

Einführung: Ein Buch der anderen Art...

Ich habe Jahre darauf gewartet, ein Buch über die COM/ActiveX-Technologie schreiben zu können.

Microsoft hat wahrlich sehr lange gebraucht, sich auf einen Namen für diese Technologie festzulegen. Mal hieß sie OLE, mal OLE-Controls, und die Bezeichnungen VBX oder Visual Basic Custom Controls waren früher auch zu hören. ActiveX war der »politisch korrekte«, gültige Begriff, als die erste Ausgabe dieses Buches erschien, doch mittlerweile hat sich COM weitaus mehr eingebürgert. Das tut aber alles nichts zur Sache. Ich habe trotzdem mit dieser Technologie gearbeitet, während sie sich weiterentwickelte. Ich wartete nur noch darauf, daß die Zeit reif für ein Buch zu diesem Thema wäre. Schlechthin wartete ich nur noch darauf, daß es mit Visual Basic möglich werden würde, Controls auf der Grundlage dieser Technologie erstellen zu können.

Als ich die Beta zu Visual Basic 5 zu Gesicht bekam, wußte ich, daß die Zeit des Wartens vorüber sein würde. Ich überlegte, wie ein Buch aussehen müßte, das einerseits höchsten Nutzen für Visual Basic-Programmierer bieten sollte, sich andererseits aber von der Vielzahl der zu erwartenden Bücher zum Thema ActiveX in den Regalen des Buchhandels unterscheiden könnte. Wie sollte ein Buch aussehen, das sowohl genügend Details für fortgeschrittene Programmierer enthalten als auch genügend Raum zur Einführung in die ActiveX-Entwicklung für Visual Basic-Einsteiger bieten würde?

Schließlich schlüpfte ich in meine Programmiererrolle und führte mir vor Augen, was ich an Technologie-Büchern mag und was nicht. Unterm Strich kam ich zu dem Ergebnis: Ich mag keine Nacherzählungen von Handbüchern.

Ein simples Zitieren der Microsoft'schen Dokumentation halte ich für witzlos. Ich denke, ein großer Teil derselben ist klar und verständlich. Ein Autor sollte dagegen eher mit neuem Stoff ankommen und möglichst eine neue, kreative Sicht auf eine Technologie bieten, statt lediglich die Handbücher nachzubeten. Außerdem hasse ich es, mich durch jede Menge bekannter Dinge hindurchzuwühlen, nur um hier und da ein neues Fitzelchen an Information ausfindig zu machen. Sie kennen sicher diese Art Bücher, von denen hier die Rede ist. Selbst in Werken für Fortgeschrittene wird einem da noch bis ins Detail erklärt, wie man Controls auf einem Formular anlegt, wie man mit der Maus herumklickt – und womöglich auch noch, wie man seinen Computer überhaupt einschaltet...

Ich war mir darüber im Klaren, daß ich ein umfassendes Buch über ActiveX und die Objekt-Programmierung mit Visual Basic schreiben wollte. Ich war mir sicher, nicht viel Zeit auf eine Nacherzählung der Dokumentation zu verschwenden. Sie werden es mir kaum glauben, aber ich halte die heutige Visual Basic-

Dokumentation für gar nicht mal so schlecht. Ich habe vielleicht gut reden, aber beim Lesen der Dokumentation zu Visual Basic 5 stellte ich fest, daß sie sogar perfekt wäre, wenn sie doch nur...

- ... der Erklärung einiger weniger Grundkonzepte mehr Raum widmen würde statt deren Verständnis vorauszusetzen.
- ... einige bestimmte Aufgaben mit mehr Beispielen illustrieren würde.
- ... einen guten Geist im Hintergrund bieten würde, der einem ein wenig unklar bleibende Passagen näher erläutern und der beim Lesen auftauchende Fragen beantworten könnte.
- ... über das reine Beschreiben einiger Vorgänge hinausgehen und erläutern würde, warum dies und jenes besonders wichtig ist, und auch hier und da mehrere Möglichkeiten anführen würde, die zum gleichen Ziel führen.
- ... mehr Informationen für fortgeschrittene Programmierer enthalten würde – wie etwa die Technologie hinter den Kulissen funktioniert, oder wie man beispielsweise ein eingebautes Windows-Control subclassen kann.

Als ich dieser Unzulänglichkeiten gewahr wurde, wußte ich, was ich zu tun hätte. Ich würde keine *ActiveX-Bibel zu Visual Basic* schreiben. Das hat Microsoft bereits mit der Dokumentation erledigt. Ich würde einfach den Kommentar dazu schreiben!

Ein talmudischer Weiser namens Maimonides schrieb bereits im Mittelalter ein Werk mit dem Titel *Guide to the Perplexed* (*Anmerkung des Übersetzers: so lautet der tiefsinnige englische Untertitel der Originalausgabe*), das ich mir zum Vorbild machte. Darin versuchte er nicht, die Aussagen der alten Schriften zu ersetzen, sondern sie zu erforschen, sie zu interpretieren, und dem einfachen Volk verständlich zu machen. Ich bin zwar kein Weiser dieses Kalibers, hoffe aber dennoch, den richtigen Ansatz gefunden zu haben.

Worum geht es in meinem »Guide to the Perplexed«?

Beim Schreiben dieses Buches habe ich die folgenden Ziele anvisiert und Philosophien verfolgt:

Konzepte

Es gibt eine Reihe grundlegender Konzepte, die ein Programmierer verstehen muß, wenn er COM-Komponenten schreiben will – die er erst recht verstehen muß, wenn er sie in Visual Basic schreiben will. Die Absicht ist, diese Konzepte tieferschürfend zu behandeln, und zwar so, daß auch Visual Basic-Einsteiger sie verstehen können. Darum geht es in Teil I.

Zielgruppe

Ich konzentriere mich auf Informationen, die sich auf die COM-basierten Features der aktuellen Visual Basic-Versionen beziehen. Wenn Sie mit Visual Basic noch gar nicht vertraut sein sollten, ist dies vielleicht nicht gerade das Buch, mit dem Sie beginnen sollten. Ich setze nämlich voraus, daß Sie Visual Basic kennen und die allgemeine Syntax der Sprache beherrschen.

Das Buch richtet sich in erster Linie an diejenigen, die sich bisher noch nicht näher mit den neuen objektorientierten Konstrukten (wie etwa Klassen) befaßt haben, wie sie mit Visual Basic 4.0 eingeführt worden waren. Es richtet sich auch an diejenigen, die noch nicht mit der OLE- bzw. der COM-/ActiveX-Technologie beschäftigt haben.

Ansatz

Ich gehe davon aus, daß Ihnen die Visual Basic-Dokumentation zur Verfügung steht. Dieses Buch soll diese unterstützen, jedoch nicht ersetzen. Auch wenn sich hier und da notwendigerweise Überschneidungen ergeben, liegt der Schwerpunkt darauf, in folgenden Punkten über die Dokumentation hinauszugehen:

- Interpretation und Klärung von unklar bleibenden Punkten und sie in verständliche Sprache übersetzen.
- Illustrative Darstellung und zusätzliche Beispiele wo es angebracht erscheint. Aufzeigen von Alternativen zur Implementierung diverser Features. Hinweise auf Abstürze bei inkorrekt implementierter Implementierung.
- Erforschung und Demonstration von Techniken, die in der Dokumentation nicht behandelt werden. Erläuterungen, wie Teile der Technologie im Hintergrund arbeiten.
- Kommentare zu und Diskussionen über die Art und Weise, wie Sie programmieren *sollten*, nicht wie Sie es *könnten*. Erklärungen dazu, warum einiges so und nicht anders gemacht wird, und hier und da werde ich auch meinen eigenen Senf hinzugeben...

Stil

Das gesamte Buch ist wie diese Einführung gehalten, in der Ich-Form. Ein Kommentar ist schließlich ein wenig mehr als nur eine kompetente Präsentation von akkurater technischer Information. Er enthält eben auch Interpretationen und Meinungen. So wie ich einerseits jede Anstrengung unternehmen werde, das Buch technisch so akkurat und verhältnismäßig umfassend zu halten, wird es darüber hinaus auf persönlicher Meinung und auf gewissen Voreingenommenheiten beruhen. Ich werde Ihnen also nicht nur Features vorstellen, wie und warum sie funktionieren, sondern ich werde Ihnen auch sagen, was ich jeweils davon halte und ob Sie es verwenden oder besser andere Ansätze verfolgen sollten.

Falls Sie auf irgendwelche technischen Ungenauigkeiten stoßen sollten, wäre ich Ihnen für eine Mitteilung per E-Mail (dan@desaware.com) dankbar, damit ich bei der nächsten Ausgabe entsprechende Korrekturen vornehmen und die Aktualisierung auf unserer Web-Site unter www.desaware.com vorstellen kann. Sollten Sie mit einer meiner Interpretationen oder Meinungen nicht übereinstimmen, können Sie mir auch gerne schreiben – ich werde Ihre Nachricht sorgfältig lesen und die Gegenansicht gegebenenfalls veröffentlichen oder einfach löschen, je nach meiner Lust und Laune...

Umfang

Sie werden wahrscheinlich festgestellt haben, daß Visual Basic ein »dickes Ding« ist. Es ist seit der ersten Version im Umfang mächtig gewachsen, sowohl in Sachen Datei-Größen als auch Leistungsfähigkeit. Damit das Buch in einer angemessenen Zeit fertiggestellt werden konnte, mußte ich mich entscheiden, welche Spannweite der Thematik es abdecken sollte. Mit Sicherheit werde ich jede Menge E-Mail bekommen, die das Fehlen des einen oder anderen Features, das zu einem perfekten Buch gehören müßte, beklagen werden. Ich werde Ihnen also verraten, was ich warum in den Inhalt aufgenommen, und was ich warum weglassen habe.

Was drin ist, und warum es drin ist...

Enthalten	Begründung
Grundlegende Konzepte zu ActiveX	Weil ich denke, daß ein Experte nur derjenige ist, der eine Materie »von der Pike auf« kennt. Seit der Einführung von Klassen und ActiveX-Technologie in Visual Basic ist ein Verständnis der der ActiveX-Technologie zugrundeliegenden Konzepte für jeden Visual Basic-Programmierer von großer Bedeutung.
Klassen	Der Umgang mit Klassen ist nunmehr ein wesentlicher Bestandteil der Programmierung mit Visual Basic. Viele Visual Basic-Programmierer sehen das jedoch noch nicht so. Meine Absicht ist es, einen Sinneswandel herbeizuführen.
Code-Komponenten	Mit der Möglichkeit, seit Visual Basic 5 multithreaded Objekte (OLE-Server, Ereignisse, EXE-Server) erstellen zu können, sind derartige Komponenten interessanter als je zuvor geworden. Sie bilden auch die Grundlage für das Verständnis von ActiveX-Controls und ActiveX-Dokumenten.

Enthalten	Begründung
ActiveX-Controls	Ich glaube ernsthaft, daß dies eines der wichtigsten Features von Visual Basic darstellt (ein anderes ist die Kompilierung in Native Code). Ich glaube, daß nun, da es möglich ist, Visual Basic-Programmierer dies auch tun wollen.
IIS-Anwendungen	Dieses in Visual Basic 6 neue Feature ist, so denke ich, das wichtigste neue Feature der Sprache für diejenigen, die an Server-Programmierung für das Internet interessiert sind. Diese Ausgabe enthält zwei neue Kapitel, die sich mit dieser Technologie befassen.

Was nicht drin ist, und warum es nicht drin ist...

Fehlend	Grund
Die Visual Basic-Entwicklungsumgebung	Der Umgang mit der Visual Basic-Entwicklungsumgebung ist nicht besonders schwierig und ist in der Microsoft-Dokumentation recht gut beschrieben. Ich werde auf dieses Thema nur eingehen, wenn es Einstellungen für ActiveX-Features betrifft.
Assistenten	Die eigentliche Idee der Assistenten besteht in der Vereinfachung von Vorgängen, so daß die Beschreibung der Verwendung eines Assistenten ziemlich überflüssig erscheint. Ich werde jedoch einen Blick auf den Code werfen, den Assistenten produzieren (ein hochinteressantes Thema), und darauf eingehen, wann Sie Assistenten einsetzen sollten und wann nicht.
Remote Automation und DCOM	Daß diese Themen überhaupt nicht behandelt würden, stimmt nicht ganz. Tatsächlich werden Sie, so hoffe ich, einen Einblick in die dahinterstehenden Konzepte bekommen. Die abertausenden Details der korrekten Konfiguration zum Einsatz dieser Technologien in der Praxis werden jedoch fehlen. Sicherheitsbelange und die Ausführung der diversen Konfigurations-Programme würden den Rahmen dieses Buches sprengen.
Microsoft Transaction Server	Auch hierzu werden Sie wiederum ein paar Informationen finden, vor allem wenn es um Kern-Themen dieses Buches wie die Lebensdauer von Objekten geht. MTS verlangt jedoch an sich nach einem eigenen Buch. Dieses Buch hingegen kann Sie lediglich in die richtige Richtung führen.

Fehlend	Grund
Sprache, Controls und alles weitere	In diesem Buch geht es um ActiveX-Technologie. Es gibt Dutzende anderer Visual Basic-Bücher zu vielen weiteren Themen. Wenn Sie beispielsweise der Zugriff auf das Win32 unter Visual Basic interessiert, sollten Sie zu meinem Buch <i>Visual Basic Programmer's Guide to the Win32 API</i> greifen.

Das gesamte Buch beruht auf der Visual Basic Enterprise-Edition. Es gibt jedoch nur wenige Punkte, die nicht auch unter der Professional-Edition funktionieren würden. Die Einschränkungen, die die Standard-Edition betreffen, können Sie in der Visual Basic-Online-Hilfe nachschlagen.

Schlußkommentar

Als ich meinem Verleger meinen Wunsch vortrug, ein technisches Buch zu schreiben, das als Kommentar zur aktuellen Dokumentation gelten und vollständig in Ich-Form geschrieben werden solle, stimmte er mir zu. Was sollte er auch sagen? Das Ergebnis halten Sie in den Händen. Das positive Echo auf die erste Ausgabe des Buches hat mich sehr erfreut. Ich hoffe, daß Sie die vorliegende, aktualisierte Ausgabe ebenfalls nützlich und vielleicht auch ein wenig unterhaltsam finden werden.

Nun möchte ich noch ein paar Punkte ins rechte Licht rücken, um mögliche Mißverständnisse zu vermeiden.

Microsoft

Zum Mitschreiben: Ich mag Microsoft. Ich glaube, daß sich Microsofts Erfolg zum größten Teil nicht auf die (legendären) hauseigenen Juristen gründet oder allein auf Marketing (was andere genauso können), sondern weil dort einfach gute Software entsteht. Großartige Software – wie eben Visual Basic.

Ich weiß, daß das Einprügeln auf Microsoft derzeit ein beliebtes Spiel ist. Viele Leute hoffen inbrünstig, daß Microsoft vernichtet oder zumindest aufgeteilt werden möge. Ich hege keine derartigen Gefühle. Trotzdem werden Sie in diesem Buch hier und da ein paar Seitenhiebe auf Microsoft oder auf einige Scherze auf deren Kosten treffen. Lassen Sie mich das also einmal klarstellen. Ich führe keinen Untergrundkampf gegen Microsoft. Ich will Microsoft auch nichts böses. Es ist nur so, daß es einfach manchmal Spaß macht, diesem riesigen, schwammigen Gebilde ein paar Nadelstiche zu versetzen. Mehr steckt wirklich nicht dahinter.

So etwa: Frage: Wie viele Microsoft-Programmierer werden zum Auswechseln einer kaputten Glühbirne benötigt? Antwort: Kein einziger – Microsoft definiert einfach Dunkelheit als neuen Standard.

Oder nehmen Sie folgende wahre Geschichte: Ein Programmierer fragte mich einmal, wie man sich denn als Software-Anbieter fühle, wo doch Microsoft der größte Fisch im Becken wäre. Meine Antwort darauf: Microsoft ist gar nicht der größte Fisch im Becken – Microsoft *ist das Becken!*

Freie Software

Viele Bücher werben damit, daß Megabyte-weise »freie Software« mitgeliefert werden würde. Wenn Sie jedoch näher hinschauen, werden Sie sehr oft feststellen müssen, daß es sich lediglich um Shareware handelt, die registriert werden muß, oder um Demos, »Cripple-Ware« (gegenüber der Vollversion abgespeckt), oder um sonstige, nicht-unterstützte (»unsupported«) Software. Fragen Sie mich bitte nicht nach dem Preis, den Sie letztlich dafür zu zahlen haben – nicht-unterstützte Software ist bei professionellem Einsatz auf lange Sicht immer zu teuer. Daher enthält dieses Buch keine »freie Software«.

Aber keine Bange – Sie werden auf der CD-ROM zu diesem Buch eine ganze Menge Software vorfinden. Diese dient jedoch nur als Lehrmaterial zur ActiveX-Programmierung. Vielleicht werden Sie entdecken, daß das eine oder andere nützlich für Ihre eigenen Anwendungen sein könnte und es somit für Ihre eigenen Zwecke adaptieren. Das wäre schön. Betrachten Sie das als eine Zugabe zum Buch. Doch suchen Sie eben bloß nicht nach freier Software. Der Grund für den Kauf dieses Buches sollte schließlich Ihr Informationshunger nach lehrreichem Stoff sein.

Desaware

Ich bin Präsident einer kleinen Software-Firma namens Desaware Inc. Dieses Unternehmen gründete ich 1991, als ich Visual Basic 1 sah und es für die Zukunft der Windows-Programmierung hielt. Ich denke, daß mir die seitherige Entwicklung rechtgegeben hat.

Sie werden hier und da auf Hinweise zu Desaware-Tools und -Produkten stoßen. Dafür gibt es zwei Gründe. Desaware ist auf ActiveX-Controls und -Tools zur Erweiterung von Visual Basic spezialisiert. Mit diesen können Sie Windows unter Visual Basic vollends ausreizen. Da ich unsere eigenen Tools und unseren eigenen Code bei meinem ActiveX-Anliegen verwende, sollte es Sie nicht überraschen, wenn ich mir einiges davon ausgeborgt habe, um einige der in diesem Buch beschriebenen Probleme lösen zu können. Ich hoffe, daß Sie nach der Lektüre dieses Buches einen Blick auf die Desaware-Produkte und Demos werfen werden, die auf der Buch-CD enthalten sind, und diese für wertvoll genug halten, in Ihr eigenes Repertoire an Hilfsmitteln aufgenommen zu werden. Sie sind es bestimmt Wert.

E-Mail

Ich bekomme eine Menge E-Mail. Darunter sind viele Fragen, die sich auf das Win32-API und auf die COM-Programmierung beziehen. Der Bogen spannt sich von kurzen Fragen zu einfachen Themen bis hin zu mehrseitigen Bitten um Problemlösungen, Fehlerbeseitigungen, Analysen und Begutachtungen von kompletten Anwendungen.

Ich wünsche, ich hätte die Zeit, jede einzelne eingehende Frage lesen und beantworten zu können. Doch ich habe sie nicht. Ein Tag hat nunmal nur vierundzwanzig Stunden. Wenn ich eine Frage erhalte, kann ich sie oft gleich aus dem Kopf heraus beantworten. Wenn nicht, versuche ich zumindest, dem Fragesteller die Richtung zur Lösung seines Problems zu weisen. Aber für mehr reicht meine Zeit wirklich nicht. Falls Sie also keine Antwort oder Reaktion auf Ihre E-Mail von mir erhalten sollten, möchte ich mich dafür bereits im voraus entschuldigen – sie würden staunen, wenn Sie wüßten, wieviel Zeit die Beantwortung auch nur einer simplen Frage kostet. Und wenn ich versuchen würde, auf alle E-Mail zu reagieren, bliebe mir für nichts anderes mehr Zeit. Das gleiche gilt, so leid es mir tut, auch für Anfragen, die an Desaware gesendet werden – wir sind stolz auf unseren qualitativ hochwertigen Support zu unseren Produkten, doch können wir weder zu diesem Buch noch zu Visual Basic einen Gratis-Support leisten.

Aktualisierungen und Berichtigungen

Ich wünschte, ich könnte Ihnen versprechen, daß dieses Buch in jeder Hinsicht fehlerfrei wäre. Ich kann Ihnen aber auch nicht das Gegenteil zusichern. Es wird sicher Fehler, Auslassungen oder Satzfehler geben. Diese treten immer auf, egal wie oft ich auch Korrektur lese und wie viele Leute die einzelnen Kapitel gegenlesen.

Wenn Sie auf ein Problem stoßen sollten, lassen Sie es mich wissen (auf höfliche Weise wäre nett – ich werde Korrekturen jedoch in jedem Fall berücksichtigen, auch wenn sie unhöflich daherkommen sollten, was öfter vorkommt, als Sie vielleicht glauben). Die neuesten Korrekturen und Aktualisierungen finden Sie auf unserer Web-Site unter www.desaware.com.

Das wär's nun in Sachen Einführung. Nun geht es weiter mit dem eigentlich, guten Stoff.

Daniel Appleman