Einstieg in die automatisierte Nutzung und in die Entwicklung von TurboCad - Anwendungen.

Erste Gliederung nach Dateitypen

Um Standardobjekte oder sich wiederholende Bearbeitungsschritte in TurboCad automatisiert einzufügen gibt es zwei TurboCad- interne Anwendungen.

1.1 Makroaufzeichnung

Der Makrorecorder zeichnet ab dem Start alle ausgeführten Bearbeitungsschritte auf bis dass er durch Stopp beendet wird. Diese Aufzeichnung kann gespeichert und auch wieder geöffnet werden um sie erneut auszuführen. Dies ist dann sinnvoll wenn man z.B. Formatierungen oder Eigenschaften für viele Objekte des Öfteren ändern möchte.

Diese Dateien haben die Endung (alt .tcm) .tcr

Im Makrorecoder können auch VisualBasic-Skript-Dateien geöffnet und ausgeführt werden.

Diese Dateien haben die Endung .vbs

1.2 Parametrischer Teile Manager

Die Erstellung eines Objektes erfolgt durch eine einfache geometrische Beschreibungssprache welche als Text eingeben wird. Parametrieteile eignen sich hervorragend für Variationen von Standardprodukten die dann in ihren Abmessungen verändert werden können.

Erstellte Parametrische Teile werden unter „Symbole und Teile“ angezeigt. Ein Vorschaubild ist natürlich sinnvoll und wird als .png hinterlegt. Das Schönste ist allerdings, dass alle Variablen unter Eigenschaften in einem Eingabefeld eingegeben werden können. (Bild: Anlage 20)

Mein Beispielprojekt dazu soll einmal ein Hängeschrank werden mit den Variablen B, H, T und den Positionsmaßen für Dübel, Rastex25 und 32 Lochreihen. Ebenso soll eine Rückwandnut mit B, T und Position variabel werden; auch die Anzahl der Schranktüren wird variabel.

Diese Dateien haben die Endung .ppm

Zusatzfunktionen oder Anwendungen werden extern erstellt und in TurboCad integriert oder per Fernzugriff direkt in TurboCad ausgeführt. Es gibt dazu in TurboCad zwei Zugänge:

1.3 dll – Schnittstelle

Dll´s sind Programmdateien die beim Start von TurboCad in speziellen Verzeichnissen gesucht und gelistet werden. Dll´s sind im Gegensatz zu com- oder exe-Dateien nicht eigenständig ausführbar, sondern benötigen einen Wirt (Host).

Diese Dateien haben die Endung .dll

1.3.1 AddOn´s - sind Zusatzfunktionen. Wie generell unter 1.3 beschrieben sind diese dll´s lediglich dem Standardverzeichnis AddOns von TurboCad zugewiesen. Bild: Anlage 4

1.4 Com- Schnittstelle

Über die Com-Schnittstelle greifen extern erstellte Anwendungen auf TurboCad zu. Während dll´s in TurboCad integriert sind, bietet der Zugriff über die Kommunikationsschnittstelle die Möglichkeit eines direkten bidirektionalen Fremdzugriffes; vorstellbar wie eine Fernsteuerung per Teamviewer©.

Diese Funktionalität ist implementiert in IMSIGXnn.dll

Wie erstelle ich nun diese Dateien?

2.1 Makro´s .tcr

Ein Makro wird im Normalfall TurboCad-intern mit dem gleichnamigen Recorder aufgezeichnet. Da man diese Makros speichern und wieder laden kann, ergibt sich auch die Möglichkeit diese gleich extern zu schreiben. Bild: Anlage 1, TurboCad Makro Editor

2.1.1 Skripte / Quelltexte können im Makrorecorder geöffnet werden –nicht zu empfehlen– .vbs

Sie können mit einem simplen Texteditor geschrieben werden. Zu nennen sind hier der einfache Windows Editor und die kostenlosen Editoren wie Notepad++ und PSPAD. In PSPAD können zudem die Skriptsprachen VB und VBScript zur Syntaxprüfung eingestellt werden. Dabei muss man natürlich selbst auf eine Konformität zum TurboCad-Sprachraum achten. PSPAD dokumentiert ebenfalls Funktions- und Prozedurköpfe.

In VisualBasic-Skripten sind Eingaben per Inputbox und Parameterübergaben möglich. Unter VBS erstellte Makros können im TurboCad-Makrorecorder geöffnet werden, jedoch auch extern gestartet werden. Dabei sollte jedoch eine TurboCad-Zeichnung geöffnet sein. Eine Abfrage ob dies der Fall ist oder eine TurboCad-Zeichnung geöffnet werden soll, kann im Skript geschrieben werden. Wegen der nicht überwachten Sprachkonformität im Makrorecoder ist hier VBS nicht empfehlenswert und führt ggf. zu Abstürzen.

2.2 Makro´s für Paramentrie-Teile .ppm

TurboCad intern kann der Quelltext im „Macroeditor für Parametrieteile“ eingegeben werden. Das ppm kann gespeichert und geladen werden. Damit kann man auch extern eine ppm erstellen. Für die externe Erstellung gilt ebenfalls 2.1.1

Bild: Anlage 2, TurboCad Makroeditor für Parametrieteile.

!! Diese Funktion nicht mit dem Thema „Parametrieteile mit Zwangsbemaßung“ verwechseln.

2.3 Dll´s und exe .dll / .exe

Hier können verschiedenste Entwicklungsplattformen genutzt werden. Das Ergebnis muss eben eine exe- bzw. dll Datei sein. Diese dll´s werden in einem speziellen TurboCad Verzeichnis gespeichert; Z.B. in C:\Programme\...\draggers.

Bild: Anlage 3

Mit Visual Basic 5/6 und Visual C++ können diese dll´s erzeugt werden. Logisch dass, man natürlich auch in diesen Programmen einen TurboCad konformen Quellcode eingeben und kompilieren muss. Es gibt definierte Regeln mit denen die dll´s zu erstellen sind. In der Entwicklungsumgebung wird auch der „Name“ der dll definiert, ggf. ein Icon angegeben sowie ein Ort benannt wo diese später in der TurboCad- Menüleiste zu finden sein werden.

2.3.1 VisualBasic - VB5/6 .dll / .exe

gibt es nicht mehr. VB5 und VB6 können, je nach Version, exe und dll´s erzeugen. Eine weiteres Microsoft Programm war VBCCE. Es basierte auf VB5 und konnte ausschließlich dll´s erzeugen.

VisualBasic 5, 6 und VBCCE gibt es nicht mehr – und toten Pferden soll man nicht nachtrauern.

Bild: Anlage 7

2.3.2 Excel 2010 -VBA

Visuell Basic für Application – kurz: VBA Ausführen 

VBA ist ja ein reduziertes VB6 und in Anwendungen wie z.B. Excel, Word, CorelDraw etc. vorhanden. Mit der Taste F11 gelangt man in den VBA- Editor, kann eine „Function" schreiben, kompilieren und ausführen. Bei der Ausführung wird gleich über die com-Schnittstelle in ein geöffnetes TC-Zeichenblatt geschrieben.

Im Excel-VBA muss man vorab unter Verweise mit „TurboCad4.1 Programmable Objects" die Tür zur TurboCad-Com-Schnittstelle bekanntgeben. Diese Datei ist ein Link zur, mit TurboCad installierten, Com-Schnittstelle. Bei meiner Version 17 ist dies imsigx17.dll. Bild: Anlage 5

2.3.3 VisualStudio 2010 Express – ist kostenfrei und beinhaltet u.a. Visual Basic 2010 Express –

Das Studio enthält, als Nachfolger von VB6, VisualBasic 2010 Express. Die .net Anwendungen benötigen immer die Bibliothek „Netframework". Sie ersetzen als kostenlose Version „Express" zukünftig auch die Skriptsprache VisualBasic-Script (VBS).

In den .net- Programmiersprachen muss unter |Verweise|"TurboCad4.1 Programmable Objects" aktiviert werden. (Angezeigt wird bei meinem TurboCad 17 dann die hinterlegte Datei imsigx17.dll). Es wird im Projektverzeichnis ..\bin\debug direkt eine .exe erzeugt. Auch das Speichern als .dll ist möglich. Bild: Anlage 6

2.3.4 Ruby ist eine kostenfreie Software

Diese Programmiersprache soll zukünftig den Wegfall von VBA in TurboCad ersetzen. Zurzeit ist Ruby zumindest in der US-Version von TurboCad 18 enthalten. Eine Dokumentation zu Ruby wird allerdings von Imsi nicht mitgeliefert.

2.3.5 Visual Basic Script -VBScript oder VBS .vbs

VBS ist eine Skriptsprache von Microsoft die ähnlich zu VB bzw. VBA ist. Es gibt allerdings nur den Variablentyp: Variant. In VisualBasic-Skripten sind Eingaben per Inputbox und Parameterübergaben möglich. Unter VBS erstellte Makros können im TurboCad-Makrorecorder geöffnet werden jedoch auch extern gestartet werden. Dabei sollte jedoch eine TurboCad-Zeichnung geöffnet sein. Eine Abfrage ob dies der Fall ist oder eine TurboCad-Zeichnung geöffnet werden soll, kann im Skript geschrieben werden.

Startet man eine Datei vom Typ vbs, startet ein windowseigener Skriptinterpreter. Der Windows Script Host (WSH) ist eine COM-basierte Laufzeitumgebung für Skriptsprachen. Eine Alternative bietet AutoIt. Ein Thema bei Skripten ist allerdings auch, dass diese mit Viren befallen sein können und gnadenlos, ohne Prüfung ausgeführt werden. Dateien vom Typ .vbs lassen sich einfach in .exe konvertieren(?). Als Editoren sind einfache Editoren brauchbar wie Notepad++, PSPad, Windows Editor etc.. Gut ist auch VBsedit (.com). Dieser Skripteditor beherrscht Debugging und unterstützt bei der Eingabe mit IntelliSense, d.h. einer Vervollständigung von Befehlen und Vorschau der Befehlsparameter. Für Scriptexperten zu empfehlen ist das TCAD Script Centre von David Bell. Vorteil: Massiver Performancegewinn gegenüber dem Makrorecorder und extern gestarteten Skripten. Nachteil: Nachbearbeitung der Skripte für eine dem Script Centre angepasste Form.

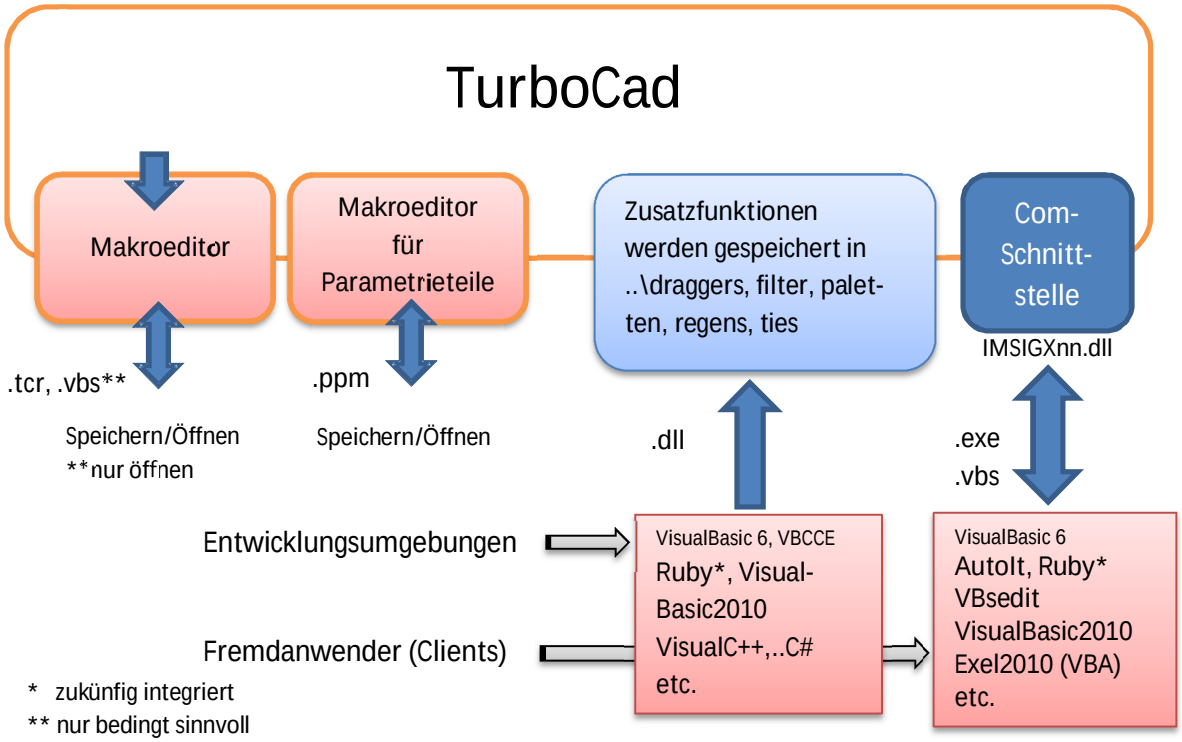
Fazit: VisualBasic-Script wird zwar noch von Microsoft unterstützt, ist jedoch zukünftig eine tote Sprache da der Nachfolger mit VisualBasic.net bereits vorhanden ist.

2.3.6 Lazarus / Delphi ??

2.3.7 Autoit – nicht nur ein Ersatz für den Windows Script Host

Empfehlenswert, wenn viele TurboCad- Aufgaben automatisiert werden müssen. Beispiel: Alle Zeichnungen öffnen und ausdrucken. Dazu übergibt Autoit wenige Befehlsstrings an TurboCad, welches diese dann intern bearbeitet. Um jedoch aufwändigere Geometrien zu erstellen ist Autoit evtl. zu langsam. Mit AutoIt erstellte .exe-Dateien sehen kompiliert aus. Tatsächlich ist jedoch die .exe selbstentpackend. Das so entpackte Script läuft dann auf dem ebenfalls entpackten Interpreter. Es lohnt sich AutoIt näher anzuschauen.

3. Entwicklungsumgebung in und um TurboCad



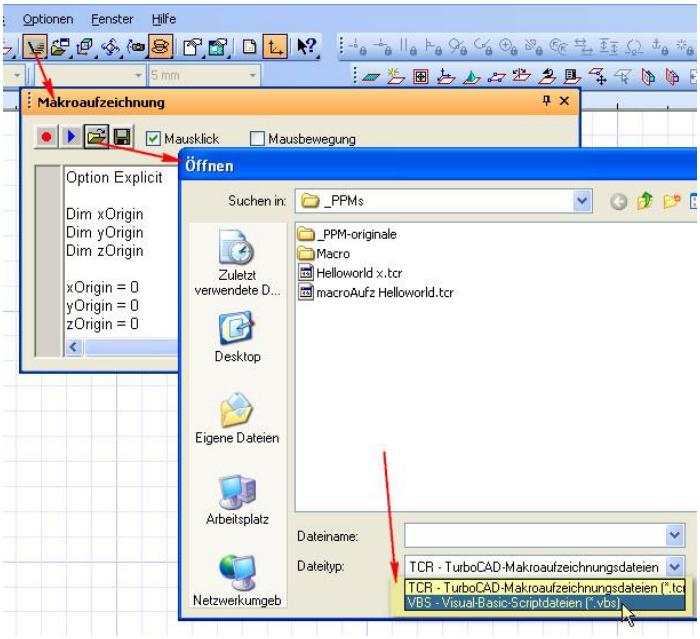
Wichtig: Bitte beachten, dass viele Bezeichnungen/Namen urheberrechtlich geschützt sind.
Ebenso ist zu beachten ist, dass ich unerfahrener Anwender bin und sicherlich manches falsch verstanden und wiedergegeben habe.

Mit freundlichen Grüßen. Leopoldi

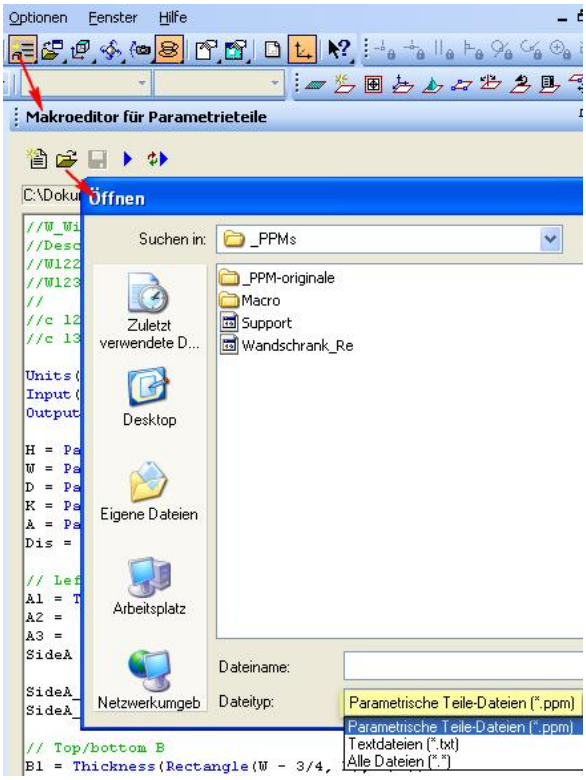
Informationen / Quellen:

http://ww3.cad.de/foren/ubb/Forum229/HTML/000048.shtml	Quell meines Wissens
http://ww3.cad.de/cgi-bin/ubb/forumdisplay.cgi?action=topics&forum=TurboCAD+SDK+VBA&number=229&DaysPrune=1000&LastLogin=&mystyle=	Turbocad SDK VBA
http://forums.turbocad.com/	US- Forum \ SDK Corner
http://www.microsoft.com/germany/express/download/	Visual Studio 2010 Express Basic etc.
http://www.pspad.com/de/download.php	Editor für Skripte
http://autoit.de/	Alternative zum Windows Script Host
http://VBSedit.com/	Editor m. Intellisense u. Debugger
http://ruby-lang.org/de/	Ruby, Infos auch in wikipedia

Anlage 1

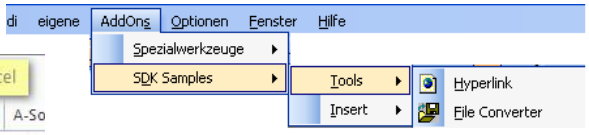


Anlage 2 (siehe auch 20)

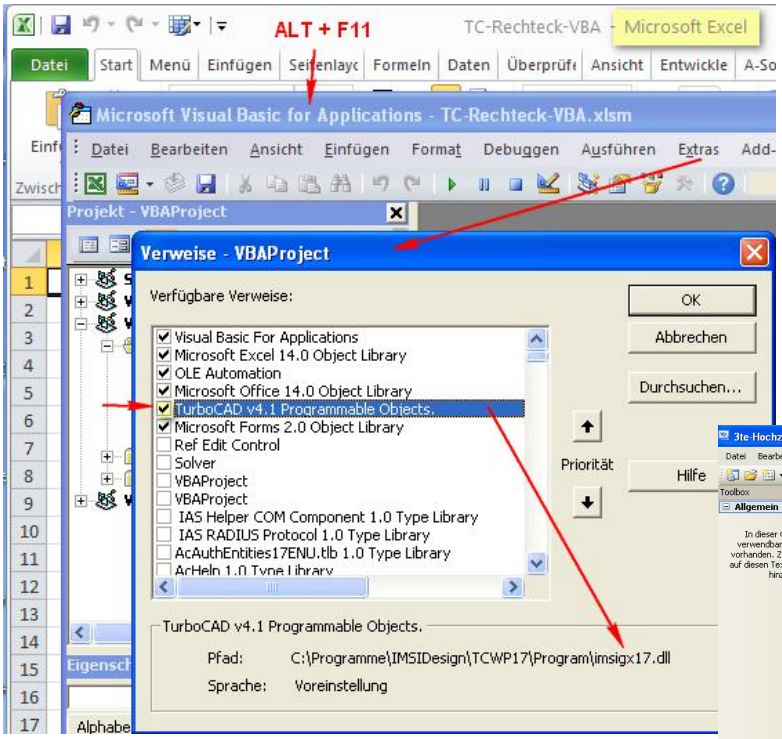


Anlage 3

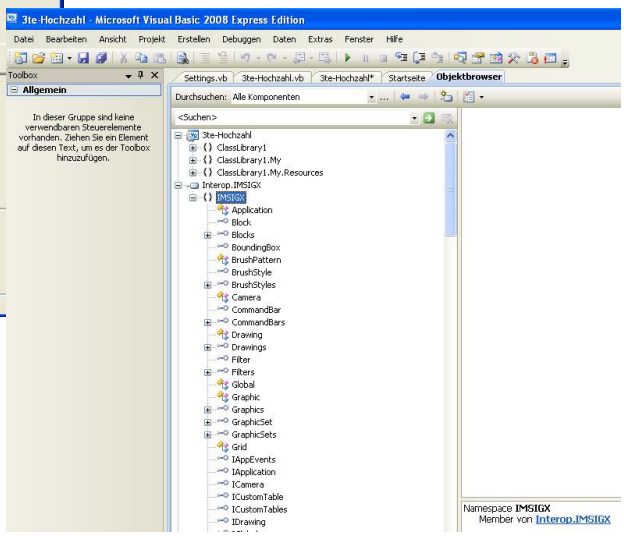
Anlage 4



Anlage 5

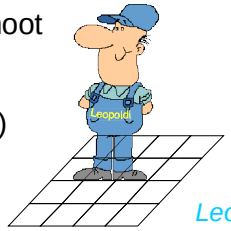


Anlage 6



Anlage 7

leider kein Screenshoot vorhanden -
aber ich halte mal den Stellplatz frei :-))

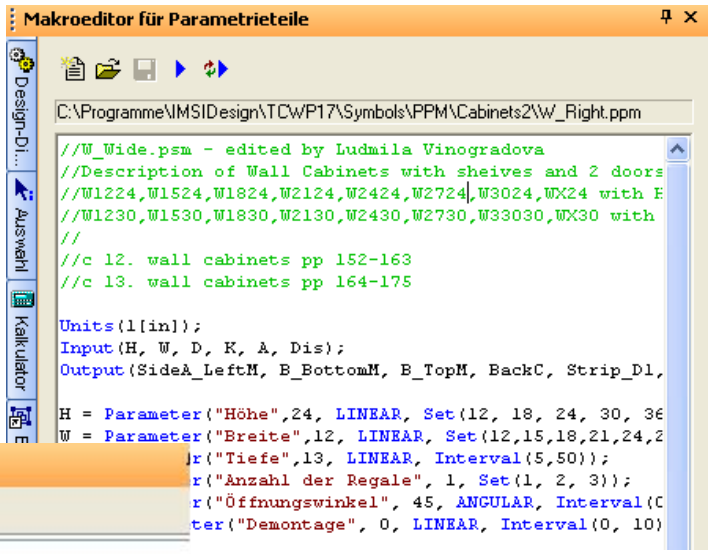


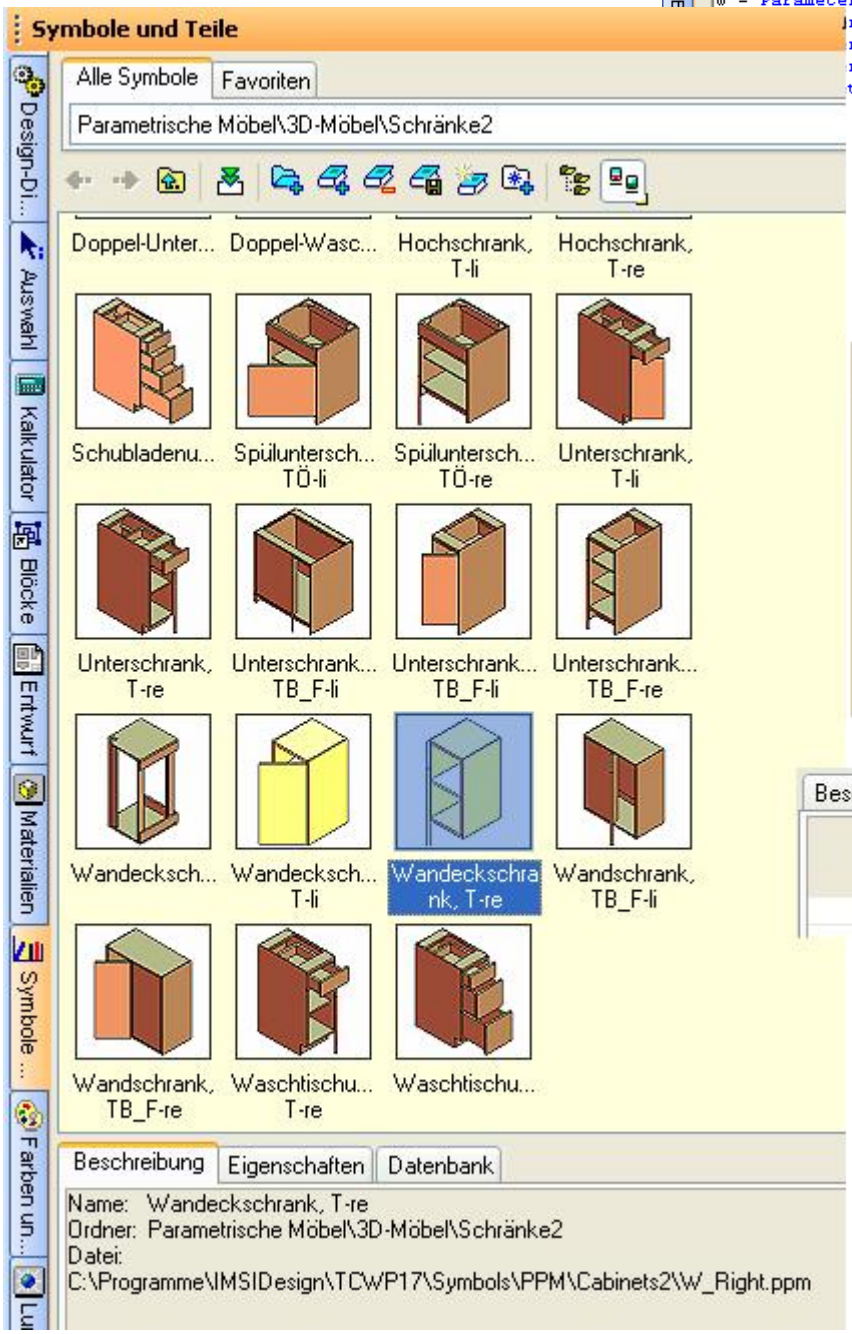
Viel Spaß beim werkeln wünscht
mfg.
Leopoldi

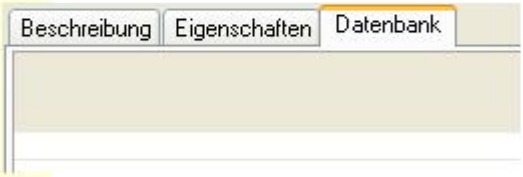
Anlage 20

Der Quellcode wird im Makroeditor erstellt.

Parametrische Teile sind in der Symboltabelle vorhanden. Dabei ist es besonders vorteilhaft, dass unter Eigenschaften die Variablen einzugeben werden können.

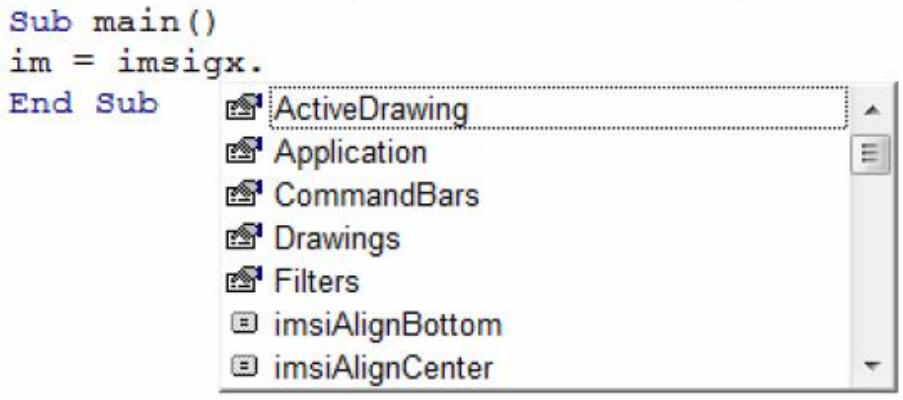




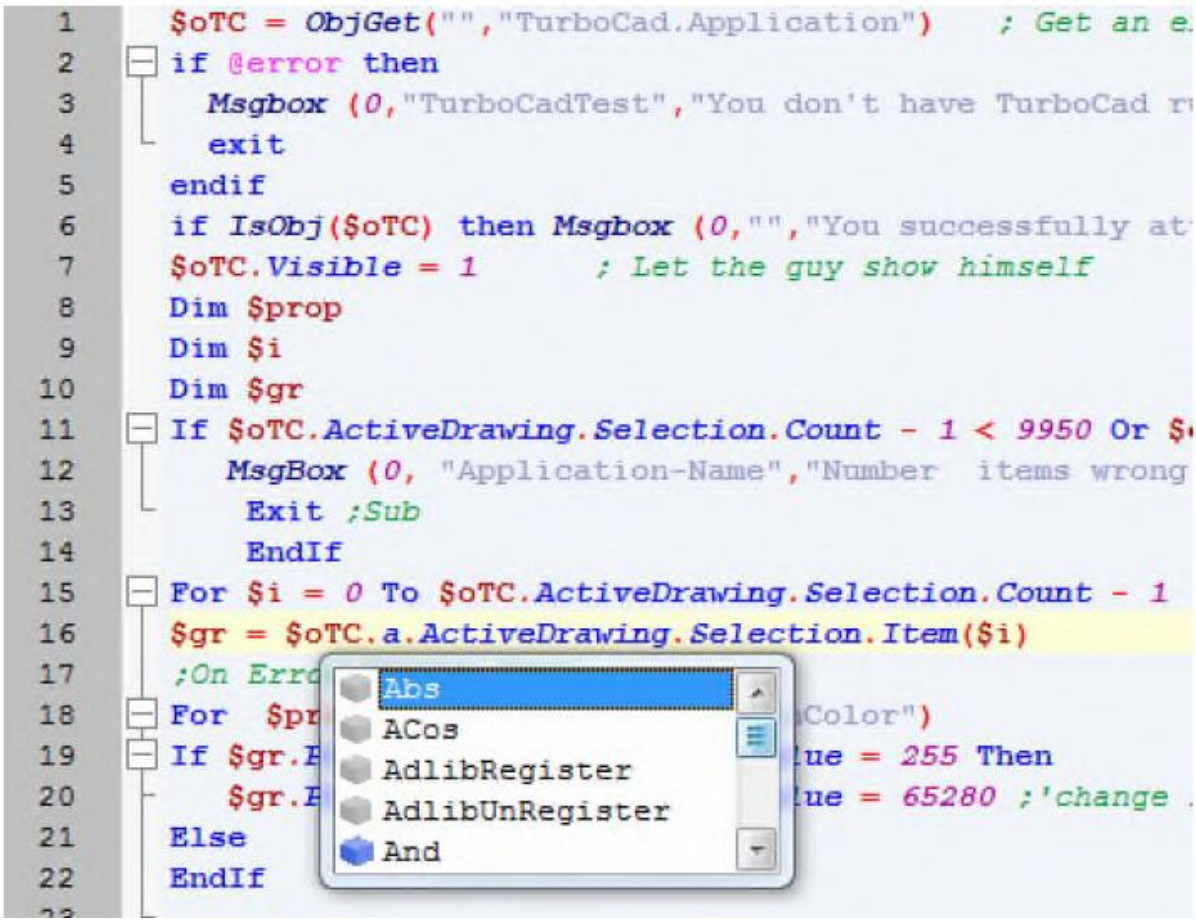


Anlagen zu VB6 und AutoIt

VB6 Projektentwicklungsumgebung (Debugger)
wenn das Turbocad-Objectmodel referenziert ist, werden die Syntaxhilfen angezeigt



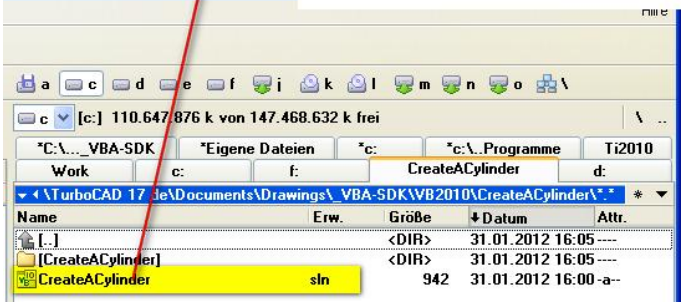
AUTOIT Projektentwicklungsumgebung (Debugger) Scite
hier gibt es auch so eine Vervollständigungshilfe, aber für die TC-Objekte nicht (oder genauer gesagt: ich habe das bisher nicht gefunden)



Anlage: Beispiel Projekt öffnen und starten in Visual Basic 2010 -

Das Projekt „CreateACylinder“ stammt aus dem amerikanischen Forum

Die .sln auf das Desktop-Icon ziehen. (sln =MS Visual Studio solution files)



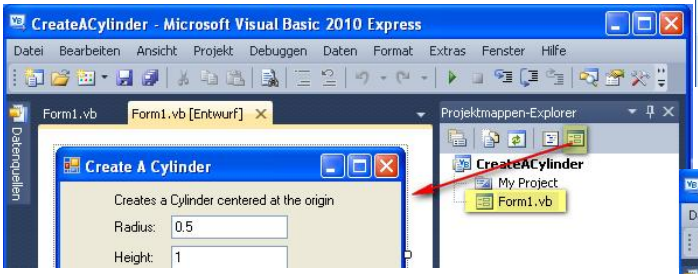
Visual Basic 2010 wird geöffnet.

Alternativ kann man Visual Basic 2010 starten und ein „Projekt öffnen“. Dann die .sln Datei wählen.

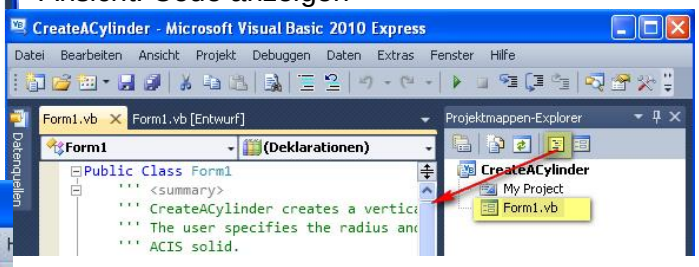
Da die Version älter als 2010 ist, wird diese konvertiert.

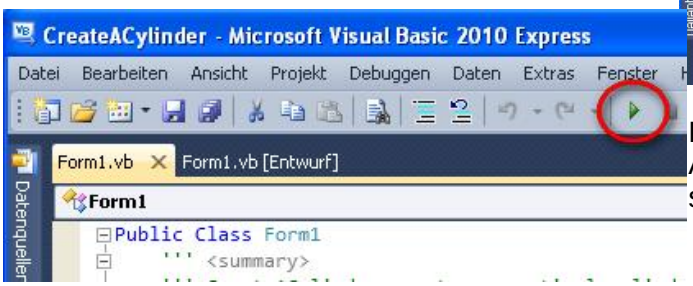
Mit „Fertig stellen“ startet die Konvertierung.
Über „Weiter“ kann man die Option Backup und eine Protokolldatei abwählen.

Ansicht: Forms



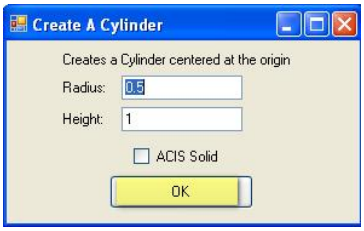
Ansicht: Code anzeigen



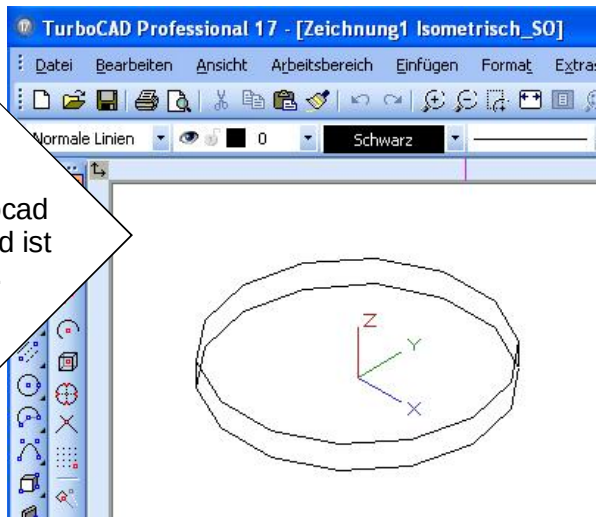


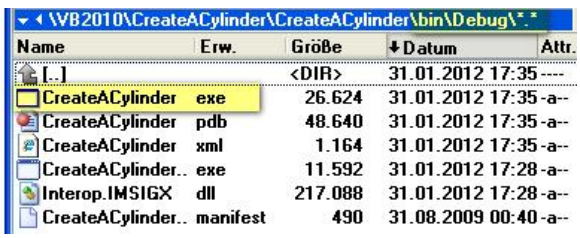
Per Pfeil oder Taste F5 die Ausführung (Debugging) starten.

Es folgt das programmierte Eingabefenster



Nach „OK“ startet Turbocad und wie von Geisterhand ist der Zylinder gezeichnet.





Im Projektverzeichnis ..\bin\debug ist auch eine **CreateACylinder.exe** erstellt worden. Diese kann man nun auch unabhängig von VisualBasic starten.