Visual Basic Editor CATIA V5

Daniel Frauenrath

Allgemein

Der Visual Basic Editor ist fester Bestandteil von CATIA V5. Im Gegensatz zum internen Editor für CATScript und CATVbs hat der Visual Basic Editor entscheidende Vorteile bezüglich Erstellung und Debugging von Scripten unter CATIA V5.

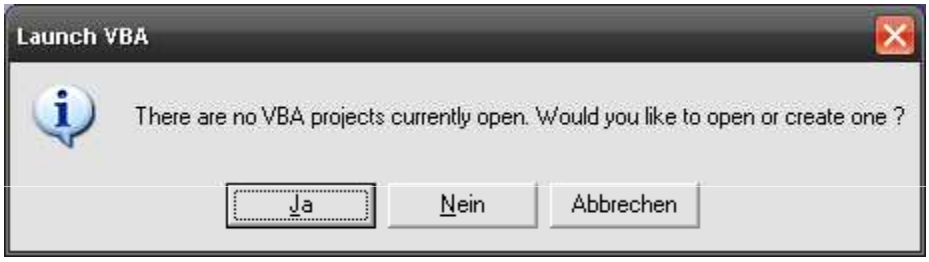
Sowohl CATScript als auch CATVbs können mit Hilfe des Visual Basic Editors erstellt und debugged werden. Die Umwandlung in CATScript bzw. CATVbs erfolgt mit Einschränkungen über Copy&Paste.

Diese Beschreibung soll einen Einblick über die Arbeitsweise mit dem Editor gewähren.

Editor starten

Starten Sie den Editor über Tools ➔ Macro ➔ Visual Basic Editor (oder Alt + F11)

Falls in den Makrobibliotheken noch kein CATVba Projekt angelegt ist, erscheint folgende Warnmeldung:



Bestätigen Sie diesen Dialog mit „Nein“. Sie können im Anschluss direkt aus dem Editor ein neues CATVba Projekt zu den Makrobibliotheken hinzufügen.

Ist bereits ein CATVba Projekt in den Makrobibliotheken angelegt wird der Editor sofort gestartet und das zuletzt aktivierte Projekt wird geöffnet.

Projekt hinzufügen

Um ein neues CATVba Projekt zu den Makrobibliotheken hinzuzufügen und damit den Visual Basic Editor nutzen zu können gehen Sie wie folgt vor:

Wechseln Sie in den Visual Basic Editor und wählen Sie

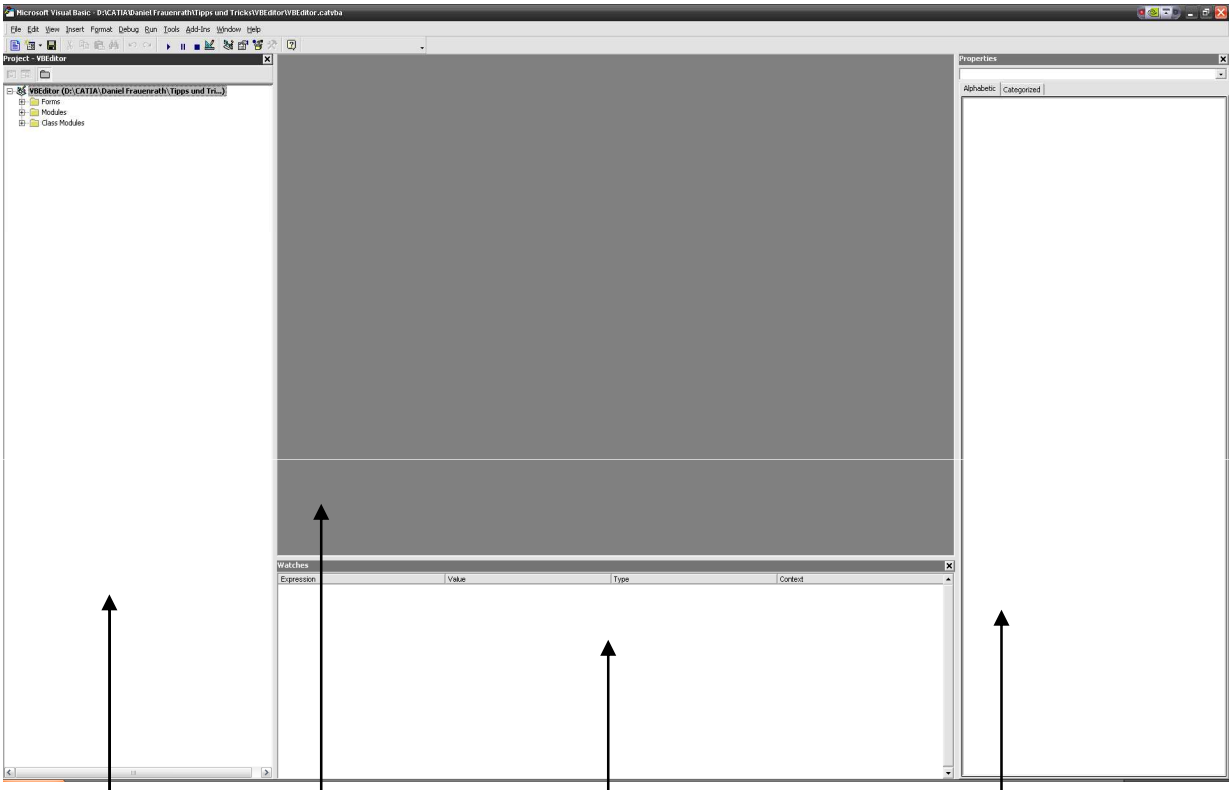
File ➔ Macro libraries

CATIA wird nun aktiviert und es erscheit die Dialogbox der Makrobibliotheken

Fügen Sie entweder ein neues (Schaltfläche „Create new library“) oder ein bereits existierendes (Schaltfläche „Add existing library“) Projekt zu den Makrobibliotheken hinzu.

Verlassen Sie im Anschluss den Dialog mit „close“ und wechseln Sie wieder in den Microsoft Basic Editor.

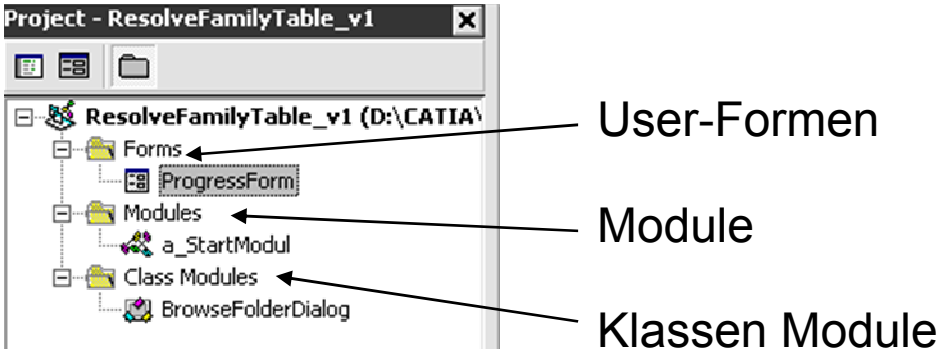
Der Editor


Codefenster
Eigenschaftsfenster
Projektexplorer
Überwachung

Der Projektexplorer

Der Projektexplorer bietet eine Übersicht über alle geladenen VBA-Projekte.

Hierbei werden die Projekte wie folgt aufgegliedert:



User-Formen:
Speicherort für alle User Formen (selber erstellte GUI)

Module:
Speicherort für alle Module (enthält auch die CATMain() Routine

Klassen Module:
Speicherort für eigene Klassen (Funktionen)

Das Codefenster

Das Codefenster enthält den Quelltext von Modulen, User-Formen und Klassenmodulen.

Ein Doppelklick (F7) auf ein Modul oder ein Klassenmodul öffnet dieses im Codefenster.

Ein Doppelklick (F7) auf eine User-Form öffnet die User-Form im Design Mode (neue Steuerelemente können hinzugefügt, verändert oder gelöscht werden).

RMT -> View Code (Shift+F7) auf eine User-Form öffnet den Code der Form im Codefenster.

Das Eigenschaftsfenster

Das Eigenschaftsfenster wird verwendet um die Eigenschaften eines ausgewählten Objekts (Modul, User-Form, Klassen Modul) zu ändern. Des weiteren können im Eigenschaftsfenster die Eigenschaften für alle Steuerelemente (z.B. Buttons, Combo-Boxen, etc.) geändert werden.

Das Eigenschaftsfenster ist vor allem beim Arbeiten mit User-Formen wichtig.

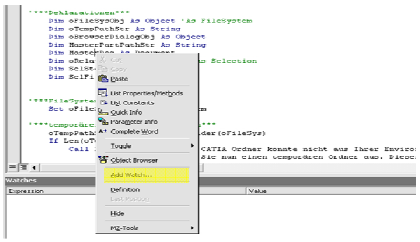
Beim Arbeiten mit User-Formen müssen häufig die Eigenschaften und das Verhallten von Steuerelementen, aber auch der Formen selbst, geändert und angepasst werden.

Das Überwachungsfenster

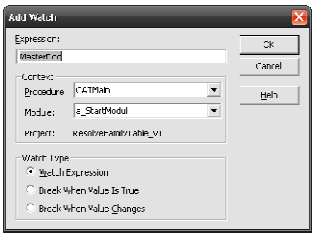
Das Überwachungsfenster (Watches) wird eingesetzt zum Überwachen von Variablen, Objekten und Ausdrücken. Hierbei wird der Name, Wert, Datentyp sowie der Gültigkeitsbereich angezeigt.

Die Vorgehensweise zum Überwachen von Variablen, Objekten oder Ausdrücken ist wie folgt:

1. Variable, Objekt oder Ausdruck im Codefenster markieren
2. RMT -> Add Watch



3. Allgemeine Einstellungen im PopUp-Dialog vornehmen



4. Bestätigung mit „OK“ zum Erstellen der Überwachung

Überwachung

Die Überwachung von Variablen, Objekten und Ausdrücken ist ein sehr hilfreiches Tool beim Debuggen (Fehlersuche) im Quellcode.

Werden Objekte überwacht, wertet der Visual Basic Editor die Unterobjekte sowie verfügbaren Klassen des Objekts gleichzeitig aus. Um diese Ergebnisse zu kontrollieren muss auf das „Plus“-Zeichen vor dem jeweiligen Objekt im Überwachungsfenster geklickt werden.

Debugging Funktionen

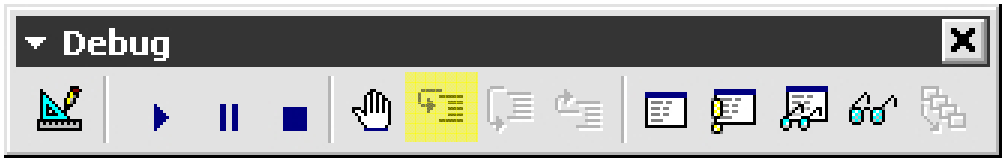
1. Einzelschritt / Step into

Der Einzelschritt (Step into) führt den Quellcode Zeile für Zeile aus. Diese Funktion wird häufig in einem fehlerhaften Quellcodeabschnitt zusammen mit der Überwachungsfunktion eingesetzt.

Das Tool eignet sich darüber hinaus auch zum erlernen der VBA-Sprache, da Ausdrücke explizit überwacht werden können.

ANMERKUNG:

Der Befehl befindet sich auf der Symbolleiste „Debug“ die standardmäßig ausgeblendet ist. Zum Einblenden der Symbolleiste wählen Sie
View ➔ Toolbars ➔ Debug



Debugging Funktionen

Um mit der Einzelschrittausführung des Quellcodes zu beginnen, muss sich der Mauscursor innerhalb der Routine befinden die debugged werden soll (z.B. CATMain()).

Nach dem ersten Klick auf die Funktion „Einzelschritt“ wird die erste Quellcodezeile (Definition der Routine) gelb hinterlegt. Die gelb hinterlegte Zeile wird erst beim NÄCHSTEN Klick auf die Funktion kompiliert und anschließend ausgeführt.

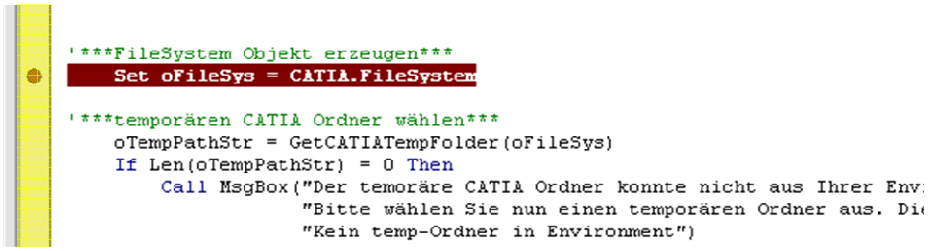
Die gelbe Markierung kann durch halten und ziehen am gelben Pfeil wieder nach oben verschoben werden. Wird zum Beispiel der Wert einer Variable während des Debuggings durch den Anwender geändert, können die Auswirkungen direkt, OHNE die Codeausführung neu zu starten, kontrolliert werden.

Debugging Funktionen

2. Haltepunkt / Breakpoint

Haltepunkte sind explizit markierte Quellcodezeilen an denen die Ausführung des Quellcodes angehalten wird. Anschließend kann der Quellcode mit Hilfe der Einzelschritte weiter ausgeführt werden.

Zum Einfügen eines Haltepunktes klicken sie mit der linken Maustaste auf den grauen Rand vor der Codezeile an der der Haltepunkt eingefügt werden soll.



Zum Löschen eines einzelnen Haltepunkts kehren Sie den Vorgang um.

Zum Löschen aller Haltepunkte eines Projektes wählen sie im Menü „Debug“ die Funktion „Clear all Breakpoints“.

Haltepunkte werden im Projekt gespeichert und sind auch nach einem Neustart der Session wieder verfügbar.

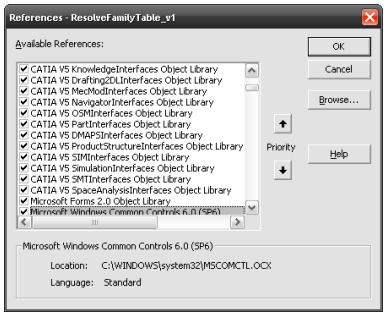
Verweise

Verweise sind „Links“ zu externen Bibliotheken. Sie werden eingesetzt um Klassen, Methoden und Datentypen von externen Programmen (z.B. MS Excel) nutzen zu können.

Diese Methode wird „early Binding“ genannt. Im Gegensatz hierzu wird beim „late Binding“ auf die Verweise verzichtet. Alle Objekte werden hierbei als Objekt deklariert. Die internen Datentypen können ohne Verweise nicht genutzt werden.

Um einen Verweis, zu einer externen Bibliothek, zum aktuellen Projekt hinzuzufügen klicken Sie auf

Tools ➔ References (der Quellcode darf nicht ausgeführt werden)



Klicken Sie auf den „Browse“-Button um den Pfad zur Bibliothek zu wählen und bestätigen Sie den Dialog mit „OK“.