

InDesign mit JavaScript automatisieren

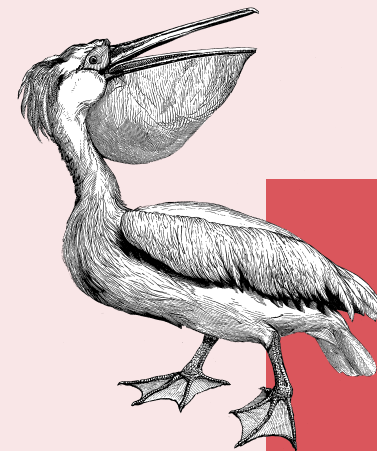
Peter Kahrel

Deutsche Bearbeitung von Martin Fischer

In jedem Layoutprozess fallen routinemäßige Aufgaben an, die sich zwar durch Handarbeit lösen lassen, aber in der Ausführung zeit- und nervenaufreibend sind. Skripten können viele dieser Aufgaben übernehmen, doch wie stellt man es an, Aufgaben zentral zu bewerkstelligen, wenn man noch nie geskriptet hat? Dieser TecFeed gibt Ihnen sowohl das Wissen als auch die Skripten an die Hand, mit denen Sie den Layoutprozess in InDesign optimieren können. Programmierkenntnisse werden nicht vorausgesetzt, Sie sollten sich aber mit InDesign CS2 bzw. CS3 gut auskennen.

Zunächst werden Sie mit den technischen Grundlagen vertraut gemacht, die Sie zum Skripten von InDesign brauchen. Sie erfahren dann, wie Sie per Skriptsteuerung Text einfügen oder löschen und Texte mit einem bestimmten Zeichen- oder Absatzformat ansprechen. Ein großer Vorteil der Skriptsteuerung liegt auf dem Feld des Suchens und Ersetzens, auf dem ein weiterer Schwerpunkt dieses TecFeeds liegt. Zu guter Letzt lernen Sie die mächtigen Möglichkeiten kennen, Tabellen per Skript anzusprechen und zu bearbeiten.

O'REILLY®



INHALT

Das ExtendScript Toolkit (ESTK) | 3

Das Objektmodell von InDesign | 5

Schnelleinstieg in JavaScript | 14

Mit Text arbeiten | 31

Suchen und Ersetzen | 39

Tabellen | 52

TecFeeds

www.tecfeeds.de

Wenn Sie InDesign mit JavaScript automatisieren wollen, müssen Sie sich mit zwei Dingen vertraut machen: mit dem Objektmodell von InDesign und mit JavaScript. Obwohl beide komplex sind, kann jeder ziemlich zügig damit anfangen, Skripten zu schreiben, wenn die ersten Hürden überwunden sind. Dieses PDF möchte zeigen, dass eine Menge nervtötender Aufgaben in InDesign mit sehr einfachen Skripten (manchmal selbst mit Einzeilern) automatisiert werden können. Diese kleinen Skripten können den Weg zu schwierigeren Skripten ebnen. Was Sie vor allem brauchen, ist Entschlossenheit.

Dieser TecFeed ist für Anwender geschrieben, die sich in InDesign gut auskennen, aber nicht notwendigerweise viel vom Skripten bzw. Programmieren verstehen. Kenntnisse in InDesign sind notwendig; schließlich gibt es wenig Ansatzpunkte zum Skripten von InDesign, wenn Sie InDesign nicht kennen. Programmierkenntnisse sind nicht entscheidend (obwohl sie natürlich von Vorteil wären). Ich bin überzeugt, dass jeder bis zu einem gewissen Grad programmieren lernen kann. Sie müssen kein Mathematiker sein, um sich solche Fertigkeiten anzueignen. Nebenbei hat das Entwickeln von JavaScript-Skripten für InDesign nichts mit Informatik zu tun: Es hat vielmehr ganz pragmatisch damit zu tun, dass etwas in InDesign zum Laufen gebracht werden soll.

Behandelt werden die Versionen Adobe InDesign CS2 und CS3.

Dieser TecFeed ist wie folgt gegliedert. Er fängt an mit einem kurzen Überblick über das ExtendScript Toolkit (ESTK), die Umgebung, in der Sie Skripten schreiben können. Wir packen hier so viel hinein, wie Sie brauchen, um damit verständlich

umgehen zu können. Der nächste Abschnitt handelt vom Objektmodell von InDesign, der Ihnen einen Aufriss nebst allgemeinen Regeln und einigen Veranschaulichungen von Eigenschaften und Methoden bietet. Danach folgt eine Einführung in JavaScript. Dies ist kein vollständiger JavaScript-Lehrgang, sondern behandelt die wichtigsten Grundbestandteile der Sprache und gibt Ihnen für den Anfang ein paar Beispiele an die Hand.

Die letzten drei Abschnitte widmen sich speziellen Bereichen, in denen Skripten nützlich sind, um ein paar Lücken in InDesign zu füllen. Diese behandeln im Wesentlichen Text. Der erste dieser Abschnitte handelt von einigen grundlegenden Textskript-Techniken. Darauf folgt ein Abschnitt, der in verschiedene Aspekte des Suchens und Ersetzens einführt. Ich werde zunächst zeigen, wie damit der Suche/Ersetze-Dialog von InDesign automatisiert werden kann, und anschließend vorstellen, wie die Suche dazu verwendet werden kann, einen flexiblen Unterschneidungseditor zu skripten. Schließlich nimmt der letzte Abschnitt Tabellen in Augenschein. Obwohl die Tabellen von InDesign recht leistungsfähig sind und die Tabellen- und Zellenformate in InDesign CS3 die Formatierung flexibler gestalten, müssen Sie auf der Benutzeroberfläche immer noch viele Formatierungen von Hand vornehmen, was überaus nervtötend sein kann. Aber glücklicherweise sind Tabellen nicht so schwierig zu skripten.

InDesigns Umsetzung von JavaScript ist plattformunabhängig. Ein Skript, das Sie auf einem Mac schreiben, läuft ebenfalls auf einem PC. Dieser TecFeed kann deshalb sowohl von Mac-

als auch von PC-Anwendern genutzt werden. Bei der Nennung von Tasten werde ich wo nötig sowohl die Namen auf dem Mac als auch die des PCs verwenden (Return/Enter, Strg/Befehl, Alt/Wahl).

Alle Skripten in diesem PDF sind geprüft und getestet worden und müssten wie angezeigt funktionieren. Trotzdem sollten Sie zum Ausprobieren jedes Skripts, auch derjenigen, die einfach und harmlos aussehen, immer eine Kopie Ihres Dokuments machen oder das Skript bei einer Testdatei anwenden. Testen Sie ein Skript niemals an einem ungesicherten Arbeitsdokument.

Schließlich möchte ich an ein paar Menschen, die in den letzten Jahren Ideen und Tipps mit mir geteilt haben, ein Wort des Dankes richten: Dirk Becker, Jonathan Brown, Michael Daumling, Martin Fischer, Olav Kvern, Dave Saunders, Shane Stanley, Alan Stearns und besonders an meinem alten Freund Teus de Jong. Der Autor kann per E-Mail über kahrel@kahrel.plus.com erreicht werden.

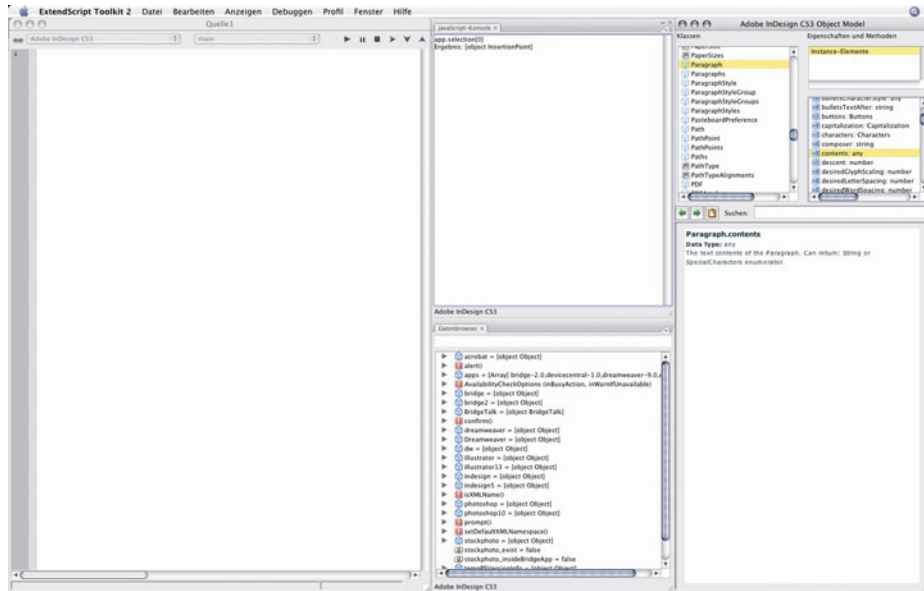
Das ExtendScript Toolkit (ESTK)

In InDesign 4 (CS2) werden Skripten im Verzeichnis `~\Adobe\InDesign CS 2\Presets\Scripts` gesichert. InDesign 5 (CS3) sind sie im Verzeichnis `~\Dokumente und Einstellungen\[Benutzername]\Anwendungsdaten\Adobe\InDesign\Version 5.0\Scripts\Scripts Panel\` (Win) bzw. im Ordner `~/Users/[Benutzername]/Library/Preferences/Adobe InDesign/Version 5.0/Scripts/Scripts Panel` (Mac) untergebracht. Skripten, die für InDesign CS2

geschrieben wurden, können auch in InDesign CS3 ausgeführt werden, wenn sie im jeweiligen Unterverzeichnis *Version 4.0 Scripts* abgelegt werden. Skripten sind reine Textdateien; sie haben immer die Endung `.jsx`. Obwohl sie in jedem Editor, der Dateien im reinen Textformat (BBEdit, Notepad) sichern kann, erfasst werden können, werden sie am besten mit dem ESTK geschrieben. Sie können einerseits direkt daraus gestartet werden, andererseits färbt das ESTK spezielle Dinge wie Kommentare und reservierte Wörter ein (Syntaxhervorhebung). Das ESTK hat viel zu bieten, aber für unser Vorhaben beschränken wir uns auf die Erklärung von ein paar Dingen.

Ein Weg, das ESTK zu starten, geht über die Skript-Palette in InDesign (*Fenster* → *Automatisierung* → *Skripten*). Dort erfolgt die Auswahl eines Skripts sowie des Befehls *Skript bearbeiten* im Fly-out Menü. Versichern Sie sich immer, dass das Bedienfeld im ESTK links oben »InDesign CS2« bzw. »InDesign CS3« anzeigt, weil der Inhalt dieses Bedienfelds dem ESTK sagt, welches Programm angesprochen wird. Das ESTK kann zum Schreiben von Skripten für die meisten Produkte von Adobe verwendet werden, und alle diese Programme, die auf Ihrem Computer installiert sind, werden in dieser Liste angezeigt. Sie können die Fenster beliebig anordnen. Der folgende Screenshots zeigt den ESTK-Startbildschirm in InDesign CS3.

Das Fenster *Quelle1* (bzw. *Datei1* in InDesign CS2) ist das Hauptfenster für das Schreiben von Skripten. Jede neue Datei, die Sie starten, wird als ein neuer Reiter (ESTK 1 für InDesign CS2) bzw. als ein neues Fenster (ESTK 2 für InDesign CS3) dargestellt.



Das Fenster mit dem Reiter *JavaScript-Konsole* ist eine Art Befehlszeile, in die Sie JavaScript-Befehle eingeben können. Das Ergebnis wird im Bedienfeld direkt darunter angezeigt. Versuchen Sie Folgendes: Schreiben Sie 'Hallo!' (mit den Anführungszeichen) und bestätigen Sie mit Enter/Return. Das gibt im Ausgabefenster Ergebnis: Hallo! aus; dieser Einzeiler könnte Ihr erstes Skript sein. Das Konsole-Bedienfeld ist in der Tat ein einfacher Texteditor. Sie können alles, was ein Skript hier ausgibt, in die Zwischenablage kopieren und irgendwo einfügen. Sie können hier ebenfalls Texte erfassen, auch wenn das nicht besonders sinnvoll ist. Um das Bedienfeld zu löschen, klicken Sie mit der rechten Maustaste hinein und wählen *Löschen* im Kontextmenü.

In einem weiteren Fenster finden Sie den praktischen Datenbrowser. Doch dazu später mehr.

Wenn Sie das ESTK zum ersten Mal starten, sehen Sie auch einen Reiter mit Skripten und einen Reiter mit dem Namen *Breakpoints*. Ich finde den Skriptenreiter nicht besonders nützlich (Sie können eine Liste der Skripten sehen, wenn Sie Strg/Befehl+O drücken oder *Öffnen* aus dem Dateimenü wählen).

Wenn Sie das ESTK über das Bearbeiten eines Skripts aus InDesigns Skriptpalette starten, wird dieses Skript rechts in einem eigenen Reiter angezeigt. Um ein neues Skript zu erstellen, drücken Sie Strg/Befehl+N.

Ein anderer Weg, das ESTK zu starten, den ich stets verwende, führt über eine Verknüpfung auf dem Schreibtisch oder im Startmenü (oder deren Äquivalente auf einem Mac). Um das ESTK zu finden, bewegen Sie sich zum Ordner `~\Adobe\Adobe Utilities\ExtendScript Toolkit`, suchen nach einer Datei namens *ExtendScript Toolkit.exe* und erstellen eine Verknüpfung. Nun können Sie das ESTK unabhängig von InDesign starten.

Wie bereits angemerkt, wird die Konsole dazu verwendet, Befehle direkt einzugeben. Aber auf meinem System ist die Schrift so klein, dass ich sie kaum lesen kann. Ferner ist diese Einzeilenbearbeitung begrenzt, deshalb verwende ich ein normales Bearbeitungsfenster für Konsolenzwecke. Um zu sehen, wie das geht, öffnen Sie ein neues Bearbeitungsfenster (Strg/Befehl+N) und schreiben 'Hallo!' hinein. Wählen Sie

Run aus dem Debugfenster (oder drücken Sie F5/Befehl+R oder klicken Sie auf den Vorwärts-Button), und die Konsole gibt Hallo! aus.

Neben den Drop-down-Menüs im Hauptfenster sehen Sie sechs kleine Buttons (das Drop-down-Menü *main* hat keine Funktion in InDesign CS2). Die ersten drei sind *Ausführen*, *Pause* und *Stop*, mit dem Sie die Ausführung eines Skripts starten, unterbrechen oder beenden können. Die letzten drei Dreiecke entsprechen *Überspringen* (Prozedurschritt), *Hineinspringen* (Einzelschritt) und *Hinausspringen* (Rücksprung) im Debugmenü. Der erste, *Überspringen*, wird benötigt, um ein Skript Zeile für Zeile auszuführen, was sehr nützlich ist, da Sie sehen können, was an jedem Punkt im Skript passiert; er heißt *Überspringen*, weil er über Funktionen hinwegspringt (Funktionen werden weiter unten im Abschnitt über JavaScript behandelt). Um auch durch Funktionen zu gehen, verwenden Sie *Hineinspringen*. Mit *Hinausspringen* verlassen Sie eine Funktion vorzeitig. Obwohl das Dokumentfenster von InDesign nicht aktualisiert wird, während Sie durch das Skript schreiten, können Sie die Werte von allen Variablen im Datenbrowser-Fenster sehen, was sehr nützlich ist. All das ist nur sinnvoll, wenn Sie Skripten schreiben, daher kehren wir später zum Debugger zurück.

Im Folgenden beschäftigen wir uns mit dem Objektmodell von InDesign und kehren dann zum ESTK-Datenbrowser zurück, um zu zeigen, wie er verwendet werden kann, um Eigenschaften und Methoden anzuzeigen, die mit Objekten von InDesign verbunden sind.

Das Objektmodell von InDesign

Das Objektmodell erkunden Sie am besten im ESTK. Öffnen Sie also das ESTK und drücken Sie Strg/Befehl+N, um eine neue Datei anzulegen. Legen Sie in InDesign ein neues Dokument an, erzeugen Sie einen Textrahmen und füllen Sie ihn mit Platzhaltertext. Platzieren Sie die Einfügemarke irgendwo im Text.

Im ESTK platzieren Sie den Cursor in dem leeren Skriptfenster, schreiben `app.selection[0]` und wählen *Ausführen* aus dem Debugmenü. (`[0]` ist ein Index; seine Bedeutung ist an dieser Stelle nicht wichtig, wird aber später klar werden.) Dieses Einzeilskript sagt Ihnen, was im InDesign-Dokument ausgewählt ist. In der JavaScript-Konsole zeigt ESTK `[object InsertionPoint]` an.

Das sagt uns, dass wir im Moment ein Objekt der Klasse »Einfügepunkt« ausgewählt haben (die Position des Cursors). Lassen Sie uns noch ein bisschen experimentieren. Gehen Sie zum InDesign-Dokument und wählen Sie ein einzelnes Zeichen aus. Kehren Sie zurück zum ESTK und wählen Sie erneut *Ausführen*. ESTK meldet `[object Character]`. Ein weiterer Versuch: Wählen Sie in InDesign ein Wort, indem Sie darauf doppelklicken; ESTK sagt Ihnen, dass die Auswahl ein Wortobjekt ist: `[object Word]`. Machen Sie weiter, indem Sie irgendwo im InDesign-Dokument dreifach klicken, um eine Zeile auszuwählen, und führen Sie das Skript mit dieser Auswahl aus: `[object Line]`. Klicken Sie viermal irgendwo in einen Absatz, und ESTK sagt `[object Paragraph]`. Schließlich wählen Sie den ganzen

Textrahmen in InDesign (Strg/Befehl+Klick) und dann *Ausführen* im ESTK; es berichtet [object TextFrame]. Sie sehen, was auch immer Sie ausgewählt haben, ESTK sagt Ihnen, was es ist (wenn Sie nichts ausgewählt haben, gibt Ihnen ESTK undefined zurück).

Bis jetzt hat uns ESTK mitgeteilt, was für eine Objektklasse unsere Auswahl war, aber vielleicht wollen wir ebenso wissen, was in diesen Objekten steckt – mit anderen Worten, was die Inhalte sind. Viele Objekte haben Inhalte; lassen Sie uns ein paar, die wir gerade gesehen haben, ausprobieren. Im InDesign-Dokument wählen Sie noch mal ein Wort durch einen Doppelklick aus. Gehen Sie ins ESTK, fügen Sie `.contents` an `app.selection[0]` an, so dass es `app.selection[0].contents` lautet, und wählen Sie *Ausführen*. Wie Sie sehen können, gibt Ihnen das ESTK nun den Inhalt des Wortobjekts als Text zurück. Versuchen Sie dasselbe mit dem ausgewählten Textrahmen, und das ESTK zeigt Ihnen den gesamten Inhalt des Textrahmens. Es ergibt durchaus Sinn, dass ESTK mit nichts antwortet, wenn Sie einen Einfügepunkt auswählen (d.h., Sie platzieren den Cursor irgendwo im Text) und nach seinem Inhalt fragen. Tatsächlich antwortet es sogar richtig literarisch mit nichts in der Art von *Alice im Wunderland*, aber Sie können es – wenig überraschend – nicht sehen.

Lassen Sie uns jedoch mit unseren Ausführungen zum Objektmodell, das wir früher als eine hierarchische Struktur beschrieben haben, weitergehen. Es ist bezeichnend für hierarchische Modelle, dass alle Elemente Verzweigungspunkte oben und unten haben – das sind Eltern (parents) und Kinder (children) –,

natürlich mit Ausnahme der obersten Verzweigungspunkte (die keine Eltern haben) und der untersten Verzweigungspunkte (die kinderlos sind). Die Eltern sind in InDesign einfach zu finden. Aber die Kinder sind ein bisschen schwieriger zu ermitteln.

Eltern

Wählen Sie in einem InDesign-Dokument einen Einfügepunkt aus. Dann löschen Sie im ESTK-Skriptfenster `.contents` und wählen *Ausführen*, um das Einzeilskript auszuführen, damit Sie sicher sein können, dass das Objekt, das Sie ausgewählt haben, ein Einfügepunkt ist. Nun hängen Sie `.parent` an `app.selection[0]` an, so dass das ESTK-Fenster nun die folgende Zeile zeigt:

```
app.selection[0].parent
```

Wählen Sie *Ausführen*, und das ESTK meldet, dass [object Story] Elternelement von unserem Einfügepunkt ist. Fügen Sie noch mal `parent` an:

```
app.selection[0].parent.parent
```

Das sagt Ihnen, dass das Elternelement eines Textabschnitts das Dokument ist. Hat ein Dokument ein Elternelement? Ja freilich. Die folgende Zeile

```
app.selection[0].parent.parent.parent
```

veranlasst das ESTK, [object Application] zurückzumelden. Sie können noch weitere Eltern hinzufügen, aber sie werden