



# **Das Java Paket *SQLite* für die Eingabe und Verwaltung von Beständen mit dem ForestSimulator**

Jürgen Nagel

Abt. Waldwachstum der Nordwestdeutschen Forstlichen Versuchsanstalt, Göttingen

© 2015-2016 J. Nagel,  
Nordwestdeutsche Forstliche Versuchsanstalt,  
Grätzelstr.2, 37075 Göttingen, Germany  
<http://www.nw-fva.de>

## Inhaltsverzeichnis

Einführung.....	2
Aufruf der Schnittstelle im ForestSimulator.....	2
Datenbank für TreeGrOSS Bestände und Probeflächenaufnahmen.....	2
Datenbank für Probekreisaufnahmen.....	4
Datenbank für Bestandeszeilen.....	5
Import von Daten aus Tabellenkalkulationsprogrammen.....	5
Literatur:.....	6

## Einführung

Für die Simulation von Waldbeständen mit dem ForestSimulator können diese in Form von XML-Dateien im sogenannten TreeGrOSS-Format eingelesen und gespeichert werden. Der Vorteil des XML-Formats ist, dass dieses mit jedem Texteditor gelesen und beschrieben, die Daten strukturiert und selbsterklärend abgelegt, diese leicht über das Internet ausgetauscht und mit verschiedenen Style-Sheets (XLS-Dateien) in unterschiedlicher Form dargestellt werden kann. Für die praktische Eingabe und die Bearbeitung der Daten ist das XML-Format jedoch nur bedingt geeignet und für den Praktiker zu kompliziert. Aus diesem Grunde verfügt das Programm ForestSimulator über einen Bestandeseditor der im offiziellen Handbuch im Kapitel 2.1.4.5 ausführlich beschrieben ist (Nagel und Hansen 2014).

Für eine leichtere Verwaltung und die Weiterverarbeitung der gemessenen und simulierten Bestandesdaten wurde der ForestSimulator um eine SQLite-Datenbankschnittstelle ergänzt. SQLite-Datenbanken<sup>1</sup> sind einzelne Dateien. Für sie ist keine eigene Server-Umgebung notwendig. Für Bearbeitung und Verwaltung der SQLite-Datenbanken gibt es viele kostenfreie Softwareprogramme, wie zum Beispiel das Firefox-Addon **SQLite Manager**<sup>2</sup> und das **Statistik-Paket R**<sup>3</sup>.

Die SQLite-Schnittstelle im ForestSimulator kann drei verschiedene SQLite-Datenbanken bedienen, in denen entweder die Daten von Beständen, Probekreisen oder Ansprachen von Bestandesschichten verwaltet werden können. Die Datenbanken für Bestände und Probekreise unterscheiden sich hauptsächlich in der Art, mit der die Koordinaten der Eckpunkte und Bäume gespeichert werden. In der Bestandesdatenbank werden kartesische und in der Probekreisdatenbank polare Koordinaten verwendet. Die Datenbanken mit ihren Tabellen und Feldern sowie deren Benutzung werden im folgenden beschrieben. Zu jeder der drei Datenbanken befindet sich ein Beispiel Verzeichnis **/ForestSimulator/data\_standsimulation** mit dem Dateisuffix **\*.db**.

## Aufruf der Schnittstelle im ForestSimulator

Die SQLite-Schnittstelle wird im ForestSimulator über das Menü **Bestand** → **SQLite Database** aufgerufen. Das Dialogfenster besteht aus drei Reiterkarten. Über diese werden die drei verschiedenen Datenbanken aufgerufen. Wenn keine weiteren Einstellungen vorgenommen werden,

---

1 <https://sqlite.org/>

2 <https://addons.mozilla.org/de/firefox/addon/sqlite-manager/>

3 <https://www.r-project.org/>

greift das Programm auf die Dateien im Datenverzeichnis des ForestSimulators zu. Es ist aber möglich anderen Datei den entsprechenden Dialogen zuzuordnen. Die Bestände lassen sich über die Vor- und Zurück-Knöpfe sowie die Such-Knöpfe auswählen. Nach der Auswahl sie können mit dem unteren Knopf „**load stand from database to ForestSimulator**“ in das Programm ForestSimulator geladen werden.

## ***Datenbank für TreeGrOSS Bestände und Probeflächenaufnahmen***

In dem Verzeichnis **/ForestSimulator/data\_standsimulation** befindet sich die SQLite-Datei **fsdatabase.db**. In dieser Datei sind einige Beispielbestände gespeichert, die über die SQLite-Schnittstelle des ForestSimulators geladen werden können. Darüber hinaus kann man in diese Datenbank auch die Werte von simulierten Beständen speichern. Die Struktur dieser Datenbank wird in den Abbildungen 1 bis 3 erklärt. In den Abbildungen wurde die Datenbank mit dem SQLite-Manager geöffnet. In die Datenbank soll ein kleiner Testdatensatz eingegeben werden. Nachdem der SQLite-Manager im Firefox als Add-on installiert ist, kann man diesen im Firefox-Menü unter **Extras → SQLite Manager** starten. Es öffnet sich ein neues Fenster und es kann die Beispieldatei **fsdatabase.db** im Verzeichnis **/ForestSimulator/data\_standsimulation** ausgewählt werden. Falls der Dateiauswahldialog die Datei mit der Endung **\*.db** nicht anzeigt, hilft es die Dateiauswahl von **\*.sqlite** auf alle Dateien umzustellen. Ein entsprechender Schalter befindet sich in der Maske. Wenn die Datei erfolgreich geöffnet ist, kann im linken Teil des Fensters unter **Tables** die Tabelle **stand** ausgewählt werden. Wählt man jetzt noch die Reiterkarte **Durchsuchen**, so sollte auf dem Bildschirm in etwa die Abbildung 1 dargestellt werden.

Für jeden Bestand der neu in die Datenbank eingetragen werden soll, muss nun in der Tabelle „**stand**“ eine (und nur eine) Zeile angelegt werden. Dieses kann mit dem Knopf **Daten hinzufügen** erledigt werden. Das Feld **\_id** wird automatisch aufgefüllt und ist die ID-Nummer für den Bestand. Jede Bestandes ID-Nummer kann nur einmal vergeben werden. Man müssen nun die Informationen für Pflichtfelder name = Bestandesname, size\_ha = Bestandesgröße [ha], month = Monat der Aufnahme und year = Jahr der Aufnahme eintragen werden. Alle übrigen Felder sind optional zu befüllen. Es empfiehlt sich der Wert -99 für numerische Felder, wenn deutlich gemacht werden soll, dass keine Informationen vorliegen. Nach dem die Zeile gespeichert ist, können die weiteren Informationen für die Eckpunkte und Bäume in beiden anderen Tabellen auf gleiche Weise eingetragen werden.

**Manager - /home/nagel/Dokumente/JnProgramme/ForestSimulation/data\_standsimulation/fsdatabase.db**

Verzeichnis ▶ (Profil-Datenbank auswählen) ⌵ Los

f(x)

fsdatabase.db ⌵

Struktur Durchsuchen SQL ausführen DB-Einstellungen

▼ Master Table (1)  
 ▼ sqlite\_master  
   type  
   name  
   tbl\_name  
   rootpage  
   sql

▼ Tables (4)  
   ▶ cornerpoints  
   ▶ sqlite\_sequence  
   ▶ **stand**  
     ▶ trees  
   ▶ Views (0)  
   ▶ Indexes (0)  
   ▶ Triggers (0)

TABLE stand Suchen Alle einblenden Datensatz hinzu												
_id	name	size_ha	month	year	lat	lon	masl	region	district	sitetype	exposition_gon	slope_percentage
1	Generated Stand	0.2	3	2008	0	0	-99.9				-99	-99.9
2	Gene	0.7	3	2008	0	0	-99.9				-99	-99.9
3	Mein Testbest...	0.5	3	2008	0	0	-99.9				-99	-99.9
4	Generated Stand	0.2	3	2008	0	0	-99.9				-99	-99.9
7	Generated Stand	0.2	3	2008	0	0	-99.9				-99	-99.9
9	Generated Stand	0.2	3	2008	0	0	-99.9				-99	-99.9
10	Generated Stand	0.2	3	2008	0	0	-99.9				-99	-99.9
11	Generated Stand	0.2	3	2008	0	0	-99.9				-99	-99.9
12	Generated Stand	0.2	3	2008	0	0	-99.9				-99	-99.9
13	Generated Stand	0.2	3	2008	0	0	-99.9				-99	-99.9
18	Generated Stand	0.2	3	2008	0	0	-99.9				-99	-99.9
19	Generated Stand	0.2	3	2013	0	0	-99.9				-99	-99.9
20	Generated Stand	0.2	3	2023	0	0	-99.9				-99	-99.9
21	BuP	1	3	2015	0	0	-99.9				-99	-99.9
26	Generated Stand	0.2	3	2023	0	0	-99.9				-99	-99.9
27	Generated Stand	7	3	2015	0	0	-99.9				-99	-99.9
28	Generated Stand	0.2	3	2015	0	0	-99.9				-99	-99.9
30	200a1	1	3	2015	0	0	-99.9				-99	-99.9
31	<b>Testbestand</b>	<b>0.2</b>	<b>11</b>	<b>2016</b>	<b>0</b>	<b>0</b>	<b>0</b>				<b>0</b>	<b>0</b>

Abb. 1: Tabelle „stand“ dargestellt im SQLite-Manager

In die Tabelle „**cornerpoints**“ werden die Bestandeseckpunkte eingetragen, weil der

ForestSimulator einen Raumbezug benötigt. Falls keine Koordinaten für die Eckpunkte und die Bäume bekannt sind, erzeugt man einfach eine quadratische oder rechteckige Fläche wie in der Abbildung 2 für den Testbestand mit der ID-Nummer = 31. Im Beispiel wurde für den Bestand 31 mit einer Größe von 0,2 ha ein Fläche mit Kantenlängen von 40m und 50m festgelegt. Bevor die Eckpunkte eingetragen werden, muss man einen Mittelpunkt für die Fläche eingeben. Dieser ist für eventuelle Drehtransformationen im ForestSimulator von Bedeutung. Im Beispiel wurde für diesen Punkt die Flächenmitte mit der Koordinate  $x = 20\text{m}$  und  $y = 25\text{m}$  gewählt. Man füllt die erste Zeile, wie es in der Abbildung 2 dargestellt ist und trägt unter **standid** = 31, **name** = **polygon**, **x** = 20.0, **y** = 25.0 und **z** (relative Höhe) = 0.0 ein. Anschließend werden die Eckpunkte der Fläche entgegen des Uhrzeigers eingetragen. Die Eckpunkte erhalten die Namen **ECK1**, **ECK2**, usw.. Die Anzahl der Eckpunkte kann bei unregelmäßigen Bestandesgrenzen größer als vier sein. Wichtig ist, dass bei allen Eingaben die ID-Nummer des Bestandes aus der Tabelle **stand** (im Beispiel **\_id** = 31) unter **standid** eingetragen wird und so der ForestSimulator die richtigen Eckpunkte zuordnen kann.

_id	standid	name	x	y	z
141	31	polygon	20	25	0
142	31	ECK1	0	0	0
143	31	ECK2	0	50	0
144	31	ECK3	40	50	0
145	31	ECK4	40	0	0

Abb.2: Auszug der Tabelle „cornerpoints“ mit den Eckpunkten.

In der dritten Tabelle „trees“ werden die Baumwerte übergeben. Da der ForestSimulator über umfangreiche Datenergänzungsrouitinen verfügt, müssen nicht für alle Variablen Werte vorliegen. Als Pflichtfelder müssen **code** = Baumartencode, **name** = Baumkennung im Feld, **year** = Jahr der Aufnahme, **age** = Alter, **dbh** = BHD [cm], **alive** = -1 oder die Jahreszahl wann der Baum ausgeschieden ist, **fac** = Flächenfaktor in der Regel 1.0 und die **standid** = Bestandes ID-Nummer (hier 31) ausgefüllt sein. Für mindestens einen Baum des Bestandes muss eine Höhenangabe vorliegen. Falls Koordinaten für die Bäume bekannt sind, sollten diese innerhalb der Eckpunktgrenzen liegen. Für die Felder **si** = Siteindex (Oberhöhe im Alter 100), **cb** = Kronenansatz [m], **cw** = Kronenbreite [m] und die Koordinaten **x**, **y**, **z** können sie 0 oder besser -9 eingeben, wenn die Information unbekannt ist. Unter **remarks** ist es möglich Bemerkungen einzufügen und bei **layer** kann die Bestandesschicht verschlüsselt werden. Die Baumkennung **name** kann auch Buchstaben wie zum Beispiel 100a enthalten.

_id	co...	na...	year	age	dbh	h	si	cb	cw	alive	status	x	y	z	crop	habitat	fac	remarks	standid	layer
2329	511	1	2016	50	40	25	-9	-9	-9	-1	0	-9	-9	-9	false	false	1		31	1
2330	511	1	2016	50	40	0	-9	-9	-9	-1	0	-9	-9	-9	false	false	1		31	1
2331	511	1	2016	50	20	0	-9	-9	-9	-1	0	-9	-9	-9	false	false	1		31	1
2332	511	1	2016	50	30	0	-9	-9	-9	-1	0	-9	-9	-9	false	false	1		31	1

Abb 3: Auszug der Tabelle trees mit den Baumwerten

## Datenbank für Probekreisaufnahmen

In dem Verzeichnis `/ForestSimulator/data_standsimulation` befindet sich die SQLite-Datei

**fsplots.db.** In dieser Datei sind einige Beispiele von Probekreisaufnahmen gespeichert, die über die SQLite-Schnittstelle des ForestSimulators geladen werden können. In der Datenbank können auch die Daten konzentrischer Probekreisaufnahmen eingegeben werden. Die Datenbank besteht aus den zwei Tabellen **stand** und **trees**. Die Tabelle **stand** entspricht der Tabelle der Probeflächendatenbank **fsdatabase.db** (siehe zurückliegenden Kapitel).

Die Tabelle **trees** in der Probekreisdatenbank unterscheidet sich von der in der Bestandesflächendatenbank lediglich in den Feldern **azimuth** = Azimut in Gon und **distance** = Entfernung zum Kreismittelpunkt [m]. Der Azimut gibt den Winkel auf Nord bezogen an. Falls dieser in Grad gemessen wurde, muss man ihn einfach vorher umrechnen (400 Gon = 360 Grad). Die Entfernung ist die Strecke vom Probekreismittelpunkt zur Baummitte. Alle Bäume müssen innerhalb des Probekreises stehen, der durch die Flächengröße in der Tabelle **stand** vorgegeben wird. Bei konzentrischen Probekreisen muss für Bäume, die nur auf den inneren Kreisen aufgenommen werden, der Flächenfaktor (**fac**) im Verhältnis zum äußeren Kreis angepasst werden. Im ForestSimulator werden Bäume mit einem Flächenfaktor größer oder gleich 2 um den ganzzahligen Wert geklont. Der aufgenommene Baum erhält die gemessene Koordinate. Die Koordinaten der geklonten Bäume werden im ForestSimulator automatisch ergänzt. Alle Klonbäume können ihrem Ursprungsbaum über die Baumnummer (**name**) zugeordnet werden. Beim Einlesen in den ForestSimulator werden automatisch 20 Eckpunkte erzeugt und alle Koordinaten in x und y Werte umgerechnet. Sie können daher im ForestSimulator aufgerufene Probekreise in der Bestandesflächendatenbank speichern.

## Datenbank für Bestandeszeilen

In dem Verzeichnis **/ForestSimulator/data\_standsimulation** befindet sich die SQLite-Datei **nutzungsplaner.db**. Diese Datei kann direkt aus der Android App „Nutzungsplaner für Rein- und Mischbestände“<sup>4</sup> übernommen werden. Man kann die Datenbank aber auch dafür nutzen, Forsteinrichtungszeilen, welche nach Baumart und Schicht erhoben wurden, zur Generierung von Beständen im ForestSimulator zu verwenden. In der Abbildung 4 ist die Datei dargestellt. Alle Bestandeszeilen, die den gleichen Namen aufweisen, gehören zum selben Bestand. In die Pflichtfelder **name** = Bestandesname, **art** = Baumartenkodierung, **schicht** = Baumschicht nach Forsteinrichtung (1 bis 4 oder 0 nicht angesprochen), **mischung** = Mischungstyp, **alt** = Alter, **hoehe** = Baumhöhe [m], **gha** = Grundfläche [m<sup>2</sup>/ha] müssen die entsprechenden Informationen eingespeichert werden. Alle anderen Felder können, aber müssen nicht ausgefüllt sein.

_id	name	art	schicht	mischung	alt	hoehe	gha	vha...	lat	lc
1	100a1	211	0	0	50	21.5	17.6			
2	100a1	511	0	0	50	24.5	7.6			
3	200a1	711	0	0	70	24.5	32			
4	BuP	211	1	0	150	40	3.1911			
5	BuP	211	1	0	140	38.8	3.1801000000000...			

Abb 4: Auszug der Tabelle Nutzungsplanung

## Import von Daten aus Tabellenkalkulationsprogrammen

Mit dem SQLite-Manager ist es auch möglich Daten aus Tabellenkalkulationsprogrammen wie MS Excel oder Libre Office Calc zu übernehmen. Die Datenübernahme geht am besten über eine sogenannte csv-Datei (comma separated values). Der Aufwand lohnt besonders dann, wenn in die

<sup>4</sup> Android App „Nutzungsplaner für Rein- und Mischbestand“, <https://play.google.com/store/apps/details?id=ftools.nutzung&hl=de>

Tabelle **trees** viele Bäume eingelesen werden sollen. In diesem Fall empfehle ich folgendes Vorgehen:

1. Exportieren Sie die Tabelle **trees** aus der Bestandesflächen- oder Probekreisdatenbank, in dem Sie im SQLite Manager die Tabelle **trees** anwählen und im Menü **Tabelle** → **Tabelle exportieren** wählen. Wählen Sie im Dialog die Reiterkarte **csv** und anschließend das Trennzeichen und ob die Zeichenketten umschlossen sein sollen, wie in der Abbildung 5. Klicken Sie auf **ok** und speichern Sie die Datei (hier trees.csv) in ein Arbeitsverzeichnis.
2. Öffnen Sie die unter 1) erzeugte Datei mit der Tabellenkalkulation. Löschen Sie die exportierten Daten und füllen Sie Spalten neu auf. Achtung die Spalte **\_id** muss leer bleiben. Speichern Sie die Datei im **csv-Format**.
3. Öffnen Sie mit dem SQLite Manager die Datenbank und die Tabelle **trees**. Wählen Sie im Menü **Datenbank** → **importieren** und füllen Sie den Dialog wie in Abbildung 6 aus. Drücken Sie auf **ok** und wenn Sie Glück haben, sind die Daten in der Tabelle. Falls das im ersten Anlauf nicht klappt, lesen Sie den nächsten Abschnitt.

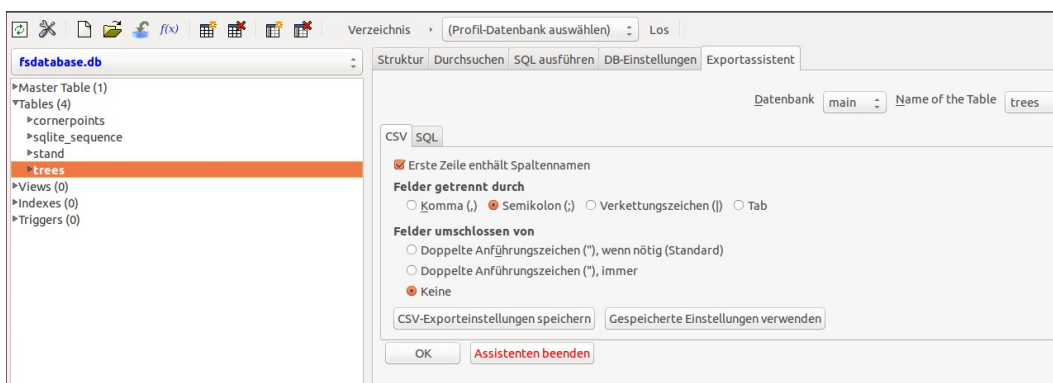


Abb. 5: Dialog zum Exportieren der Tabellen nach csv

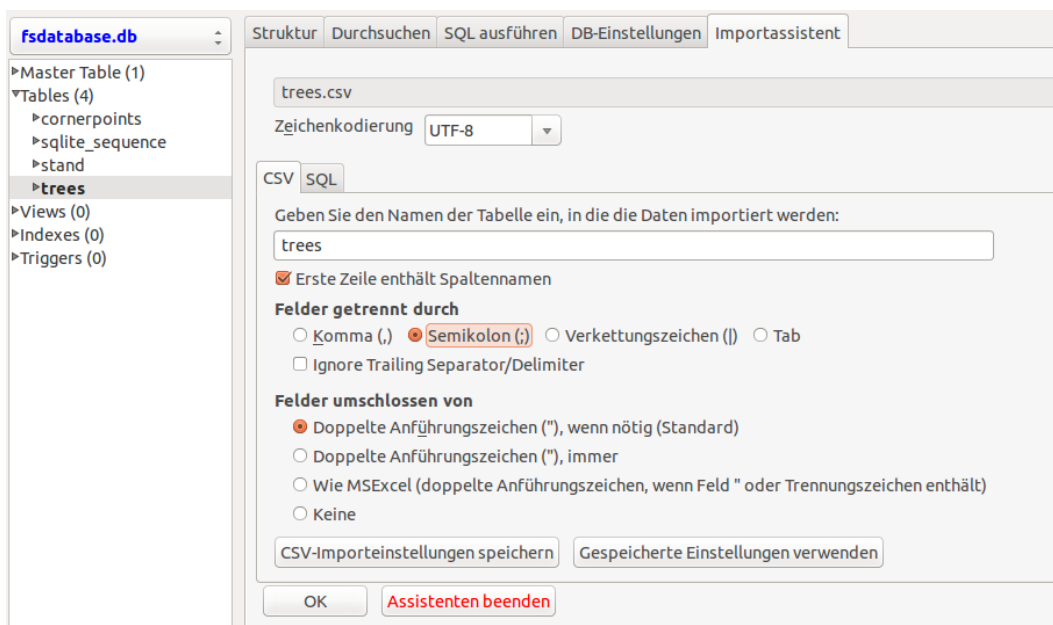


Abb. 5: Dialog zum Importieren der csv-Datei in die Tabellen

Mögliche Fehler beim Ex- und Import sind:

1. Die Daten werden in der Tabellenkalkulation nicht richtig angezeigt: Dies hat meist damit zu tun, dass das Tabellenkalkulationsprogramm nicht richtig eingestellt ist, bzw. die csv-Datei

ein anderes Dezimaltrennzeichen verwendet. In dem Fall können Sie entweder das Tabellenkalkulationsprogramm umstellen oder die csv-Datei mit einem Texteditor öffnen und mit „Suchen und Ersetzen“ alle Punkte nach Komma tauschen. Dann die Datei erneut mit der Tabellenkalkulation öffnen und wenn die Daten richtig angezeigt werden, die beiden restlichen Schritte erneut ausführen.

2. Die csv-Datei lässt sich nicht einlesen: Die Spalte **\_id** muss leer sein. Es kann sein, dass der SQLite Manager und csv-Datei das Dezimaltrennzeichen unterschiedlich verwenden. Also müssen Sie dieses in der csv-Datei mit „Suchen und Ersetzen“ berichtigen. Prüfen Sie auch, durch welches Zeichnen die Felder getrennt werden und stellen Sie das Richtige ein. Wenn diese 3 Dinge passen, kann es eigentlich keine Probleme beim Import geben.

Natürlich gibt es noch viele andere Wege die Daten bequem zu übernehmen.

### ***Literatur:***

**Hansen, J.; Nagel, J. (2014):** Waldwachstumskundliche Softwaresysteme auf Basis von TreeGrOSS - Anwendung und theoretische Grundlagen. Beiträge aus der NW-FVA, Band 11, 224 S.