

## Praxistipps FrameMaker: Komfortabler Arbeiten mit Skripten

*Deutlich langsamer und weniger umfangreich als bei Adobe InDesign erfolgte in der Vergangenheit die Weiterentwicklung von Adobe FrameMaker. Da bei FrameMaker bislang eine Makroschnittstelle wie VBA fehlt, blieben viele Funktionalitäten, die sich die Anwendergemeinde wünschte oder schlicht dringend benötigte, auf der Strecke.*

**A**ber auch FrameMaker lässt sich anpassen und erweitern. Dazu bietet Adobe eine Schnittstelle an, mit deren Hilfe sich Plug-ins erstellen lassen. Die Schnittstelle heißt Frame Developer's Kit oder schlicht „FDK“ und erfordert ausgeprägte Programmierkenntnisse in der Sprache C++. Gibt es spezielle Probleme, die sich mit den FrameMaker-eigenen Bordmitteln nicht lösen lassen, bleibt dem Anwender die Wahl, die binären FrameMaker-Dateien in das MIF-Format zu konvertieren, um dort globale Ersetzungen vorzunehmen. Mit dieser Vorgehensweise lässt sich allerdings nur ein kleiner Teil der Probleme lösen. Alternativ kann der Anwender externe Plug-ins einsetzen. Wenn deren Funktionalität allerdings nicht den jeweiligen Anforderungen entspricht, bleibt häufig nur, ein Plug-in programmieren zu lassen.

### Es geht aber auch anders

Die Firma ElmSoft hat bereits vor sieben Jahren eine Schnittstelle zwischen FrameMaker beziehungsweise der Schnittstelle FDK und dem Anwender geschaffen: Das Plug-in „FrameScript“, das in Deutschland unter dem Namen „ElmScript“ vertrieben wird. Damit ist der fortgeschrittene Anwender in der Lage, auf alle Objekte zuzugreifen, die das FDK zur Verfügung stellt, beispielsweise die Objekte FrameMaker-Session, Dokumente, Absätze oder Formatierungen, und kann maßgeschneiderte Problemlösungen schaffen. Das Erlernen der Programmiersprache FrameScript ist nicht schwer. Auch Anwender ohne Erfahrung im Programmieren können bereits nach wenigen Tagen erweiterte Scripts schreiben. FrameScript erfordert keine explizite Deklaration von Variablen, diese werden bei Bedarf automatisch in den korrekten Datentyp konvertiert. Nicht spezifizierte Befehlsoptionen werden meist mit schlüssigen Vorgabewerten belegt. Ein FrameScript-Script – oder einfacher ein „FrameScript“ – ist eine Textdatei, die vom FrameScript-Plug-in zur Laufzeit kompiliert wird. Zur Ausführung des Scripts muss sowohl FrameMaker als auch das FrameScript-Plug-in auf dem jeweiligen Rechner installiert sein.

### Lösungen mit FrameScript

Welche Probleme mit welchem Script gelöst werden können, zeigt die folgende Auflistung, die einen ersten Überblick über die Möglichkeiten von FrameScript bietet.

- Aus Prozessgründen soll eine lokale Formatierung (fett, hochgestellt) vermieden werden. „PreventLocal-

Formatting.fsl“ verwendet bei Anwahl der entsprechenden Befehle oder Buttons die entsprechenden Zeichenformate (so vorhanden).

- Viele Plug-ins erstellen jeweils ihr eigenes Menü im Hauptmenü und überfrachten damit die Menüleiste. „GroupPluginMenus.fsl“ verschiebt alle oder ausgewählte Plug-in-Menüs in ein spezielles Menü.
- Das Einfügen von Variablen ist etwas umständlich und der Dialog kann nicht geöffnet bleiben. „MultiCatalog.fsl“ erlaubt das Einfügen von Variablen, das Zuweisen von Absatz- und Zeichenformaten, Vorgabeseiten und Umbrucheinstellungen über einen nicht-modalen Dialog.
- Eine Farbe soll durch eine andere ersetzt werden. „ReplaceColors.fsl“ erledigt dies in allen Objekten, die eine Farbe enthalten können; optional löscht es nicht verwendete Farben.
- Für die Platzierung von Grafiken ist ein komplexes Konstrukt aus mehreren Rahmen und Tabellen erforderlich. „AdjustFigures.fsl“ erstellt nach Auswahl einer zu importierenden Grafik dieses Konstrukt automatisch und trägt die Grafik-ID ein.
- Die Fehlermeldungen einer zu dokumentierenden Software werden in einer XML-Datei (Text) und einer Excel-Datei (Struktur) verwaltet. „UpdateFCRTTable.fsl“ liest diese Daten aus, erzeugt eine entsprechende Tabelle und formatiert diese.
- Eine Dokumentation enthält umfangreiche Belegungsformate, die mehrfach zugewiesen sind. „UpdateCondComments.fsl“ fügt für den Review die Kurznamen dieser Formate in den Fließtext ein.
- Ein umfangreiches Kundenverzeichnis enthält URLs und Mailadressen. „InsertHyperlinks.fsl“ fügt jeweils Hypertextmarker ein, um in der ausgegebenen HTML- oder PDF-Datei interaktive Links zu erzeugen.
- Beim Anlegen eines neuen Textrahmens wird immer das Absatzformat „Haupttext“ verwendet. „SetDefaultTextFramePgFmt.fsl“ ermöglicht die Anpassung des Formats.
- Es soll festgestellt werden, welche Vorgabeseiten in einer Dokumentation verwendet werden. „MasterPageUsage.fsl“ erstellt eine Liste oder einen Index der verwendeten Vorgabeseiten.
- Zur Bereinigung einer Dokumentation sollen die verwendeten Schriften identifiziert werden. „ReportFontUsage.fsl“ listet auf, wo die einzelnen Schriften verwendet werden (Absatz- und Zeichenformate, direkt im Text, auf Vorgabe- und Referenzseiten oder in Tabellenformaten).

- Für eine schnelle Prüfung oder zum korrekten Generieren eines Buchs sollen alle Dokumente geöffnet werden. „OpenDocsSilently.fsl“ umgeht die Fehlermeldungen, die für jedes Dokument in einem eigenen Dialog angezeigt werden (fehlende Schriften oder Grafiken), zeigt aber eine zusammenfassende Meldung der Fehler an.
- Die FrameMaker-Funktion „Gehe zu Seite“ unterstützt nur den Seitenwechsel im aktuellen Dokument. „GotoPage.fsl“ erlaubt, zu einer beliebigen Buchseite zu wechseln (auch: „3-12“ oder Vorgabeseite „Titel“).
- Innerhalb einer Tabelle in die benachbarten Zellen zu wechseln, ist nur mit der Maustaste oder kryptischen Tastaturkürzeln möglich, zum Beispiel „Esc tmu“. „TableNavigation.fsl“ erlaubt das Navigieren in Tabellen mit den Pfeiltasten.
- Zur Übersetzung müssen die Dokumente in das MIF-Format konvertiert werden, nach der Übersetzung wieder zurück. „FM2MIF2FM.fsl“ erledigt dies für das gesamte Buch.
- Nach der Übersetzung sind Texte häufig abgeschnitten, weil der Textrahmen zu klein angelegt wurde, beispielsweise Grafikbeschriftungen. „FindNextOverflow.fsl“ identifiziert diese Stellen.
- Nach der Übersetzung muss die Sprache aller Absätze umgestellt werden; Templates sind keine vorhanden, und aufgrund verwendeter Übersreibungen soll ein Formatimport nicht stattfinden. „ChangeBookPgLanguage.fsl“ ändert die Sprache und Trennregeln aller Absätze.
- Ein Glossar soll nach der Übersetzung alphabetisch sortiert werden. „SortParagraphSections.fsl“ erledigt das anhand bestimmter Absatzformate.
- Mit der Zeit sammeln sich immer mehr Grafiken an, die nicht mehr verwendet werden. „MoveUnused-Insets.fsl“ verschiebt diese in einen eigenen Ordner.
- Die Formatkataloge einer Dokumentation wachsen mit der Zeit immer mehr an. „RemoveUnused-Formats.fsl“ löscht alle Formate, die nicht mehr verwendet werden.
- Die Pflege und Analyse eines Templates erfordert eine visuelle Instanz der jeweiligen Formate. „TemplateTool.fsl“ erstellt einen Report mit Mustern aller Formate, Thumbnails der Vorgabeseiten und Auflistung der Parameter. Zusätzlich erlaubt es die Kommentierung der einzelnen Formate, die Änderung direkt im Report und das Zurückschreiben der geänderten Formate in das Template.
- Die Ausrichtung von Marginalien soll sich auf linken und rechten Seiten unterscheiden. „AlignSideHeads.fsl“ passt die Ausrichtung beim Speichern automatisch an.
- Die einzelnen Kapitel einer Betriebsanleitung sollen jeweils eigene Inhaltsverzeichnisse erhalten. „DocTOC.fsl“ fügt diese am Kapitelbeginn ein und aktualisiert sie beim Buch-Update.
- Die Acrobat-Lesezeichen in Fremdsprachen anderer Codepages werden von FrameMaker fehlerhaft erstellt. „PDF-Controller.fsl“ erzeugt korrekte Lesezeichen und erlaubt auch die Steuerung anderer PDF-Eigenschaften wie dem FrameMaker-Dokument angepasste Seitenzahlen.

#### Links

- FrameScript Homepage (en): [www.framescript.com](http://www.framescript.com)
- FrameScript-Infos (de): <http://framescript.eu/>
- itl-Scripts: [www.itl.de/html/deutsch/consulting/fsl/default.html](http://www.itl.de/html/deutsch/consulting/fsl/default.html)

Klaus Müller  
itl AG  
[klaus.mueller@itl.eu](mailto:klaus.mueller@itl.eu)

## Praxistipps Word: Nummerierte Listen

*Nummerierte Listen sorgen bei manchem Anwender für Kopfzerbrechen. Das Nummerieren mit der entsprechenden Funktion erweist sich als nicht besonders stabil. Immer wieder muss der Anwender eingreifen und den Beginn der Liste von Hand neu festlegen. In dieser Ausgabe wollen wir Ihnen daher zeigen, wie sich mit Formatvorlagen und der Gliederungsfunktion eine sicherere Nummerierung für Listen erzeugen lässt.*

Nummerierte Listen, beispielsweise für schrittweise Handlungsfolgen, sind ein fester Bestandteil von Gebrauchs- und Betriebsanleitungen. Die erfolgreiche Umsetzung einer solchen Liste liegt in der Gliederungsfunktion, die Word für Überschriften bereithält. Insgesamt können neun Gliederungsebenen in einem Dokument oder einer Dokumentenvorlage verwendet werden. In den seltensten Fällen sind jedoch mehr als vier oder fünf Überschriftenebenen notwendig. Die restlichen, freien Gliederungsebenen helfen bei der Listennummerierung. Zunächst werden aber zwei Absatz-Formatvorlagen

benötigt, die später mit den freien Gliederungsebenen verknüpft werden.

### Festlegung der Formatvorlage

Für den „Trigger“, der festlegt, dass eine Liste bei „1.“ beginnt, erstellen Sie die Formatvorlage „NumListe-Start“. Absätze, die Sie mit „NumListeStart“ formatieren, enthalten keinerlei Inhalte. Damit dieser leere Absatz im Layout nicht weiter stört, sollten Sie eine relativ kleine Schriftgröße wählen, zum Beispiel 1 Punkt. Eleganter ist