

# VBScript Handbuch



VBScript Handbuch, Ch. J. Goddard, M. White, franzis-Verlag, , D-85622 Feldkirchen 1997, ISBN 3-7723-6963-4, 540 Seiten, mit CD.

Martin Weissenböck

**W**as ist VB Script? Der Zugang zum Internet wird bequemer und die Internet-Browser werden leistungsfähiger, wenn nicht alle Daten zwischen Client (dem PC, mit dem die Internet-Verbindung aufgebaut wird) und dem Server (dem Rechner, der angesprochen wird) ständig ausgetauscht werden müssen und statt dessen Daten vor Ort, also am lokalen PC, vorverarbeitet werden. Damit werden viele Effekte möglich: logische Kontrolle der Daten vor Ort, Animation von Seiten usw. Daher werden immer mehr Programmierfunktionen in die Browser eingebaut. In den Netscape-Produkten sind es Java und JavaScript. Microsoft hat für den Internet-Explorer zusätzlich das bewährte Visual Basic adaptiert: VB Script heißt das Wunderding, das natürlich mehr können soll, als alle seine Vergleichsprodukte.

In insgesamt 15 Kapiteln werden folgenden Themen behandelt: VB Script (Überblick) - Internet Explorer (Version 3) - Erstellen eines VB Script-Dokuments (Verbindung zu HTML-Elementen) - VB Script-Sprachelemente (Variablen, Arrays, Datentypen, Ausdrücke, Operatoren, Konstanten, Verzweigungen, Schleifen, Unterprogramme, Fehlerbehandlung) - Hinweise zum Programmaufbau - Das Objektmodell - Formulare - ActiveX-Controls - Internet Control Pack - CGI, ISAPI, Web-Server.

Im Anhang ist eine gute gegliederte Übersicht über die VB Script-Sprache zu finden.

Auf der beiliegenden CD sind unter anderem zu finden (Zitat):

„Cool Edit: ein voll ausgestatteter Wave-Editor für Windows ... Frame-IT!: ein mit Besonderheiten vollgepackter HTML-Frame-Generator. HotDog: ein selbständiger HTML-Editor. ... Microsoft ActiveX SDK: Tools, die Ihnen bei der Entwicklung von Internet-ActiveX-Controls helfen. Microsoft Internet Explorer 3.0. Paint Shop Pro: der populäre Bitmap-Editor von JASC, Inc. ... Vision Storyboard: ein Entwicklungswerkzeug, das beim Planen, Do-

kumentieren und Präsentieren von Microsofts Visual Basic-Applikationen hilft.“

Vom Aufbau und vom Inhalt her also eine sehr interessante Zusammenstellung. Der Leser sollte aber meiner Ansicht nach schon Erfahrung mit Visual Basic haben. Bei Vorkenntnissen könnte aber der eigentliche Sprachteil kürzer sein. Das Buch eignet sich trotzdem für interessierte Anfänger ebenso wie für Fortgeschrittene, die auf VB umsteigen wollen.

Soweit die guten Nachrichten. Nun die schlechten: das Thema VBScript ist offenbar so aktuell und das Bedürfnis nach neuen Büchern so groß, daß anscheinend keine Zeit mehr war, das Buch vor dem Drucken noch einmal auf Fehler und ähnliches hin durchzulesen. Auch die Tücken der Textverarbeitung haben an mehreren Stellen zugeschlagen. Hier einige Beispiele:

- Manche Textverarbeitungssysteme setzen automatisch typographische Anführungszeichen. Beispiel: „String“. Wird aber in einem Programm eine Zeichenkette (ein String) geschrieben, sollte der so aussehen: "String". Im VBScript-Handbuch schaut das dann oft so aus: "String". So kann ein Text aber nicht ins Programm eingegeben werden. Werden Texte von einer Textverarbeitung in eine andere übernommen, gibts immer wieder Konvertierungsfehler. Strings erscheinen dann plötzlich als òStringó (ab Seite 117). Ich kann mir vorstellen, daß das für einen Anfänger besonders verwirrend ist.
- Ein besonderer Leckerbissen für alle, die den ASCII, den Unicode, Windows-Zeichensätze und anderes auswendig wissen, ist folgende Zeile von Seite 459: „Das Resultat liegt zwischen - Symbol 112 \f "Symbol" \s 10 /2 und Symbol 112 \f "Symbol" \s 10 /2 rad.“ Kommt übrigens ein paar Mal vor.
- Auf Seite 492 fehlt dann das Pi endgültig: „Zur Umwandlung eines in Altgrad ... angegebenen Winkels in

das Bogenmaß *rad* multipliziert man den Winkel mit /180.

- Seite 464, zum Kosinus: „Die Winkelangabe *rad* kann in Altgrad umgewandelt werden, indem man ihn mit 180/ multipliziert.“ Erstens wäre die Umwandlung von Altgrad in Radiant an der Stelle wichtiger, zweitens ist *die Winkelangabe* auch nach der Rechtschreibreform feminin und drittens fehlt wieder das Pi.
- Übersetzen ist ja gut, aber die Operatoren „And“ und „Or“ sollten doch besser nicht zu „und“ und „oder“ werden (Seite 104 und 106).
- Im Verzeichnis der Funktionen werden abwechselnd Großbuchstaben und Groß/Kleinbuchstaben verwendet. Das spielt zwar in VBScript keine Rolle, ist aber sicher auch etwas verwirrend. Der Editor macht jedenfalls aus „EXP“ automatisch „Exp“ usw.
- Auf Seite 470 erfahren wir, daß „zum Beispiel &H8F den Binärausdruck 100001111" darstellt.
- Die Beschriftungen in den Abbildungen sind fast immer zu klein und nahezu unleserlich. Wenn Felder außerdem „dunkelgrau“ unterlegt sind, sollte das Ganze nicht fast schwarz sein.
- Ein besonderes Problem beim Abrufen von Programmzeilen in Büchern ist der Umgang mit dem Zeilenbruch. Sinnvollerweise wird in dem Buch das Zeichen ↵ verwendet, um eine Fortsetzungszeile zu kennzeichnen. So gesehen ist die folgende Zeile (Seite 169) bemerkenswert:

```
FAIL_CONNECT = 1001 : LOSE_CONNECT = 1002 :
```

```
CORRUPT_DATA =  
↵1003
```

Das heißt: zuerst werden mehrere Basic-Anweisungen in eine Zeile geschrieben, dann paßt der Text nicht und dann wird *noch eine* neue Zeile begonnen. Richtig wäre:

```
FAIL_CONNECT = 1001: LOSE_CONNECT = 1002: CORRUPT_DATA = 1003
```



# Hammer und Siegel-Schneebeesen



Microsoft® Anwender Training: Microsoft® Word 97  
 Training-Programmierung, Redmond Verlag GmbH&Co KG  
 D-85704 Unterschleißheim, ISBN 3-932575-34-2

Fritz Eller

“Stellen Sie sich vor, daß vor Ihnen ein Hammer und ein Schneebeesen liegen.

Beide haben vergleichbare Eigenschaften: einen Stiel (!) aus Holz und an einem Ende ein Gebilde aus Metall. Beide Objekte beherrschen aber völlig unterschiedliche Methoden. Der Hammer verfügt über die Methode “Hämmern” und der Schneebeesen über die Methode “Rühren”. Versuchen Sie die Methoden auszutauschen. Dann passiert ein Malheur bzw VBA wird Sie darauf aufmerksam machen, daß Sie etwas Unzulässiges vorhaben. Anders ist es mit den Eigenschaften eines Objektes. Sie können z.B. den Holzstiel (!) beider Objekte durch einen Plastikstiel oder auch das Metall durch ein anderes Material ersetzen. Das ändert nichts an der jeweiligen Methode, nach der ein Objekt arbeitet höchstens am Resultat.”

Alles klar. Eigentlich logisch, oder? Nach diesen Ausführungen kennt mann/frau sich aus! Mit Objekten, Eigenschaften und Methoden. Beim Hammer mit dem hölzernen Stiel und beim Schneebeesen, deren beider Stiele auch aus Plastik sein könnten.

In dieser Art gibt es einiges zu lesen im “Microsoft Anwender Training” zur Programmierung mit VBA unter Word 97.

Vorausgesetzt wird, daß der Anwender, der mit dem rund 160 Seiten dicken Buch im A4-Format arbeitet (zum

einfach dasitzen und lesen ist es nicht sehr geeignet), mit Word 97 gearbeitet hat und (vermutlich auch) die beiden aufbauenden Bücher aus der selben Reihe beherrscht. Klar, denn wer kommt sonst auf die Idee, mit Visual-Basic für Word Makro und kleine Programme zu fabrizieren.

Das obenstehende Zitat ist ein Ausschnitt aus dem Buch (S.27).

Der generelle Aufbau ist klar und sach-



lich:

Ein Ziel wird genannt, der Lösungsweg wird in mehreren Punkten dargestellt und der Programmcode wird als Ergebnis ausgegeben. Die oft lästige

Tipparbeit übernimmt keine beilegte Diskette oder CD, Handarbeit ist gefragt – dagegen ist ja auch nichts auszusetzen, schließlich häufen sich ja die beigelegten CDs anderer Bücher und diverser Zeitschriften ja schon rund um den PC des Autors.

Nachteilig sind nur die Details, die die Profischmiede des Redmond Verlages sich hier leistet.

Lösungswege werden zwar beschrieben, der Programmcode stimmt aber mit den Vorgaben nicht überein:

z.B. Erstellen Sie eine Prozedur VORLAGE, die dann im Ergebnis-Quellcode plötzlich Sub Statuszeile() heißt.

Ungenau arbeiten im Programmieren(wenn's auch nur Makros sind) ist eine Sache. In einem Ausbildungsskriptum eines renommierten Verlages, das ein interessierter Programmierereinsteiger aber im Selbststudium durchzukauen versucht, aber “Kleinigkeiten” wie Punkte und/oder Leerzeichen falsch/zuviel/zuwenig oder gar nicht setzt, kann schon zu Erstaunen, Selbstzweifeln und/oder Wutausbrüchen führen. Meist ist es ein Kombination von allem, besonders dann, wenn man erkennt, daß nicht einer selbst die Fehler macht, sondern die Redmond Verlag-Autoren schlampig, sehr schlampig gearbeitet haben.

Ist das denn der Stil (!) von Bill Gates?....

oder

```
FAIL_CONNECT = 1001
LOSE_CONNECT = 1002
CORRUPT_DATA = 1003
```

Daher sollten Setzer, Übersetzer und Lektor auch die Sprache beherrschen, damit falsche Zeilenumbrüche vermieden werden. Noch ein Tip an den Verlag: andere Schriftart verwendet!

- Das Flußdiagramm auf Seite 133 (Abbildung 6-5) ist etwas eigenartig und hat leider kein passendes Programmstück im Text.
- Die Abbildung 6-8 auf der Seite 140 sollte den Unterschied zwischen eine „Do While/Loop“- und einer „Do/Loop While“-Anweisung erläutern. Erstens wären mir Struktogramme auch lieber als Flußdiagramme (Programmablaufpläne). Zweitens steht zweimal „DO

WHILE...LOOP“ drinnen und nicht einmal „DO WHILE...LOOP“ und einmal „DO...LOOP WHILE“. Ich habe bis jetzt immer geglaubt, daß ich den Unterschied zwischen beiden Arten kennen, aber die Abbildung habe ich überhaupt nicht verstanden.

Wer über solche Probleme hinweg sehen kann, findet in dem Buch eine rasche Einführung in VB Script. Etliche Programme auf der CD sind recht nützlich.