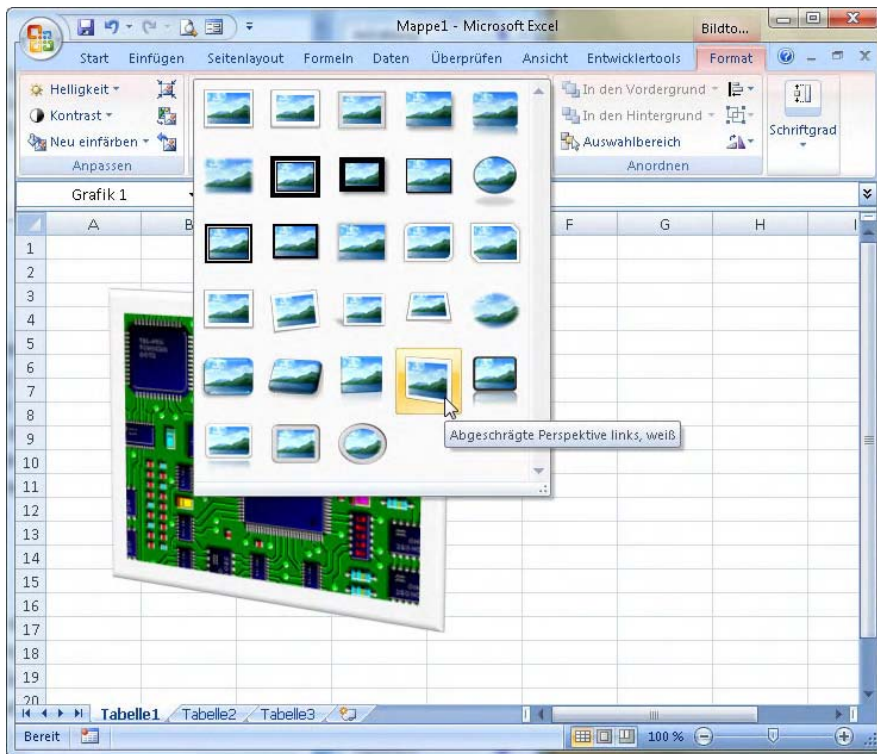


Abbildung 2.2: Kataloge zeigen per Vorschaubild und zudem »live« im Dokument an, wie sich ein Befehl auswirken wird.

Weil die angezeigte Miniaturvorschau unter Umständen nicht ausreicht für eine Beurteilung, überträgt Excel die Einstellungen, die dem aktuell gewählten Katalogelement entsprechen, dynamisch auf das Arbeitsblatt. Mit dieser *Livevorschau* kann man die Auswirkungen einer möglichen Formatierungsänderung also schon vorab und im Kontext des Dokuments bewerten.

Neue Symbolleisten

Die neue *Minisymbolleiste* hilft ebenfalls bei der Vermeidung unnötiger Dialogfeldeingaben und Mauswege. Klickt man einen Zellwert mit der rechten Maustaste an, so erscheint darüber ein halbtransparentes Tool, das einem die wichtigsten Formatierungswerkzeuge zur Verfügung stellt. Damit kann man dann »schnell mal eben« Schriftart und -größe verändern, Text- und Hintergrundfarben ändern, Zahlenformate einstellen und dergleichen mehr.



Abbildung 2.3: Die neue Minisymbolleiste fasst die wichtigsten Formatierungsoptionen an einem Ort zusammen.

Trotz ihres Namens ist die Minisymbolleiste keine Symbolleiste im herkömmlichen Sinn, ihr Befehlsvorrat lässt sich also nicht den Wünschen des Anwenders entsprechend verändern. Das funktioniert nur noch bei der *Symbolleiste für den Schnellzugriff*, die oberhalb der Multifunktionsleiste zu finden ist und in die man mithilfe des Dialogs DATEI | EXCEL-OPTIONEN | ANPASSEN jeden verfügbaren Excel-Befehl integrieren kann.

Das neue Datei-Menü

Hinter dem blumigen *Office-Button* in der linken, oberen Ecke des Excel-Programmfensters befindet sich nicht etwa das »Office-Menü«, wie man angesichts seiner Lage meinen könnte, sondern das neue DATEI-Menü. Es verdankt seinen Namen wohl der Tatsache, dass es funktional weitgehend mit dem DATEI-Menü früherer Excel-Versionen übereinstimmt und die Befehle zum Öffnen, Speichern, Drucken und Verteilen von Dokumenten enthält.

Darüber hinaus sind aber alle übrigen Menüs und Symbolleisten aus Excel 2007 entfernt worden und lassen sich auch nicht über einen (nicht vorhandenen, aber von vielen Umsteigern herbeigesehnten) »Kompatibilitätsschalter« wieder zum Vorschein bringen. Das einzige bedienungstechnische Zugeständnis an alte Excel-Hasen ist die Tatsache, dass die meisten Tastenkürzel weiterhin funktionieren. Das gilt allerdings im vollen Umfang nur für Funktionstasten sowie Kombinationen mit der Strg-Taste.

Die menüorientierten Tastenkürzel, bei denen zuerst die [Alt]-Taste gedrückt werden muss, sind zum größten Teil unwirksam.

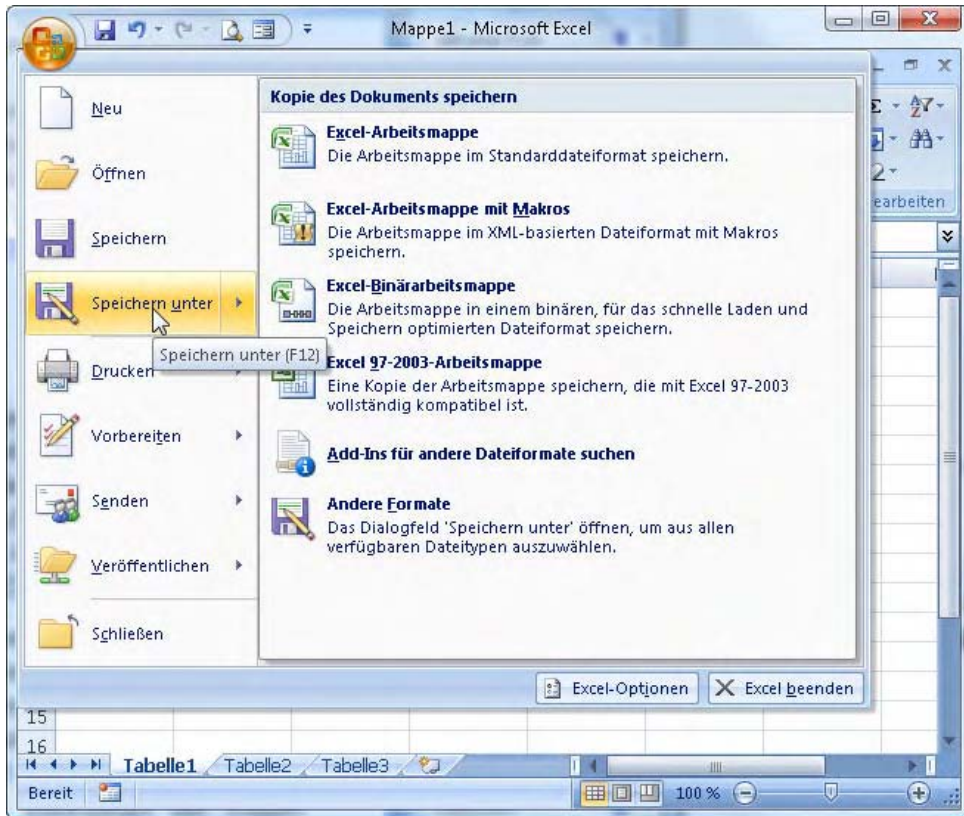


Abbildung 2.4: Mit Ausnahme des Datei-Menüs gibt es in Excel 2007 keine Menüs mehr.

* * * TIPP

Auch wenn die originalen Menüs und Symbolleisten aus Excel entfernt wurden, so lassen sie sich doch mit etwas Handarbeit und XML-Code nachbilden. Wie das im Detail funktioniert, wird ausführlich im Abschnitt 8.3.4 beschrieben.

2.2 Neue Programmfunktionen

Neue Diagramm-Engine

Excel 2007 greift auf eine neue Diagramm-Engine zurück, die nun auch in Word und PowerPoint 2007 zum Einsatz kommt. Damit lassen sich nüchterne Zahlen in informa-

tive Grafiken umsetzen. Das wichtigste Merkmal der neuen Diagrammfunktion ist eine deutlich verbesserte Ausgabequalität. Das zeigt sich insbesondere bei den dreidimensionalen Diagrammtypen.

Über die kontextsensitiven Befehlsregister ENTWURF, LAYOUT und FORMAT lassen sich aber auch einfache Kreis- und Säulendiagramme beispielsweise mit echten 3D-Effekten, sauber gerenderten Schattenverläufen, Transparenzen oder geglätteten Kanten ausstatten. Bei Bedarf ist sogar ein Griff in die Trickkiste erlaubt, wo »glühende Farben« und andere TV-verdächtige Spezialeffekte auf ihren Einsatz warten.

SmartArt-Grafiken

Die *Schematischen Darstellungen* von Excel 2003 gibt es nicht mehr. Stattdessen kann der Anwender nun die neuen *SmartArt*-Diagramme verwenden, um Prozesse, Abläufe oder Beziehungen grafisch darzustellen. Im Gegensatz zum erwähnten Vorgänger stehen nun wesentlich mehr und wesentlich aufwendiger gestaltete Diagrammtypen zur Auswahl, die sich sehr unkompliziert beschriften, konfigurieren und über das kontextsensitive Befehlsregister SMARTART-TOOLS nachbearbeiten lassen.

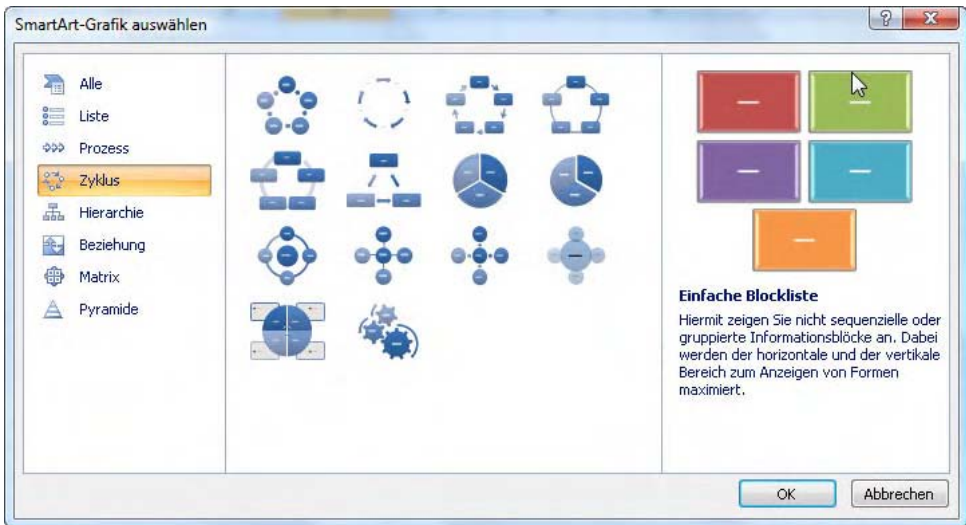


Abbildung 2.5: SmartArt-Diagramme eignen sich insbesondere für die Visualisierung von Prozessen und Hierarchien.

Da die SmartArt-Ergebnisse professionellen Standards entsprechen, können sich wohl viele Excel 2007-Eigner die Anschaffung von separaten Diagramm-Tools wie Visio beispielsweise sparen.

Bedingte Formatierung

Die stark erweiterte *Bedingte Formatierung* erlaubt nun erstmals die grafische Darstellung von Zahlen unmittelbar im Arbeitsblatt. Der Trick besteht darin, die Zellen einer markierten Zahlenreihe mit halbtransparenten, farbigen Balken zu hinterlegen. Dabei entspricht die Länge jedes einzelnen Balkens dem prozentualen Wert der zugehörigen Zelle im Vergleich zum größten Wert innerhalb der Markierung. Dadurch lassen sich Trends, Muster oder abweichende Einzelwerte etwa auch ohne Erstellung eines Diagramms sofort erkennen.



Abbildung 2.6: Datenbalken-Diagramme veranschaulichen den relativen Wert einer Zahl direkt in der Tabelle.

Anstelle solcher *Datenbalken* kann man auch *Farbskalen* oder *Symbolsätze* verwenden. Im letzteren Fall verdeutlicht Excel die relativen Zellwertigkeiten nicht mit Balken, sondern beispielsweise mithilfe von vorangestellten Pfeilsymbolen oder Ampellichtern. Das erlaubt zwar nur wenige Abstufungen (maximal fünf), lässt den Anwender aber Entwicklungen beispielsweise besonders schnell erkennen.

Größere Arbeitsblätter

Die Maximalgröße für Arbeitsblätter beträgt nun eine Million Zeilen und 16.000 Spalten gegenüber 65.536 Zeilen und 256 Spalten in Excel 2003. Da es wohl in absehbarer Zeit kaum Kalkulationen geben dürfte, die derart große Tabellen erfordern, dürften von dieser Kapazitätsaufstockung insbesondere die vielen Anwender profitieren, die Excel als Datenbank »missbrauchen«.

Neue Dateiformate

Arbeitsmappen und andere Excel-Dokumente werden nun standardmäßig nicht mehr in einem binären Format gespeichert, sondern in einem neuen, das auch von Menschen gelesen werden kann und auf der Auszeichnungssprache XML (steht für »Extensible Markup Language«) beruht. Dieses neue »Open XML« genannte Format unterscheidet sich aber grundlegend von dem XML-Format, das schon in Excel 2003 zur Auswahl stand.

Für den Anwender zeigt sich die Existenz des neuen Dateiformats eigentlich nur in den neuen Dateiendungen, die sich um ein angehängtes »X« von den bisherigen unterscheiden. So werden nun Arbeitsmappen beispielsweise nicht mehr in XLS-Dateien gespeichert, sondern in XLSX-Dateien. Vorlagen sichert man anstatt in XLT- nun in XLTX-Dateien.

Das gilt allerdings nur, solange Arbeitsmappen und Vorlagen *keine* VBA-Makros enthalten. Ist das der Fall, tritt ein »M« an die Stelle des angehängten »X«. Für Arbeitsmappen mit Makros lautet die korrekte Dateiendung dann XLSM, für Vorlagen mit Makros XLTM. Excel-Add-Ins enthalten ja grundsätzlich Makros, weshalb es hier nur eine Dateiendung gibt, die XLAM lautet.

Beim Speichern von Excel-Dokumenten stehen die alten Dateiendungen aus Kompatibilitätsgründen weiterhin zur Wahl. Dann erfolgt die Speicherung allerdings im alten Binärformat und unter Verzicht auf Dokumentelemente, die mit Excel 2007 neu eingeführt wurden. So lassen sich beispielsweise die neuen *Datenbalken*-Diagramme, die im Rahmen der *Bedingten Formatierung* zur Verfügung stehen, nur in dem neuen Open-XML-Format speichern.

> > > HINWEIS

Open-XML-Dateien sind übrigens keine Dateien im eigentlichen Sinn, sondern »getarnte« ZIP-Archive. Wer sich den (aus vielen XML-Dateien bestehenden) Inhalt eines solchen Archivs anschauen möchte, kann die Endung ».zip« an den Dateinamen anhängen (via Umbenennen-Befehl) und das Archiv dann per Doppelklick im Windows-Explorer öffnen.

2.3 Neues in Sachen Programmierung

Angesichts des völlig neuen Erscheinungsbilds könnte man meinen, Microsoft hätte das Kalkulationsprogramm von Grund auf neu programmiert. Was aber – Gates sei Dank! – nicht stimmt. Stattdessen basiert auch Excel 2007 noch auf dem guten alten »Component Object Model« oder kurz: COM. Das heißt grob verkürzt nichts anderes, als dass man auch die neueste Excel-Version immer noch über COM-fähige Entwicklungstools programmieren kann. Und da steht das »eingebaute« und kostenlose VBA immer noch leistungsmäßig in der ersten Reihe – was die meisten Entwickler wohl freuen dürfte.

Schon jetzt aber gibt es einige (zumeist anspruchsvollere) Programmieraufgaben, die sich mit VBA entweder gar nicht mehr oder nur noch teilweise lösen lassen. Für einen Teil dieser Fälle benötigt man nur einen Texteditor und gute XML-Kenntnisse. Andere Problemfälle verlangen nach einem neuen, äußerst mächtigen, aber leider auch sehr teuren Entwicklungswerkzeug, das sich »Visual Studio Tools for Office« nennt.

Kompatibilitätskrücke Add-Ins-Register

Das Festhalten an der Programmierschnittstelle COM bedeutet auch, dass zahllose Excel-Lösungen, die für frühere Versionen des Kalkulationsprogramms entwickelt wurden und in Gestalt von »makrohaltigen« Arbeitsmappen sowie VBA- und COM-Add-Ins vorliegen, weiterhin funktionieren. Das gilt sogar für Lösungen, die sich über Menüerweiterungen oder eigene Symbolleisten in die Oberfläche früherer Excel-Versionen eingeklinkt hatten – obwohl es diese Oberfläche gar nicht mehr gibt! Wie ist das möglich?

Antwort: Weil Excel 2007 zwar weiterhin die *CommandBars*-Objekte (siehe Abschnitt 8.1.3) unterstützt, entsprechende Programmieranweisungen aber nicht mehr in *echte* Menüs und Symbolleisten umsetzt, sondern in Nachbildungen, die zu hundert Prozent aus den Elementen der neuen RibbonX-Oberfläche bestehen. Und diese »simulierten« Befehlsleisten – so der Oberbegriff von Menü- und Symbolleisten – werden auch nicht mehr an den ursprünglich vorgesehenen Bildschirmpositionen angezeigt, sondern in einer einzigen Befehlsregisterkarte namens ADD-INS. Die bildet somit eine Art Kompatibilitätskrücke, welche die Bedienbarkeit älterer Excel-Lösungen grundsätzlich sicherstellt.

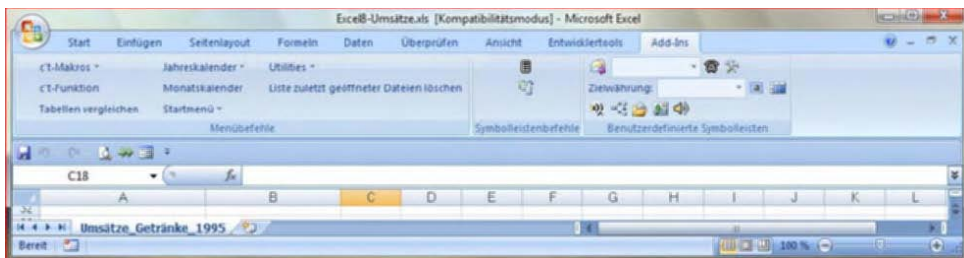


Abbildung 2.7: Menüs und Symbolleisten älterer Lösungen zeigt Excel 2007 jetzt nur noch im Add-Ins-Register an.

Dass es sich beim ADD-INS-Register in der Tat nur um einen Behelf handelt, zeigt sich an deutlichen Einschränkungen in Sachen Handling. So lassen sich die hier angezeigten Befehlsleistenattrappen beispielsweise nicht mehr verschieben, andocken oder manuell anpassen, und zugeordnete Tastenkürzel für den schnellen Aufruf von selbst gebauten Menükommandos funktionieren auch nicht mehr. Angesichts des radikal neuen Ober-

flächenkonzepts erscheint das ADD-INS-Konstrukt aber dennoch als guter Kompromiss zwischen Fortschritt und Kompatibilität mit vorhandenen Excel-Lösungen.

Zu- und Abgänge im Objektmodell

Viele neue Excel-Features spiegeln sich auch im Objektmodell des Kalkulationsprogramms wider. Und zwar in Form von neuen oder erweiterten Objekten, die sich mit VBA nutzen lassen.

Bei den Neuzugängen ist unter anderem das *Databar*-Objekt erwähnenswert, das die Programmierung der oben vorgestellten Datenbalken- und anderer Zelldiagramme ermöglicht (siehe Abschnitt 10.6). Die neuen Darstellungstalente der erweiterten Diagramm-Engine lassen sich unter anderem über die neuen Chart-Methoden *ChartStyle* und *ApplyLayout* (siehe Abschnitt 10.2.2) wecken.

Den Erweiterungen der Objektbibliothek stehen aber auch einige »ausgeblendete«, das heißt nicht mehr unterstützte Objekte, Methoden und Eigenschaften gegenüber. Zu den prominentesten Abgängen gehört das *FileSearch*-Objekt, das die simple Programmierung einer Dateisuchfunktion ermöglichte und in zahlreichen Excel-Lösungen (die jetzt nicht mehr funktionieren!) zum Einsatz kam. Mithilfe einer »selbst gebauten« rekursiven Funktion (siehe Abschnitt 5.6.2) lässt sich das Problem jedoch halbwegs elegant beheben.

Eine vollständige Auflistung aller Objektmodelländerungen finden Sie hier:

<http://msdn2.microsoft.com/en-us/library/bb149067.aspx>

Anpassen der Multifunktionsleiste

Dank der *CommandBars*-Objekte lassen sich – wie schon erwähnt – auch in Excel 2007 noch Menüs und Symbolleisten anlegen, die dann jedoch ausschließlich in der Befehlsregisterkarte ADD-INS erscheinen. Wer aber in die Zukunft denkt, sollte sich von *CommandBars* und dem ADD-INS-Register verabschieden und die Funktionen seiner Lösung lieber gleich in die neue Multifunktionsleisten-Oberfläche integrieren. Bei diesem Vorhaben kommt man mit VBA allein aber nicht mehr zurecht.

Das liegt daran, dass die Elemente der Multifunktionsleiste nicht mehr im Objektmodell von Excel abgebildet werden. Das schließt jetzt programmierte Änderungen der Benutzeroberfläche *zur Laufzeit* einer Excel-Lösung aus – was Microsoft damit begründet, dass solche Manipulationen in der Vergangenheit oft missbräuchlich und zum Nachteil des Anwenders vorgenommen wurden.

Wenn eine Excel-Lösung die RibbonX-Oberfläche ändern oder erweitern will, kann sie das jetzt nur noch »im Vorfeld« erledigen, das heißt, während sie geladen wird. Die Anpassung geschieht aber nicht mehr per VBA, sondern über XML-Code, der in die Datei der Excel-Lösung eingebettet ist und ausschließlich beim Öffnen ausgeführt wird.

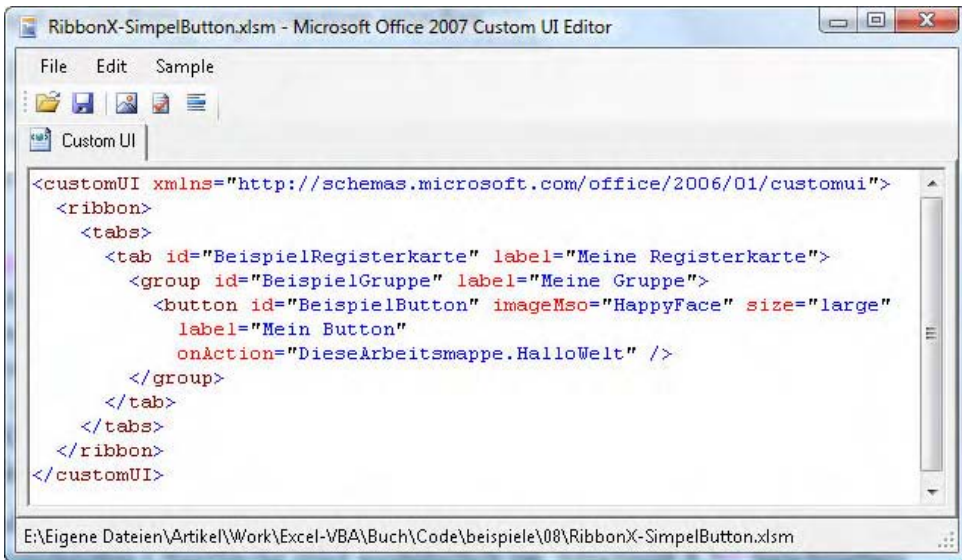


Abbildung 2.8: XML-Code bestimmt jetzt das Design von Oberflächenanpassungen; für die Funktion sind VBA-Makros zuständig.

Die XML-Anweisungen bestimmen aber nur das »Design« der gewünschten Oberflächenanpassung, indem sie der Multifunktionsleiste beispielsweise eine neue Befehlsregisterkarte hinzufügen und darin eine neue Schaltfläche anlegen. Für die Funktion – also das, was beim Anklicken dieser Schaltfläche geschieht – ist nicht mehr der XML-Code, sondern eine sogenannte Callback-Routine zuständig. Und dabei handelt es sich um eine spezielle Form von VBA-Makro.

Die Einzelheiten über das (gar nicht so schwierige) Anpassen der Multifunktionsleisten-Oberfläche sind ausführlich in Abschnitt 8.3 beschrieben. Dieser endet übrigens mit einer Passage, die eine vollständig funktionierende Nachbildung des klassischen Excel 2003-Menüs mit den Mitteln der RibbonX-Architektur beschreibt.

Die Grenzen von VBA

Neben der großen Zahl von Aufgaben, die sich mit VBA immer schon lösen ließen, erlaubt das Entwicklungs-Tool nun also die programmierte Nutzung aller neuen Excel-Funktionen, die in das Objektmodell aufgenommen wurden. Darüber hinaus kann man mit VBA den Funktionscode für individuelle (in XML codierte) Anpassungen der neuen Multifunktionsleisten-Oberfläche schreiben.

Trotz dieses ansehnlichen Leistungsspektrums aber gibt es Aufgaben, denen VBA nicht mehr gewachsen ist. Dazu gehört etwa das Anlegen von individuell gestalteten *Aufgabenbereichen*, die, je nachdem, ob sie anwendungsweit oder nur auf Dokument-

ebene verfügbar sind, als »Custom Task Panes« oder »Action Panes« bezeichnet werden. Die Entwicklung von maßgeschneiderten *Smart Tags* – jenen Minimenüs, die dem Anwender zu bestimmten Dokumentinhalten passende Aktionen zur Verfügung stellen – gelingt ebenfalls nicht mit Bordmitteln.

Und schließlich endet die Macht von VBA auch beim programmierten Erstellen von *SmartArt*-Grafiken. Die gehören zwar zu den interessantesten Excel-Neuheiten, haben aber (genau wie die Multifunktionsleiste) keinen Eingang in das Objektmodell des Kalkulationsprogramms gefunden und lassen sich ausschließlich per XML-Code generieren.

Visual Studio Tools for Office

Wer sich nicht mehr mit den geschilderten und vielen weiteren Beschränkungen von VBA abfinden will (oder kann), hat derzeit nur eine Alternative, und die heißt Visual Studio Tools for Office oder kurz VSTO. Was es damit auf sich hat, wird ausführlich in Abschnitt 15.7 beschrieben.

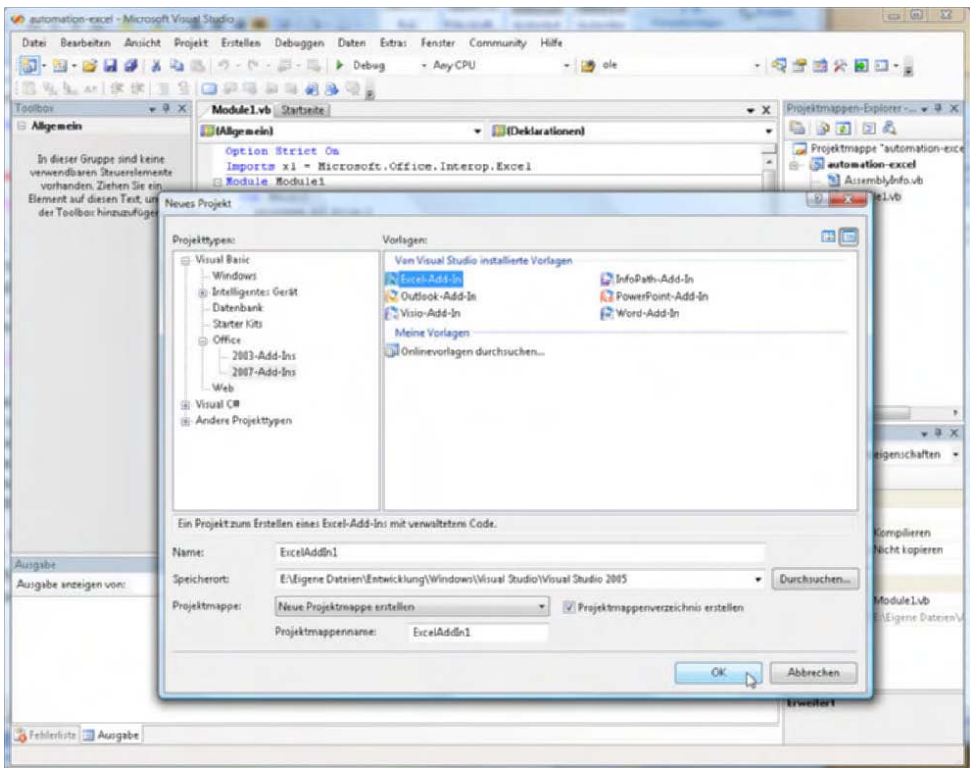


Abbildung 2.9: Die Visual Studio Tools for Office sind eine professionelle, aber auch teure Alternative zu VBA.

2.4 Probleme und Inkompatibilitäten

!!! ACHTUNG

*Auch in Excel 2007 kann es beim Bearbeiten beziehungsweise Testen von VBA-Code zu Abstürzen kommen. Speichern Sie regelmäßig! Beachten Sie auch, dass Excel nicht immer vollständig abstürzt. Manchmal kommt zwar die obligatorische Systemfehlermeldung, Excel bleibt aber im Speicher und blockiert weiter alle zuletzt geöffneten Dateien (ohne dass aber eine Möglichkeit besteht, diese noch zu speichern). Damit Sie wieder richtig weiterarbeiten können, müssen Sie Excel ganz beenden. Dazu verwenden Sie den Task-Manager von Windows XP/Vista, den Sie mit **Strg** + **Alt** + **Entf** zum Vorschein bringen. Es erscheint eine Task-Liste, aus der Sie Excel auswählen und gewaltsam stoppen können.*

Allgemeine Probleme

- ▶ VBA kennt noch immer keine Optimierungen bei der Auswertung von Bedingungen: Eine Abfrage in der Form $\text{If } x \geq 0 \text{ And } \text{Sqr}(x) < 3$ führt bei negativen Zahlen in x zu einem Fehler. (Dieses Problem besteht in Visual Basic schon seit der ersten Version, d.h., wir haben die Hoffnung auf Besserung auf diesem Punkt aufgegeben.)
- ▶ Bei allen Excel-Versionen gibt es Probleme mit dem Operator *Is* zum Objektvergleich. Dieser Operator sollte feststellen, ob zwei Variablen auf dasselbe Objekt verweisen. Leider funktioniert das nicht immer.
- ▶ Der Wechsel von der VBA-Entwicklungsumgebung zu Excel klappt nicht, wenn gerade der Objektkatalog das aktive Fenster ist. Sie müssen in der Entwicklungsumgebung zuerst ein anderes Fenster anklicken.
- ▶ Wenn Sie eine Logitech-Radmaus verwenden, funktioniert das Mousrad in der VBA-Entwicklungsumgebung nur, wenn Sie die zugehörige Maus-Software installieren. Warum das Mousrad bei allen anderen Programmen auf Anhieb (ohne Software-Installation) funktioniert und nur im VBA-Editor den Dienst verweigert, ist rätselhaft geblieben.

Probleme mit MS-Forms-Dialogen bzw. -Steuerelementen

Excel und die meisten VBA-Kommandos sind blockiert, solange sich der Eingabecursor in einem MS-Forms-Steuerelement in einem Tabellenblatt befindet. Nur bei Buttons kann das durch `TakeFocusOnClick=False` verhindert werden. (Die Defaulteinstellung lautet allerdings `True` und ist der Grund, weswegen es mit Buttons in Tabellenblättern oft Probleme gibt. Die auftretenden Fehlermeldungen sind ohne jede Aussagekraft.)

Wenn im Tabellenblatt auch andere Steuerelemente verwendet werden, muss der Eingabecursor per Programmcode (etwa durch *Worksheets(n).[A1].Activate*) in eine Zelle gesetzt werden, um damit sicherzustellen, dass er nicht auf ein Steuerelement gerichtet ist.

Generell bereitet die Verwendung von Steuerelementen in Tabellenblättern (statt in Formularen) enorme Probleme und löst mitunter nicht nachvollziehbare Fehler und zum Teil sogar Excel-Abstürze aus. Betroffen von diesen Problemen ist insbesondere die Beispieldatei 07\userform.xlsm, aus der bei einer zurückliegenden Neuauflage dieses Buchs einige Beispiele entfernt werden mussten; diese Beispiele funktionierten unter Excel 2000 noch problemlos, verursachten unter späteren Excel-Versionen aber Abstürze.

Inkompatibilitäten gegenüber Excel 2003

Der Wechsel von Excel 2003 auf Excel 2007 bringt zwar grundlegende Änderungen bei der Bedienung des Kalkulationsprogramms mit sich, aber nur wenige Unterschiede in Sachen VBA. Und so konnten wir die meisten Beispieldateien ohne Probleme aus der letzten Auflage dieses Buches übernehmen – nach der notwendigen Konvertierung in die neuen Open-XLM-Dateiformate und einigen inhaltlichen Aktualisierungen, versteht sich.

Von diesem Befund gibt es nur zwei Ausnahmen. Die erste betrifft das Web-Services-Beispiel aus Kapitel 15, das dem Server trotz korrekter Programmierung mitunter keine Ergebnisse entlockt und Excel in seltenen Fällen sogar zum Absturz bringen kann. Microsoft betrachtet das verwendete Web Services Toolkit zwar als ein »*not supported product*«, wird aber dennoch wohl in absehbarer Zeit eine aktuellere und auf Office 2007 zugeschnittene Version auf den Markt bringen, bei der das Problem hoffentlich nicht mehr auftaucht.

Das zweite Problem betrifft die Beispieldatei Chart.xlsm aus Kapitel 10, wo beim Anlegen eines Tagesprotokolls keine Diagramme erzeugt werden. Beim Debugging zeigt sich, dass Excel 2007 an verschiedenen Codestellen den Fehler 9 generiert, obwohl die monierten Anweisungen unter Excel 2003 fehlerfrei laufen. Hier scheint ein generelles Excel-Problem vorzuliegen, was möglicherweise durch ein späteres Programm-Update behoben werden wird.