

Dokumentation AppleScript-MySQL-Werkzeuge

created: Saturday, May 1 2010

printed: Thursday, August 19 2010

Installation

Das Script »mysql_objects.scptd« muss in den Ordner »/Library/Application Support/ js_aux_scripts« kopiert werden. Wenn Sie einen anderen Ort benutzen, muss die Eigenschaft »private_script_home« im Script entsprechend angepasst werden.

Script-Aufruf

Laden Sie das Script und weisen es einer Variablen zu. Beispiel:

```
set library_path to "js_aux_scripts:mysql_objects.scptd"
set app_support to (path to application support folder) as string
set DB_Library to (load script file (app_support & library_path))
```

Basis-Technik

»mysql_objects.scptd« ist ein Bundle, das ein AppleScript und PHP-Scripts enthält. »do shell script« wird genutzt, um vieles PHP zu überlassen, statt es in AppleScript selbst zu erledigen.

Handlers und Properties des Scripts

as_objecs.scpd

Hinweis: Das Objekt oberster Ebene als eigene Datei, zum Laden und Zuweisen zu einer Variablen.

```
new_link(server_url, user_name, user_password, database)
```

Resultat: Script

Dokumentation AppleScript-MySQL-Werkzeuge

created: Saturday, May 1 2010

printed: Thursday, August 19 2010

Zweck: Objekt mit allen nötigen Informationen zur Datenbankverbindung erzeugen

Diese Funktion erzeugt ein neues Verbindungsscript zu einem MySQL-Datenbankserver. `server_url` kann eine Portnummer in der Form `adresse:port_nummer` spezifizieren. Erzeugen des Objekts baut selbst keine Verbindung zum Server auf.

new_converter()

Resultat: Script

Zweck: Werkzeuge für diverse Umwandlungen anbieten

Die Funktion erzeugt ein neues Script mit Umwandlungswerkzeugen zwecks einfacherer Handhabung unterschiedlicher Repräsentation von Daten.

new_helper()

Resultat: Script

Zweck: Vorwiegend intern

Die Funktion erzeugt ein neues Script mit Hilfsfunktionen, die normalerweise nicht für den direkten Aufruf gemeint sind.

link Objekt

do_select(select_statement)

Resultat: Record

Zweck: Ausführen einer SELECT-Anweisung

Die Funktion sendet eine SQL-Anweisung an den Server. Sie liefert einen Record mit `rows`, `cols` und `record_data`. `record_data` ist eine Liste von Listen, die die Tabelle vom Server repräsentiert.

Der Ergebnisrecord hat die Elemente `rows`, `cols` und `record_data`. `rows` ist die Anzahl der Zeilen, die die Abfrage lieferte oder `-1`, wenn ein Fehler auftrat. `cols` ist die Anzahl der Spalten in der Ergebnistabelle.

`record_data` ist eine Liste von Listen für die Zeilen des Abfrageresultats. Im Falle eines Fehlers enthält sie eine Liste mit zwei Elementen: Fehlercode und Fehler-text.

do_query(sql_statement)

Resultat: Record

Dokumentation AppleScript-MySQL-Werkzeuge

created: Saturday, May 1 2010

printed: Thursday, August 19 2010

Zweck: Senden einer SQL-Anweisung, die keine Daten liefert (INSERT, UPDATE...)

Die Funktion führt eine SQL-Anweisung aus und nimmt an, dass sie kein SELECT ist. Der gelieferte Record hat die Elemente rows, insert_id, error_code, error:string.

rows ist die Anzahl der von der Query betroffenen Zeilen.

insert_id ist der Autoinkrement-Wert bei einem INSERT. (0, wenn dieser nicht existiert.)

error_code ist 0, wenn kein Fehler auftrat.

error_string ist leer, wenn kein Fehler auftrat.

fetch_binaries(select_statement, folder_path, file_name_pattern)

Resultat: Record

Hinweis: Schreibt Daten auf die Festplatte

Zweck: Binärdaten vom Server holen

Die Funktion führt eine SQL-Anweisung aus und nimmt ein SELECT an, dass Binary Large Objects (BLOBs) in der ersten Spalte liefert. Die zweite Spalte (wenn vorhanden) wird angenommen als ein Text, der für die Dateinamen nützlich ist.

Für jeden gefundenen Datensatz wird der BLOB-Inhalt in einer Datei in dem von folder_path angegebenen Ordner gesichert. Der jeweilige Dateiname wird von file_name_pattern festgelegt. file_name_pattern ist ein beliebiger Text, der bestimmte Platzhalter enthalten kann:

<<NUM>> Der NUM-Platzhalter wird durch die Nummer der Datei in der Ergebnistabelle ersetzt. Zahlen werden links mit 0 aufgefüllt, entsprechend der größten Nummer der gefundenen Datensätze. Beispiel: 8 bei weniger als zehn Datensätzen, 08 bei weniger als 99 etc.

<<NAME>> Der NAME-Platzhalter wird durch den Inhalt der zweiten Spalte der Ergebnistabelle ersetzt. Er wird nicht ersetzt, wenn die Abfrage nur eine Spalte liefert.

<<DATE>> Der DATE-Platzhalter wird durch das aktuelle Datum im Format wie 2011-05-25 ersetzt.

<<TIME>> Der TIME-Platzhalter wird durch die aktuelle Uhrzeit im Format wie 19.20.54 (Stunden, Minuten, Sekunden) ersetzt.

DATE und TIME sind die Werte unmittelbar vor dem Sichern der ersten Datei. Sie sind für alle Werte in der Liste gleich.

Die Funktion überschreibt stillschweigend bestehende Dateien. Wenn mehr als ein Datensatz zurückgeliefert wird, muss mindestens der Platzhalter <<NUM>>

Dokumentation AppleScript-MySQL-Werkzeuge

created: Saturday, May 1 2010

printed: Thursday, August 19 2010

oder <<NAME>> benutzt werden, um unterschiedliche Dateinamen zu erzeugen.

Der Ergebnis-Record hat die Einträge rows, cols und record_data. rows ist die Anzahl der Zeilen, die die Abfrage lieferte oder -1 wenn ein Fehler auftrat. cols ist immer 3, wenn die Abfrage erfolgreich war. record_data ist eine Liste von Listen. Im Erfolgsfall: Eine für jede gesicherte Datei. Jede Liste hat 3 Elemente, in dieser Reihenfolge: Nummer der Datei im Datenbank-Ergebnis, Wert der zweiten Spalte und Name der Datei.

Im Fehlerfall hat die Liste zwei Elemente: die Fehlernummer und den Fehler-text.

escape_string(string/string_list)

Resultat: String oder Liste von Strings (entsprechend dem Argumenttyp)

Zweck: Strings vorbereiten, die in SQL sicher sind

Wendet die notwendigen Escape-Techniken auf einen String an, um ihn in einer SQL-Anweisung zu benutzen. Wird eine Liste von Strings übergeben, kommt eine Liste von Strings mit den Escape-Zeichen zurück. Die Methode prüft die Datenbank für nötige Escape-Strategien. Diese Methode reagiert nicht auf die Property raise_errors.

converter object

float2mySQL(theFloat)

Resultat: string

Zweck: Make number strings independent from locale

The function makes sure the decimal separator in the string returned is a dot, not a comma.

Note that MySQL accepts strings like "1.23E+5" (=123000) for integers. So this function may be used for the types INT and BIGINT as well.

date_convert(the_date, resulttype)

Resultat: string

Zweck: Create string representation of date object in one of several versions.

Converts an AppleScript date to a string acceptable for the database. Possible result types are unix, sql_date, sql_time and sql_datetime.

Dokumentation AppleScript-MySQL-Werkzeuge

created: Saturday, May 1 2010

printed: Thursday, August 19 2010

UNIXtime(the_date)

Resultat: string

Zweck: Create UNIX timestamp

Converts an AppleScript date to a Unix time stamp (same as `date_convert(the_date,"unix")`).

SQLdate(the_date)

Resultat: string

Zweck: Create SQL compatible date representation

Converts an AppleScript date to a SQL date string (same as `date_convert(the_date,"sql_date")`).

SQLtime(the_date)

Resultat: string

Zweck: Create SQL compatible time representation

Converts an AppleScript date to a SQL time string (same as `date_convert(the_date,"sql_time")`).

SQLdatetime(the_date)

Resultat: string

Zweck: Create SQL compatible date / time representation

Converts an AppleScript date to a SQL date / time string (same as `date_convert(the_date,"sql_datetime")`).

new_date(theyear, themonth, theday)

Resultat: date

Zweck: Create date object not depending on locale settings

Creates an AppleScript date out of three integers. The time part is 0.

If theyear is a two digit number, it will use 2000, if theyear is less than 75. Else 1900 will be used.

(Change the property `Threshold2000` if you need another threshold.)

new_time(theyear, themonth, theday, thehour, theminute, thesecond)

Resultat: date

Zweck: Create date object not depending on locale settings, including time

Creates an AppleScript date out of six integers. Two digit year behaviour is as in `new_date`.

Dokumentation AppleScript-MySQL-Werkzeuge

created: Saturday, May 1 2010

printed: Thursday, August 19 2010

string2date(datestring, formatstring)

Resultat: date

Zweck: General tool to convert date strings to date objects

Converts datestring to an AppleScript date, using formatstring to interpret groups of digits. Possible characters in formatstring are: Y for year, M for month, D for day, h for hour, m for minute, s for second. The groups of digits in datestring are extracted first. Each group is interpreted as an integer to read as a date component as given by the formatstring character in the same position. string2date("5-3-2010","MDY") is May the 3rd of 2010 . string2date("5-3-2010","DMY") is March the 5th of 2010.

If any of the format characters M, D, Y is missing, the default is given be the current date. The default for time components is 0. If a format character appears more than once, the last occurrence wins.

If the year has one or two digits only, the same rules as in new_date apply.

If formatstring contains any character not in the given lists, it will ignore the group of digits in its position. We suggest to us B or b like bitbucket.

merge_names_values(name_list, value_list)

Resultat: string

Zweck: Easy creation of SQL statements from field name / value lists

Takes a list of table field names and a list of field values to create a string as it typically appears in UPDATE or INSERT statements. Names are enclosed in backquotes, values are enclosed by single quotes. Name and value are connected by =, all parts by commata. merge_names_values({"first","second","age"}, {"John","Doe",25}) returns `first` = 'John', `second` = 'Doe', `age` = '25'. Note that MySQL does allow the quotes for numbers.

merge_lists_2(outer_glue, inner_glue, leftlist_quote, rightlist_quote, leftlist, rightlist)

Resultat: string

Zweck: partly internal. Merging lists of names and values

More general handler, merge_names_values is based on. Values in leftlist are surrounded by leftlist_quote, rightlist values are surrounded by rightlist_quote, leftlist and rightlist strings are connected by inner_glue and the entire sequence by outer_glue. merge_names_values({"first","second","age"}, {"John","Doe",25}) is just shorthand for merge_lists(" ", " = ", "'", "'", {"first","second","age"}, {"John","Doe",25})

merge_lists(outer_glue, inner_glue, rightlist_quote, leftlist, rightlist)

Resultat: string

Zweck: Legacy version of merge_lists_2

created: Saturday, May 1 2010

printed: Thursday, August 19 2010

Legacy version of preceeding to stay compatible with older code. Simply calls merge_lists_2 with empty leftlist_quote.

Script helper object

Note: Parent of the other two object types. Contains basic tool functions used in several places

link_text(glue, list_of_parts)

Resultat: string

Zweck: Concatenate elements of list with some separator

Combines strings in the list_of_parts with glue as a seperator. Avoids double separators and removes them at beginning and end. Typical use is path construction.

list2command(string_list)

Resultat: none

Zweck: Tool to prepare strings for shell script commands

Applies the quoted form property to a list of strings for use in shell script commands.

SAR(haystack_text, needle_text, replace_text)

Resultat: string

Zweck: Reasonably fast search and replace in a string

Search and replace (case sensitive). Searches haystack for every occurance of needle and replaces needle by replace_text

split2list(the_strg)

Resultat: List

Zweck: Regard as private.

explode(delimiter, the_strg)

Resultat: List

Zweck: Split some string into a list

Turns the_str into a list, using delimiter for splitting

Property raise_errors

Zweck: activate standard AppleScript errors

Dokumentation AppleScript-MySQL-Werkzeuge

created: Saturday, May 1 2010

printed: Thursday, August 19 2010

Default is false. If true, problems with database access in `do_select` and `do_query` will raise an error rather than returning the error in a record. Error value and number are the same as the values for records.

Error Codes

Codes returned by the MySQL server start at 1000. A list can be found at

<http://dev.mysql.com/doc/refman/5.0/en/error-messages-server.html>

Own errors in the connection to the SQL server start at 5000:

5002: Connect to database server failed

5003: Selecting database on server failed