

Ordnung für Leopoldi ;-)

Ich versuchs mal ... einige Sachen weiß ich ja auch selbst nicht.

Ausführbare Dateitypen

- 1.1 **.ppm** ist soweit korrekt, Name entweder parametric part macro oder Parametric part manager

Sprache ist eine einfache geometrische beschreibungssprache, ähnlich VBS, umbenennen und laufenlassen von vbs scripten funktioniert jedoch **nicht**.

Ppm eignet sich besonders für den Aufbau von prototypbibliotheken mit denen man sehr schnell durch ändern der parameter - Höhe Schrank, Breite, Anzahl Türen als Beispiel – sehr schnell neue Zeichnungsteile generieren kann.

Ich halte das Konzept für sehr leistungsfähig, auch Geschwindigkeit ist ok Nachteile: Eigentlich nicht dazu geeignet tools oder kleine programme zu entwickeln – der schwerpunkt sind wirklich parameterteile. Dazu kommt:

Beim ppm bist du mutterseelenallein – gibt zwar dokumentation aber es sind kaum anwendungen oder Beiträge in den einschlägigen Foren zu sehen.

- 1.3 **tcm** hat mit McIntosh nichts zu tun , einfach der Name für **turbocadmacro**, dateiendung der früheren vba macros
- 1.4 **tcr** richtig, heißt wohl turbo cad recorder(oder record)
- 1.5 richtig
- 1.6 falsch – keine Exe sondern eine dll – dynamic link library. Diese Dateien enthalten zwar ausführbaren code, werden aber nicht wie eine exe als alleinstehende Anwendung gestartet, sondern brauchen eine Wirtsanwendung – neudeutsch host um ausgeführt zu werden.

In Turbocad besteht die möglichkeit mittels bestimmter Programmierumgebungen - vor allem Visual Basic oder Visual C++ jeweils in der Version 5/6 dlls zu entwickeln die bestimmte Tätigkeiten innerhalb der Turbocad werkzeuge übernehmen können wie z.B. 3.Masszahl von Bernd Ruethschilling. Es gibt dazu bestimmte regeln mit denen die dlls erstellt werden müssen, bei der Erstellung des Projekts in der Entwicklungsumgebung wird auch ein Name für die dll vergeben und ein Ort wo sie später in der menüleiste zu finden ist. Des weiteren gibt es noch die möglichkeit mittels eines resourceneditors der Entwicklungsumgebung ein Icon zu kreieren, wodurch sich die fertige dll in keiner weise von den eingebauten werkzeugen unterscheidet. Gespeichert wird diese dll im draggers unterverzeichnis des program ordners. TCad lädt dann beim hochfahren die dlls aus dem draggers ordner in den speicher. Bei der dll von bernd sieht man auch folgendes – die dll hat einen kleinen Fehler, wohl durch nicht wieder freigegebene objekte nach abschluss der anwendung, wird die dll nur mitgeladen weil sie im draggers ordner steht, dann passiert nichts, wird sie aufgerufen, dann stürzt Tcad beim runterfahren ab. – Dies ist ein sehr anschauliches Beispiel für die Mechanismen die hier wirken.

Wie erstelle ich diese Dateien?

- 2.1 TC intern genau

- 2.2 Scripte Quelltexte können mit einem simplen Texteditor geschrieben werden.

Empfehlung kostenlos: **PSPAD** Text bzw. Programmeditor, bei dem verschiedenste Syntaxen? Eingestellt werden können, u.a. auch VB und VBScript. Nebenbei Dokumentiert dieser Editor auch Funktions und Prozedurköpfe, kann mit hex rechnen ..

2.3 stimmt

2.4 stimmt teilweise, wird von microsoft nicht mehr verkauft, ist jedoch immer wieder mal auf ebay erhältlich.

Weitere möglichkeit vor ca. 11 12 jahren hat microsoft ein programm kostenlos rausgerückt welches VBCCE hieß – Visual Basic Control Creation Edition. Damit sollte der Markt für ActiveX objekte unterstützt werden. Die VBCCE war ein Visual Basic 5.0 Professional, bei dem die Möglichkeit exe dateien zu erzeugen ausgegraut war - dafür kann man damit dlls erstellen! Vielleicht hast du ja noch irgendeine alte cd rumliegen...

Dll ist **nicht** Exe, dlls kann man wenn sie fachgerecht und tcad konform erzeugt wurden im draggers ordner als ausführbare Kommandos speichern. Exe dateien kann man mit TCAD auch benutzen, dies wäre dann eine ausführbare datei die irgendwo auf der Festplatte liegt, und beim Aufruf aufs laufende tcad zugreift, um z.B. mit externen Daten Kurvenzüge in Tcad zu erstellen. Dadurch dass hier gewissermassen von aussen auf TCAD zugegriffen wird, ist so eine exe bedeutend langsamer als eine dll im draggers ordner die ja auf die TCAD objekte ohne die Schnittstelle zu einem anderen programm zugreifen kann hoffe ich habe es hinreichend erklärt

2.5 VBNet damit habe ich noch nicht gearbeitet, gibt jedoch einige beispiele in der hilfe und im US forum

2.6 Lazarus / Delphi gibt wenig beispiele für delphi ich mach damit nichts

2.7 Du meinst wahrscheinlich autoit wohl interessant wenn umfassende aufgaben automatisiert werden sollen. Beispiel alle zeichnungen aus diesem oder jenem verzeichnis öffnen und ausdrucken - d.h das programm übergibt wenige Befehlsstrings und tcad werkelt dann dafür ist es ok. Beim Cad ist aber sehr oft die notwendigkeit da umfangreiche geometrien heißt viele viele punkte zu übergeben und dafür ists meiner meinung nach viel zu langsam.

Wie führe ich die verschiedenen Dateien nun aus bzw. integriere sie???

3.1 Makros: AddOns|SDK Samples|Insert nein das gilt nicht für makros sondern nur für dlls

3.2 Makros: Öffnen im Makroeditor ja großer Vorteil von VBS Scripten im gegensatz zu .tcr: In VBS können per inputbox eingaben und parameter übergeben werden, das geht wie ich weiß in tcr scripten nicht. Dafür beim aufruf ein klick mehr VBS Makros können natürlich auch ausserhalb von tcad mittels doppelclick aufgerufen werden, sollten dafür jedoch eine Abfrage enthalten ob tcad schon läuft und es im zweifel Per programm selbst laden.

Empfehlenswert installation des TCAD Script Centre von david bell

Vorteile: Massiver Performancegewinn gegenüber Makrorecorder und extern gestarteten Scripten,
Möglichkeit das script mit iconbutton zu belegen

Nachteil: Nachbearbeitung der scripte für eine dem script centre angepasste form