

## **Stand von DoDi's VBTools (Ende Juli 95)**

Zunächst möchte ich mich bei allen registrierten Anwendern für die Verzögerung entschuldigen, die sich bei der Auslieferung der neuen Versionen ergeben hat. Sie erhalten die neuen Versionen immer noch vorab ohne Beta-Test, in der Hoffnung, daß nicht allzu viele Fehler übriggeblieben sind. Die nochmals überarbeiteten Versionen und die endgültige Dokumentation werden später nachgeliefert.

Während sich die Entwicklung der Entwicklungswerkzeuge (**VBPro, VBMap, VBDiff**) auf wenige Verbesserungen beschränkt, ist die Entwicklung des Discompilers (**VBDIs**) so stürmisch, daß noch keine endgültige Version in Aussicht ist. Daher habe ich mich entschlossen, zunächst eine Zwischenversion des Discompilers zu verteilen, damit das Warten auf die nächste Version nicht zu lang wird. **VBDIs** hat Nachwuchs bekommen, die neue Version **VBMDIs** unterstützt speziell die Optimierung von VB-Programmen.

Leider hat die Dokumentation wie üblich mit der Entwicklung noch nicht Schritt halten können, daher hier die **letzten Änderungen** in Kürze:

### **Allgemeines**

Die **Installation** der Programm ist so einfach, daß hierfür kein Setup-Programm erstellt wurde. Für den Discompiler empfiehlt sich ein eigenes Verzeichnis, die übrigen Tools können Sie direkt in Ihr VB-Verzeichnis installieren. Wenn Sie Archive erhalten haben, entpacken Sie diese im gewünschten Verzeichnis, andernfalls kopieren Sie die Dateien oder die ganzen Verzeichnisse auf Ihre Platte.

Da die Tools zur Entwicklung von VB-Programmen dienen sollen, wird ein korrekt **installiertes Visual Basic 3.0** vorausgesetzt. Alle Programme benötigen das Laufzeitsystem (VBRUN300.DLL) und die Controls GRID.VBX und SPIN.VBX, die zu VB gehören und daher auch **nicht mitgeliefert** werden. Bei Problemen prüfen Sie bitte die Dateien VB.INI im Windows-Verzeichnis und AUTOLOAD.MAK im VB-Verzeichnis, ob die darin enthaltenen Pfadnamen korrekt sind.

Wenn Sie gleichzeitig **Visual Basic 4.0** installiert haben, können weitere Anpassungen notwendig werden. In VB.INI wird der Eintrag **vbpath=** ausgewertet und dort die Datei AUTOLOAD.MAK gesucht. Daraus wird der bei der Installation angegebene Pfad der VBX-Dateien ermittelt, dies funktioniert bei VB 4.0 nicht mehr. Ebenso funktioniert dann die Erzeugung der Makefiles nicht mehr richtig, da die Makefiles in 4.0 ein völlig neues Format haben. Bei größeren Problemen werden alle Funktionen gestrichen, die auf VB.INI und AUTOLOAD.MAK basieren, oder (ungern) durch Installations-Optionen ersetzt.

Vereinzelt wurden unter **Windows 95** Fehler beim Starten der Programme beobachtet, hier kann noch keine Abhilfe geschaffen werden. Zur Diagnose der Fehler wurde (speziell beim Discompiler) eingebaut, daß die Startform sofort nach dem Starten des Programms angezeigt wird. Wenn daher Fehlermeldungen kommen bevor die Form dargestellt wird, liegt dies am Zusammenspiel zwischen Visual Basic (d.h. der VBRUN300.DLL) und Windows 95, möglicherweise auch an einer unvollständigen Installation von Visual Basic (obwohl dies auf die Ausführung der Programme eigentlich keinen Einfluß haben sollte).

Nach dem Erscheinen der Form wird (beim Discompiler) protokolliert, auf welche Dateien bei der Initialisierung zugegriffen wird. Möglicherweise tritt dabei ein Mangel von VB zutage, da nach Auskunft von Microsoft unter nicht näher spezifizierten Umständen die Ermittlung des Pfades der Anwendung nicht richtig funktioniert. Dann sollte jedoch eine entsprechende Fehlermeldung erscheinen 'VBRUN300.300 nicht gefunden' und ein Dialog zur manuellen Vorgabe des Verzeichnisses für den Discompiler angezeigt werden.

## Projekt-Manager VBPro

Bei der bedingten Erzeugung der Quellen werden nun Debug-Statements erkannt (Stop, Debug), die in den fertigen Programmen nicht vorkommen sollten. Enthält eine Zeile einen dieser Strings, kann die Zeile in den Quelltext übernommen oder weggelassen werden. Wird die Erzeugung abgebrochen, enthält die Quelldatei den Text bis zu der fehlerhaften Stelle, so daß diese im Originalcode leicht zu finden ist.

Im Grid wurde ein **Popup-Menü** eingebaut, das mit der **linken** Maustaste aufgerufen wird. Damit kann der Wert eines Feldes einfacher geändert werden, auch das Löschen einer ganzen Zeile ist möglich. Da die Verwendung der linken Maustaste technisch notwendig ist, aber damit die Einstellung der Zellengröße des Grids nicht mehr möglich ist, wird zusätzlich der Status der Zusatz Tasten (Shift, Control...) abgefragt. Im Script-Menü können Sie einstellen, ob das Popup-Menü erscheinen soll, wenn eine Zusatztaste gedrückt ist oder nicht. Die Voreinstellung ist, daß das Menü ohne Zusatz Tasten erscheint, somit müssen Sie zum Ändern der Zellengröße eine Zusatztaste drücken. Wenn im Menü 'Shift Popup' markiert ist, sind diese beiden Funktionen vertauscht.

Bitte beachten Sie, daß die bedingte Compilierung weiterhin Quellen benötigt, die **als Text** gespeichert sind.

Im Juni wurde der Schutz von Programmen gegen Discompilierung in Angriff genommen, die ersten Ergebnisse sind in den Projekt-Manager eingebaut. Dazu wird (vorläufig nur) eine Diskette erstellt, auf die das compilierte Programm kopiert und dabei modifiziert wird. Außerdem kann dabei eine Versions-Information eingefügt werden, die im Dateimanager von Windows für Workgroups unter Eigenschaften angezeigt und auch von den Installations-Funktionen (ver.dll) verwendet werden kann. Für die Verfolgung von Raubkopien wurde eine Anwender-Datenbank eingebaut, in die Sie alle registrierten Anwender für Ihre Programme eintragen können; aus dieser Datenbank wählen Sie beim Erstellen einer Diskette den Anwender aus, der das registrierte Programm erhalten soll, und diese Information wird beim Kopieren ebenfalls in das Programm eingebaut. Außerdem können weitere Dateien (Dokumentation, spezielle DLLs...) auf die Diskette kopiert werden, die während der Entwicklung nicht kopiert werden müssen.

Der beim Kopieren eingebaute Schutz der Programme wirkt sich zunächst so aus, daß der Discompiler mit einem Fehler abgebrochen wird, was nicht so elegant ist wie die bisherige Fehlermeldung, aber wirksamer. Weitere Änderungen sind in Arbeit, aber diese ganzen Maßnahmen sind eher kosmetischer Natur, es läßt sich nicht vorhersagen, wie ein anderer Discompiler darauf reagiert. Anzumerken bleibt noch, daß dieser **Schutz natürlich nur in der registrierten Version** des Projekt-Managers eingebaut wird.

Die Erzeugung **mehrsprachiger Versionen** eines Programms wird nun durch die Option Übersetzung im Datei-Menü unterstützt. Dabei werden alle Forms des Projekts nach Caption- und Text-Eigenschaften durchsucht und die gefundenen Controls und Texte in einer Tabelle aufgelistet. Dort tragen Sie neben den vorgefundenen Texten die entsprechenden Übersetzungen ein; gleichlautende Texte müssen nicht extra angegeben werden, die Texte werden dann unverändert übernommen. Solange die Option Übersetzung markiert ist, werden beim Kopieren der Quellen die übersetzten Texte in die Forms eingetragen. Die Tabelle wird unter dem Projektnamen mit der Erweiterung **trx** abgespeichert, Änderungen sind aber nur im Projekt-Manager möglich.

Die Übersetzungs-Funktion ist (vorläufig) **nicht** in der Shareware-Version (Lite) verfügbar!

Die Anpassung an verschiedene Graphik-Auflösungen muß weiterhin in getrennten Quellen erfolgen, da hierfür eine statische Änderung der Forms oft nicht ausreicht und die automatische Umrechnung der Koordinaten nicht einmal in Visual Basic funktioniert.

Rechtzeitig zum Erscheinen von VB 4.0 sollen auch Funktionen implementiert werden, die eine Umwandlung von Projekten von VB 3.0 auf 4.0 durchführen oder zumindest unterstützen, soweit dies notwendig ist.

### **Crossreference VBMap**

Hier wurde erstmalig die **Anzeige binär gespeicherter Module** realisiert. Mit einem Doppelklick auf das Feld mit dem Symbol- bzw. Modulnamen wird das entsprechende Modul angezeigt, egal ob es binär oder als Text gespeichert ist; als Erinnerung an diese Funktion ändert sich der Mauszeiger über diesem Feld. Die Anzeige der Module erfolgt ähnlich wie im Interpreter in einem geteilten Fenster, in dem zwei verschiedene Textstellen angezeigt werden können. Die Auswahl des Moduls und der Textstelle erfolgt über die zwei Combo-Boxen oben im Fenster, auf eine Trennung der Funktionen nach Objekten wurde dabei verzichtet. Teile des angezeigten Quelltextes können über das Clipboard in den Interpreter oder andere Programme übernommen werden.

Für große Projekte kann nun der **Umfang der Listen** vorgegeben werden. Dazu wurden zunächst einfache Buttons (Labels) oberhalb der globalen und lokalen Symboltabelle angeordnet, die folgende Einstellungen ermöglichen:

Bei den **globalen** Symbolen können Variablen, Konstanten, Typen und Unterprogramme ausgeblendet werden, mit '\*' werden wie bisher alle Symbole dargestellt.

Bei den **lokalen** Symbolen können nun wahlweise auch die globalen Definitionen eines Moduls angezeigt werden.

Die Auswahl der Optionen erfolgt durch Anklicken der Buttons, wobei abgewählte Optionen grau dargestellt werden.

### **Discompiler VBDis3 und VBMDis3**

Zunächst die schlechte Nachricht:

Aufgrund unerwarteter Eigenschaften großer Programme müssen weite Teile des Discompilers überarbeitet werden. Da dies noch einige Zeit dauern wird, wurde die aktuelle Version eingeschränkt. Sie prüft nun ab, ob in einem Programm gewisse Grenzen überschritten werden und verweigert dann die weitere Bearbeitung. Dies kann leider vor dem Erscheinen von VB 4.0 nicht mehr geändert werden und wird wohl in der Lite-Version als Einschränkung erhalten bleiben.

Doch nun zu den guten Nachrichten:

**VBMDis** erlaubt den Vergleich der Quelltexte mit dem ausführbaren Programm! Erzeugen Sie das ausführbare Programm und geben Sie VBMDis den Makefile an. Dann startet der Discompiler (wie in VBDis), danach erscheinen zwei Fenster mit den Modulen des Interpreters und des ausführbaren Programms. Die Fenster sind jeweils geteilt in die Anzeige des Textes und die Anzeige der Tokens für die aktuelle Zeile.

**Für die Anzeige der Tokens müssen die Quellen unbedingt binär gespeichert werden.**

Zusätzlich können die Variablen angezeigt und (wie die Unterprogramme) mit Namen versehen werden, auch die Zuweisung von Datentypen ist möglich, wenn die Variablen und Konstanten im Programm nicht verwendet wurden und der Discompiler daher die Typen nicht ermitteln konnte.

**VBCtrl** wurde auch überarbeitet und von überflüssigem Ballast befreit, und die Abspeicherung funktioniert nun auch in der Lite-Version. Die Beschreibungen werden automatisch gespeichert, wenn das Verweis-Fenster geschlossen wird.

Allerdings kann VBDIs (Lite) mit den im neuen Format gespeicherten Beschreibungen nichts anfangen.

Die Datenstrukturen des Discompilers wurden komplett überarbeitet, der Discompiler startet nun wesentlich schneller und sollte auch sonst etwas schneller geworden sein, weitere Optimierungen sind in Arbeit.

Wegen möglicher Fehler bei der Erzeugung von Forms im Binär-Format müssen diese weiterhin aus der binär gespeicherten Form und dem als Text gespeicherten Code zusammengesetzt werden. Beachten Sie bitte die dafür nötige **Vorgehensweise**:

1. den Discompiler starten, EXE-Datei und Verzeichnis für die Ausgabe auswählen
2. nach der Meldung 'Forms als Text speichern' den Interpreter mit dem erzeugten Makefile starten, **ohne** den Discompiler zu beenden!
3. im Interpreter alle **Forms als Text speichern** (Save As...)
4. den **Interpreter beenden** und im Menü des Discompilers wählen 'Forms zusammenfügen'

Erst dann ist das Projekt komplett erzeugt und kann im Interpreter gestartet werden.

Fehlermeldungen des Interpreters stammen normalerweise von nicht richtig oder garnicht ermittelten Variablentypen. In diesen Fällen suchen Sie weitere Vorkommen der Variablen, die Deklaration oder weitere Aufrufe desselben Unterprogramms mit bekannten Variablentypen. Externe Funktionen (Declare...) können teilweise aus der Hilfe zum API für VB entnommen werden, wobei mit Änderungen von festen Datentypen in 'As Any' zu rechnen ist und umgekehrt. Einige Funktionen mit nicht aussagekräftigen Ergebnissen können statt als Function auch als Sub deklariert sein, in diesem Fall (erkennbar am Aufruf des Unterprogramms) muß die Deklaration so bleiben.

Einige Custom-Controls verlangen eine spezielle Entwicklungs-Version bzw. -Lizenz des VBX oder zusätzliche DLLs wenn sie im Interpreter eingesetzt werden sollen. In diesen Fällen kann das Programm oft nicht gestartet werden, die entsprechenden Controls werden in den Forms durch rote Pictures ersetzt.

Zuletzt bleiben eigentlich nur noch Fehler beim Aufruf externer Funktionen zur Laufzeit übrig, die auf Fehler in der jeweiligen Deklaration zurückzuführen sind. In diesen Fällen müssen Sie mit den Parametertypen und 'ByVal' experimentieren.

Sollten weiterhin Probleme mit fehlenden Variablen oder Datentypen auftauchen, die aufgrund eindeutiger Merkmale eigentlich automatisch richtig zu ermitteln sein sollten, müssen vor der Erzeugung des Codes ggf. mehrere Scan-Durchläufe durchgeführt werden. In diesem Fall bitte ich um eine entsprechende Rückmeldung, dann wird ein Menü-Eintrag 'Projekt erzeugen' eingefügt, der eine nochmalige Analyse des Programms durchführt und dann hoffentlich bessere Quellen erzeugt.

### **Ausblick**

Der Schutz von Programmen gegen Discompilierung kann nicht mit einem öffentlich zugänglichen Tool verbessert werden, da sich dann die Änderungen beobachten lassen, die dieses Tool an den Programmen vornimmt und diese in einem beliebigen Discompiler entsprechend berücksichtigt werden können. Nur der Schutz gegen **VBDIs3** kann im Zuge der Optimierung eingebaut werden, obwohl auch dies die Gefahr des Bekanntwerdens dieser Maßnahmen bedeutet. Da nach den vorliegenden Erfahrungen die Namen aller Controls

problemlos aus den ausführbaren Programmen entfernt werden können, erhöht die Bearbeitung Ihrer Programme mit **VBPro** den Schutz gegen Discompilierung beträchtlich!

Vielleicht haben Sie eine Idee, wie Programme geschützt werden können, ohne daß jeder Interessent die Schutzmechanismen ausspähen und umgehen kann?

Für zweckdienliche Hinweise, Fehlermeldungen und Verbesserungsvorschläge bedankt sich der **Autor von DoDi's VBTools**

**Dr. H.-P. Diettrich**  
**Weißenburgstr. 2 c**  
**D-70180 Stuttgart**

**MausNet:** Hans-Peter Diettrich @ S  
**Fido:** über WINDOWS.VB.GER oder Hans-Peter\_Diettrich@s.maus.de (?)

Meine Adresse wird sich im Laufe des Jahres ändern, alle registrierten Benutzer werden - wie über andere Neuerungen auch - auf dem Laufenden gehalten.