

AppleScript – RagTime – MySQL

Erstellt: Donnerstag, den 19. August 2010

Gedruckt: Donnerstag, den 19. August 2010

Quellen und Informationen

Eine ausführliche Einführung zu RagTime und AppleScript steht auf

<http://experts.ragtime.de/experts/node/2485>

Die Informationen zu der Script-Library für MySQL-Zugriff mit Bordmitteln steht unter

<http://www.j-schell.de/node/77>

Weitere Script-Beispiele für RagTime sind:

http://experts.ragtime.de/experts/taxonomy_menu/1/3/30

Ablauf

Basis-Aktionen in RagTime

Script »Bilder platzieren«

Script »Bilder platzieren« fragt nach einem Ordner, in dem sich PNG-Bilder befinden. Es legt ein neues RagTime-Dokument mit einer Seite an und platziert die im Ordner gefundenen PNGs links auf der Seite. Die Rahmen sind je in den Proportionen an die Bilder angepasst, die Skalierung erfolgt so, dass alle Bilder untereinander auf die Seite passen.

Script-Abschnitte:

Zuerst wird mit `choose folder` ein Ordner gewählt, es wird eine Liste mit den Namen aller `.png`-Dateien in diesem Ordner aufgebaut (System Event scripting).

Ein RagTime-Dokument wird angelegt, dass aus einem Layout besteht.

Die Bilder werden in das Dokument geladen. Dabei werden die Höhen und Breiten ausgelesen, um für weitere Aktionen die Gesamthöhe zu finden, und die maximale Breite (im Script bislang aber nicht benutzt).

AppleScript – RagTime – MySQL

Erstellt: Donnerstag, den 19. August 2010

Gedruckt: Donnerstag, den 19. August 2010

Aus der Seitenhöhe und der Höhe aller Bilder wird ein Skalierungsfaktor gebildet.

In einer Schleife wird für jedes Bild ein Rechteck angelegt, die Abmessungen ergeben sich aus Höhe und Breite des jeweiligen Bilds und dem Skalierungsfaktor. In jedem angelegten Rahmen wird das zugehörige Bild platziert.

Script »Sterne zeichnen«

Beispiel für ein interaktives Script: In einem RagTime-Dokument wird ein Grafikobjekt ausgewählt, vorzugsweise eine Ellipse. Das Script fragt nach, wie viele Ecken der Stern haben soll, und zeichnet ihn als transparentes Polygon über das ausgewählte Objekt. Dabei werden Drehungen, Skalierungen und Scherungen des Objekts berücksichtigt.

Die Details des Scripts spielen hier nicht alle eine Rolle. Aber folgende Hinweise:

```
set rt_color_names to (name of every named color)...
```

Beispiel für Umgang mit Style Sheets. Hier werden alle bestehenden Farbnamen ausgelesen. Ist einer der an Anfang definierten Namen für transparente Farben vorhanden, wird diese benutzt. Anderenfalls wird keine Füllung gesetzt.

```
tell window 1...
```

Eigenschaften des ausgewählten Objekts werden ausgelesen. Die Anweisungen gehen an window 1 (statt document 1), weil Auswahl eine Eigenschaft von Fenstern ist, nicht aber von Dokumenten.

Alles ist mit try gekapselt, da viele mögliche Auswahlen (ein Buchstabe zum Beispiel) die Aktionen nicht sinnvoll erlauben würden.

```
set the_angle to 2 * pi / the_points...
```

Der wohl abstrakteste Teil des Scripts: Einrechnen der Transformation des ausgewählten Objekts in die zu erzeugenden Punkt-Koordinaten.

```
if the_points mod 2 = 1 then...
```

Es gibt zwei getrennte Codes für Sterne mit gerader und ungerader Eckenzahl. Bei einer ungeraden Zahl wird immer ein Stern aus einer Linie gezeichnet.

Anderenfalls werden mehrere Polygone gezeichnet (so wenige, wie für die Eckenzahl möglich)

```
set point_num to my_lowest_divider(the_points)...
```

AppleScript – RagTime – MySQL

Erstellt: Donnerstag, den 19. August 2010

Gedruckt: Donnerstag, den 19. August 2010

ermittelt im Zweig »gerade Anzahl von Punkten« die Anzahl der Eckpunkte für jedes Einzelpolygon. Anschließend wird die Anzahl der nötigen Polygone ermittelt.

```
on error...
```

Irgend eine Meldung, die letztlich zu allem passen muss, was schief gehen kann.

An dieser Stelle ist eine Liste von Listen erzeugt, die für jedes Polygon (ggf. nur eines) die Liste der Punkte enthält. Jedes der Polygone wird gezeichnet.

```
tell selection...
```

Ziel der Anweisungen ist selection, also das ausgewählte Objekt. Das wird auf die geeignete Zeichenfläche hochgereicht. Damit werden die Polygone auf der richtigen Seite oder in der richtigen Zeichnungskomponente angelegt.

```
if color_found then...
```

nimmt dabei wieder auf, ob eine transparente Farbe im Dokument gefunden wurde.

```
-- basic trigonometry...
```

Es folgen ein paar Handler, die man braucht, um Koordinatentransformationen (skalieren, rotieren, scheren) in der Ebene vorzunehmen.

Für sin(angle) hab ich die Taylor-Reihe einfach aus meinem uralten Taschenbuch der Mathematik von Bronstein/Semendjajew abgeschrieben.

MySQL-Zugriff

Für sich genommen hat der Abschnitt mit RagTime nichts zu tun. Das Script ist in der Tat hervorgegangen aus einem Projekt, in dem FileMaker-Daten mit einer MySQL-Tabelle abgeglichen werden mussten. Das betreffende Script läuft jetzt seit vielen Jahren in einer Endlosschleife, ohne zu murren.

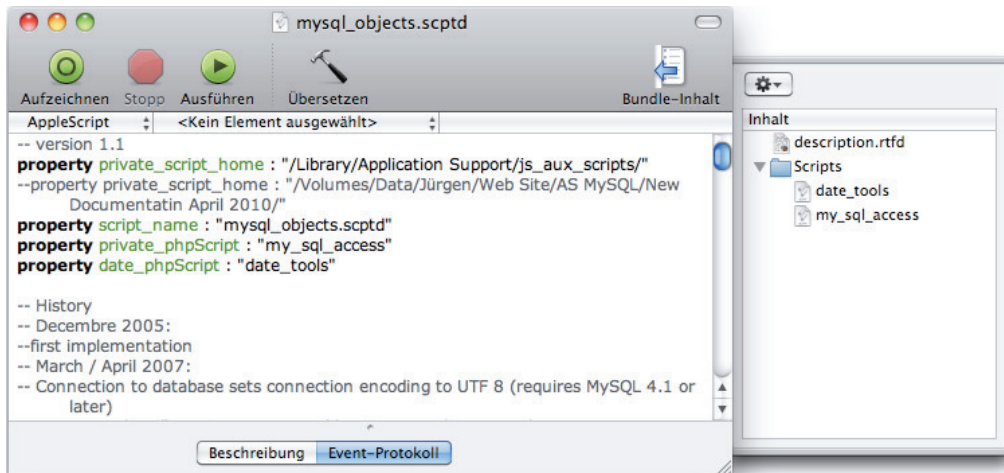
Ordner »js_aux_scripts«

Dieser Ordner muss in /Library/Application Support gelegt werden. In dem Ordner liegt das AppleScript mit den Handlers, die die Verbindung zum MySQL-Server abwickeln.

AppleScript – RagTime – MySQL

Erstellt: Donnerstag, den 19. August 2010

Gedruckt: Donnerstag, den 19. August 2010



Die Abbildung zeigt das Script-Bundle mit ausgefahrenem Bundle-Inhalt-Slider. Die Dateien »my_sql_access« und »date_tools« sind PHP-Scripts, die die eigentliche Arbeit machen. Das AppleScript verdeckt die ganzen do shell script Anweisungen, damit mit den SQL-Anweisungen wie in einem normalen AppleScript umgegangen werden kann.

Außer dem Zugriff auf den MySQL-Server stellt die kleine Bibliothek noch diverse Helfer-Funktionen. Umwandlung eines AppleScript-Datumsobjekts in einen MySQL-kompatiblen String ist eine davon. Listen von Feldnamen und -werten bequem zu einer Form wie »`feld_1` = 'abc', `feld_2` = '234',...« zu bringen, eine andere.

Script »testing_installation_en«

Das Testscript für die Installation. Öffnet man es und lässt es laufen, müssen Daten zurück kommen. Wenn ja, ist die Installation definitiv lauffähig.

Das Script verbindet sich mit dem MySQL-Server, den die RagTime.de Sales GmbH als Demo-Server für das Produkt RagTime Connect betreibt. Auf diesem Server darf aus nahe liegenden Gründen nicht geschrieben werden, aber zum Experimentieren in unserem Projekt ist er sehr gut geeignet. (Das Design der RagTime Connect-Beispiele stammt von mir, ich bin da nicht ganz neutral.)

Das Testscript enthält auch die allerwichtigsten Grundlagen für die Bibliothek:

Die ersten drei Zeilen laden die Script-Bibliothek ein eine Variable namens »DB_Library«. Zeilen dieser Art enthält jedes Script, dass diese MySQL-Verbindung benutzt.

```
set link to new_link("download.ragtime.de", "ragtime", "", "en-geography") of DB_Library
```

AppleScript – RagTime – MySQL

Erstellt: Donnerstag, den 19. August 2010

Gedruckt: Donnerstag, den 19. August 2010

erzeugt ein Objekt namens »link«, dass die Verbindungsparameter zum Server kennt (URL, Benutzername, Kennwort und Namen der Datenbank).

Im Folgenden wird eine Beispielabfrage angelegt (Variable »query_strg«) und diese mit dem Handler »do_select« in »link« abgeschickt. Das Resultat ist ein AppleScript-Record, der die Anzahl der gefundenen Datensätze, die Anzahl der Felder pro Datensatz und die Daten enthält. (»rows« und »cols« gibt es ein zweites Mal als »num_rows« und »num_cols«. Diese Varianten existieren nur, weil mit Namen wie »rows« bisweilen Namenskonflikte auftreten. In RagTime zum Beispiel.)

RagTime und MySQL-Zugriffe

Dokument »Städte suchen.rtd«

Die eine Seite des Dokuments hat vier Rahmen: Zwei Tabellen und zwei Knöpfe. Die Tabelle oben links hat ein Eingabefeld für einen ISO-Ländercode.

Zum Testen: Folgende Codes existieren in der MySQL-Tabelle.

USA	GBR
NLD	FRA
JPN	DEU
IRL	CZE

Der Knopf »Daten holen« führt ein im Dokument gespeichertes AppleScript aus, dass eine SQL-Abfrage an den Server schickt und das Resultat in die große Tabelle auf der Seite schreibt.

Das Script »Script aus »Städte suchen« ist identisch mit dem im RagTime-Dokument abgelegten. Man kann es zum Test auch vom Script Editor aus laufen lassen. In den »Antworten« sieht man dann die ganze Kommunikation des Scripts mit der MySQL-Bibliothek und dem RagTime-Dokument.

Öffnen Sie einmal im RagTime-Dokument das Rechenblatt für die Eingabe in einem eigenen Fenster (rechts-Klick in die Tabelle und »Komponente öffnen«).

In Zelle A10 wird die ganze SQL-Abfrage ausgerechnet. Sie besteht aus zwei konstanten Teilen (Zellen A5 und A7), sowie der Eingabe.

In Zeile

```
set query_strg to cell 10 of column 1
```

AppleScript – RagTime – MySQL

Erstellt: Donnerstag, den 19. August 2010

Gedruckt: Donnerstag, den 19. August 2010

ließt das Script die Abfrage aus.

```
set the_result to do_select(query_strg) of link
```

führt die Abfrage aus.

```
repeat with X from 1 to num_rows...
```

ist eine doppelte Schleife, die die gelieferten Werte in den ersten drei Spalten der Tabelle einträgt.

Fließkommazahlen kommen mit Punkt als Dezimaltrenner von MySQL zurück. Das könnte man im Script ändern, aber ich hab RagTime so eingestellt, dass Punkte akzeptiert werden (Extras → Grundeinstellungen → Berechnung).

Der Knopf »Löschen« könnte ein AppleScript nutzen, hier wird aber eine Formel der Tabellenkalkulation genutzt, die einen Bereich löscht.

Dokument »Bilder aus DB.rtd«

Dieses Beispiel verknüpft das Beispiel Bilder platzieren mit der SQL-Abfrage. Der Knopf »Bilder holen« ließt aus der Tabelle oben rechts eine SQL-Abfrage aus, die vom Server eine Sequenz von Bildern und Namen holt.

Das Script, das der Knopf benutzt, ist identisch mit »Script »Bilder aus DB««.

```
set query_strg to cell 10 of column 1
```

ließt die Abfrage aus dem RagTime-Dokument aus.

```
set the_result to fetch_binaries(query_strg, tf, "<<NAME>> <<NUM>>.jpg") of link
```

führt die Abfrage aus. Der Handler »fetch_binaries« ist speziell für die Abfrage binärer Daten aus der Datenbank vorhanden. Die Antworten des Servers werden in einen Ordner gesichert (Argument tf). Der Name, den die Dateien dabei bekommen, wird von dem Schema »"<<NAME>> <<NUM>>.jpg"«.

Die Antwort auf diese Anweisung enthält alle Bildnamen, die der Handler gesichert hat. Diese werden mit

```
copy item 1 of item X of (record_data of the_result) to end of file_names
```

in einer Schleife in eine Liste namens »file_names« geschrieben.

Die Bilder werden in dem Dokument mit derselben Strategie eingesetzt, wie im

AppleScript – RagTime – MySQL

Erstellt: Donnerstag, den 19. August 2010

Gedruckt: Donnerstag, den 19. August 2010

Script »Bilder platzieren«.

Ein kleiner Trick: in der Zeile

```
make new rectangle at beginning...
```

bekommen die Rahmen je den Namen »zu_löschen«. Das macht es wesentlich einfacher, sie anschließend mit einem Script wieder gezielt zu löschen (Script »Script »Bilder Löschen««).

Script »RagTime Suche Länder«

Dieses Script ist ein altes Beispiel aus meinem Fundus. Es muss ein RagTime-Dokument mit einem Rechenblatt namens »Rechenblatt 1« angelegt werden. In Zelle B1 tippt man den Anfang eines Ländernamens (»B« zum Beispiel) ein und lässt das Script laufen. In den Spalten ab D werden Daten zu den Ländern eingetragen.

An diesem Script interessiert hier besonders ein Aspekt, den die bisherigen Beispiele ignorieren: Sicherheit!

```
set search_val to escape_string(search_val) of db_link
```

ist die spannende Zeile. Der Handler »escape_string« sichert einen Text (oder eine Liste von Texten) mit Escape-Zeichen so ab, dass keine SQL-Injektionen mehr möglich sind. Dieser Schritt ist entscheidend bei allen Daten, die aus einer Anwender-Eingabe stammen. Der Handler benutzt die PHP-Funktion »mysql_real_escape_string«, die mit dem Datenbank-Server Kontakt aufnimmt, um Details zu den nötigen escape-Regeln zu erhalten. Die Funktion sollte recht sicher sein.