

Bei der Integration der MMIX-Komponenten boten sich zwei Möglichkeiten:

- als Kommandozeilenaufruf im Hintergrund
- als interner C Aufruf

Ersteres wird realisiert als Aufruf, welcher im Hintergrund eine Kommandozeile öffnet, einen Befehl ausführt und den Output in die IDE-Komponente piped.

Bei der zweiten Möglichkeit werden die benötigten Teile der MMIX-Umgebung als C-Datei in die IDE eingebunden. Diese werden dann im Hintergrund mit den nötigen Dateien ausgeführt. Hierzu muss der Quelltext der C-Dateien verändert werden sodass der Input nicht von der Konsolenzeile stammt und kein Output auf eben diese zurückgeht.

Dies wird bewerkstelligt mit Change-Files (.ch). Hierzu später mehr.

Zuerst die Vorteile und Nachteile beider Möglichkeiten und warum für was entschieden wurde.

Vorteile des Kommandozeilenaufrufs:

- leicht einzubauen: Da man hierfür lediglich die exe-Dateien benötigt kann dieser Aufruf sehr leicht eingebaut werden. Im Code wird dies durch lediglich eine Zeile bewerkstelligt. Diese Zeile ruft den Befehl auf und piped gleichzeitig den Output in eine String-Variable.
- keine Änderungen am eigentlichen C-Code: Man muss den C-Code nicht ändern wodurch die exe-Dateien einfach durch den Nutzer ausgetauscht werden können. Sollte eine neue Version der MMIX-Komponenten herauskommen können diese einfach ausgetauscht werden, ohne dass die IDE geupdatet werden muss. Wird die IDE dann so intelligent geschrieben dass sie sich automatisch die nächste/vorhandene MMIX-Installation sucht muss der Nutzer der IDE nicht mal sagen wo sich diese befindet.

Nachteile des Kommandozeilenaufrufs:

- Im Hintergrund wird ein Kommandozeilenfenster geöffnet: Der Aufruf des Kommandozeilenbefehls verursacht ein tatsächlich „physisches“ Öffnen eines Kommandozeilenfensters. Dieses Fenster trägt nicht gerade positiv zum Erscheinungsbild der Anwendung bei. Im Endeffekt sieht es dann so aus dass sich ein cmd-Fenster öffnet irgendetwas geschieht, aber kein Output angezeigt wird und dann nach kurzer Zeit sich dieses Fenster wieder schließt, worauf dann die eigentliche Ausgabe in der IDE erfolgt.
- Benötigte „Installation“ von MMIX (je nach Bauart der IDE): Wird die IDE so gebaut dass sich die MMIX-Exe-Dateien nicht im selben Verzeichnis

befinden müssen, muss entweder der Nutzer dem Programm sagen wo sich die Dateien befinden oder das Programm muss sich diese selbst suchen. Da es die Benutzbarkeit des Programms stark beeinträchtigt und das einfache Bediengefühl darunter leidet, können wir dem Nutzer nicht zumuten der IDE zu sagen wo sich die Dateien befinden. Im Endeffekt muss ein Suchalgorithmus gebaut werden, welcher den ganzen PC durchsucht wenn im vorher gespeicherten Verzeichnis keine MMIX-Dateien gefunden werden konnten.

Vorteile der internen/integrierten C-Variante:

- Der Benutzer bekommt nichts vom Aufruf mit: Da die C-Datei intern ausgeführt wird. Es öffnet sich kein Kommandozeilenfenster und es muss auch keine Datei ausgewechselt werden oder der Pfad dafür eingegeben werden.

Nachteile der internen/integrierten C-Variante:

- Der C-Code muss zuerst verändert werden: Obwohl sich die Integration des C-Codes auch wieder nur aus einer Zeile besteht muss der Original-MMIX-Code erst einmal umgeschrieben werden sodass er mit den Änderungen klar kommt. Dies geschieht mit sogenannten Change-Files (.ch). Mehr dazu wird später geklärt.

Im Endeffekt fiel die Entscheidung auf die interne Variante.

Auch wenn die externe Variante einfacher einzubauen ist, kann man dem Nutzer den Aufwand nicht zumuten auch noch eine MMIX-Installation zu haben und diese auf dem neuesten Stand zu halten. Der Benutzer geht davon aus dass er eine Entwicklungsumgebung bekommt wenn er eine IDE installiert.

Außerdem gilt es als „unsauber“ wenn ein Programm bei einem Druck auf einen Button eine Kommandozeile öffnet und diese leer wieder schließt ohne mir zu sagen was vorgefallen war oder was das schwarze Fenster da überhaupt tut.

Sicher ist es ein massiver Vorteil der externen Variante dass man den eigentlichen C-Code nicht ändern muss. Hierdurch können Fehler durch den Entwickler (mich) ausgeschlossen werden. Zu bevorzugen ist die interne Variante, bei der der Code durch Change-Files verändert werden muss.