

Tahofileformat 1.00

2.1.2019

Inhaltsverzeichnis

1	Vorwort	1
2	Grundaufbau	1
3	<options>	1
4	<params>	1
5	<mapPubSrc> und <mapallsrc>	2
5.1	<name>	2
5.2	<prefix>	2
5.3	<type>	2
5.4	<maxThreads>	2
5.5	<url>, <ext>, <offset>, <ksize> und <mapspf>	2
5.5.a	Pixelkarten (Basis und Overlay) nur <url>	2
5.5.b	Vektorkarten	3
5.5.c	(X)API	4
6	<OsmIds>	4
6.1	<ID>	4
6.1.a	<name>	4
6.1.b	<IDV>	4

1 Vorwort

Tahofiles dienen dazu die Einstellungen des Programms [Taho](#) zu speichern. Dazu gehören auch die Downloadquellen für OSM-Tiles u.ä.. Da ich diese auch für andere Programme benutze sind diese in einem Extrfile ausgelagert. Taho könnte aber auch alles aus einem File lesen.

2 Grundaufbau

Es ist ein File im XML-Format. Es enthält ein Element mit dem Namen "taho" und ein oder mehrere Unterelemente. Es folgen die Hauptelemente:

3 <options>

Hier wird alles abgelegt was man unter Bearbeiten/Optionen einstellen kann. Da es wenig sinnvoll ist dies per Hand zu ändern gehe ich hier nicht auf die einzelnen Elemente ein. Wer aber trotzdem etwas ändern will sollte kein Problem haben die Elemente den Optionen zuzuordnen. Einfach Einstellungen speichern und sich das File ansehen.

4 <params>

Hier sind die Einstellungen eines Projektes, also i.W. alles was man im Hauptfenster einstellen kann. Ansonsten gilt hier das Gleiche wie bei <options>

5 <mapPubSrc> und <mapallsrc>

Dieses Element enthält die Infos zu den Kartenquellen. Standardmäßig befinden sie sich in den Files defsrcP.taho und mydefsrc.taho. Dabei wird defsrcP mit dem Element <mapPubSrc> nur gelesen und sollte normalerweise auch nicht verändert werden. So kann dies über die Update Funktion einfach ersetzt werden ohne, daß persönliche Änderungen verloren gehen. Mydefsrc mit dem Element <mapallsrc> wird meist auch nur gelesen, aber wenn man z.B. eine ID eingibt wird das File neu geschrieben. Es enthält dann alle Quellen die aus einem <mapallsrc> gelesen wurden und natürlich die Ids. Für jede Quelle gibt es ein Element <src>. In diesem kommen die folgenden Elemente vor:

5.1 <name>

Der Name wird zur Auswahl benutzt und für die Ordnernamen. Also darf er auch nur Zeichen enthalten die für Ordnernamen erlaubt sind.

5.2 <prefix>

Die erzeugten Karten erhalten dieses Kürzel als Namensanfang. Da die Overlays nicht alleine geladen werden wird bei diesen keine Prefix benötigt.

5.3 <type>

0: Basispixelkarte; 1: Overlays; 2: Vektorkarten; 3: (X)Api Befehle; 100: Standard Basispixelkarte

5.4 <maxThreads>

Wieviel Threads dürfen maximal benutzt werden wenn von dieser Quelle geladen wird. Ist der Wert 0 oder nicht gesetzt können beliebig viele gestartet werden.

5.5 <url>, <ext>, <offset>, <ksize> und <mapspf>

5.5.a Pixelkarten (Basis und Overlay) nur <url>

Die Basiskarte hat <type>0</type> (eine kann mit Typ 100 zum Standard gemacht werden) und die Overlays <type>1</type>

Die URL kann auf zwei Arten angegeben werden. Alle mir bisher bekannten OSM - Server speichern z.B. Tile (x=1,y=2,zoom=3) unter

BASISURL/3/1/2.png

hier reicht es die BASISURL anzugeben, alternativ könnte man aber auch mit Platzhaltern arbeiten, also für obiges Bsp:

BASISURL/\$Z/\$X/\$Y.png

letzteres ist sinnvoll wenn die Syntax von der normalen abweichen sollte. Sollte in so einer URL ein \$ vorkommen müsste es verdoppelt werden. Will man auf diese Weise Karten anderer Quellen laden muss man die jeweiligen Nutzungsbedingungen beachten. Statt „\$X“ als Platzhalter kann auch „{X}“ oder „\${X}“ verwendet werden, Dabei wird Groß/Kleinschreibung ignoriert. Gleiches gilt natürlich für \$Y und \$Z.

Bei Quellservern mit persönlicher ID sieht die URL z.B. so aus:

<https://tile.thunderforest.com/cycle/{z}/{x}/{y}.png?apikey=meineID>

Um diese Quellen in defsrcP aufnehmen zu können habe ich für die ID einen Platzhalter benutzt, das sieht dann so aus:

https://tile.thunderforest.com/cycle/{Z}/{X}/{Y}.png?apikey={ID_Thunderforest}

Wichtig ist, daß es mit „{ID_“ anfängt und mit „}“ aufhört. Bei allen URLs mit gleicher ID sollte natürlich der gleiche Platzhalter verwendet werden.

Im entsprechenden Taho-File (ab Vers 2.10) gibt es zusätzlich für jede Quelle einen Prefix der für die

Filenamen benutzt wird. Die meisten Quellen bieten die Tiles als png. Files ab, aber es gibt Ausnahmen, deshalb kann man mit <ext> die Endung definieren, ist dies nicht vorhanden wird png verwendet.

5.5.b Vektorkarten

Diese haben <type>2</type>.

Bisher sind mir 2 Arten von URLs bekannt:

<http://openstreetmap.teddynetz.de/latest/img/63273/63273621.img.gz>

<http://osm.smash-net.org/srtm/53273621.img.gz>

beide bestehen aus einer Basisurl die in <url> abgelegt wird, also:

<url><http://openstreetmap.teddynetz.de/latest/img/></url>

<url><http://osm.smash-net.org/srtm/></url>

bei der ersten kommt dann noch ein Unterordner für jeweils 1000 Karten, bei der 2. sind alle im Hauptordner. Hat der Unterordner die Nummer Kartennummer/Anzahl_Karten_pro_Ordner muss man zusätzlich <mapspf> auf Anzahl_Karten_pro_Ordner setzen. Gibt es keine Unterordner kann dies entfallen bzw als Anzahl 0 benutzt werden, also für die beiden o.g. Quellen:

<mapspf>1000</mapspf>

<mapspf>0</mapspf>

Die Kartennummer berechnet sich nach:

$$\text{Number} = ((\text{int})((\text{lat} + 90) / \text{ksize}) + (\text{int})((\text{lon} + 180) / \text{ksize}) * (\text{int})(180 / \text{ksize})) + \text{offset}$$

Zusätzlich muss also noch ksize und offset definiert werden. Bei den beiden o.g. Quellen ist ksize=1 Grad:

<ksize>1.000000</ksize>

und der Offset 63240001 bzw 53240001

<offset>63240001</offset>

<offset>53240001</offset>

Damit ergibt sich insgesamt:

```
<src>
  <name>CompTeddy latest</name>
  <prefix>CTL</prefix>
  <url>http://openstreetmap.teddynetz.de/latest/img/</url>
  <type>2</type>
  <offset>63240001</offset>
  <ksize>1.000000</ksize>
  <mapspf>1000</mapspf>
</src>
<src>
  <name>SRTM</name>
  <prefix>SRTM</prefix>
  <url>http://osm.smash-net.org/srtm/</url>
  <type>2</type>
  <offset>53240001</offset>
  <ksize>1.000000</ksize>
  <mapspf>0</mapspf>
</src>
```

5.5.c (X)API

Diese werden als `<type>3</type>` gesetzt.

Im Programm sind Vektorkarten und API Zugriffe vermischt. Man kann also auch gleichzeitig IMG und OSM Daten laden. Natürlich muss auch hier eine URL definiert werden, z.B.:

`<url>http://xapi.openstreetmap.org/api/0.6/map?bbox=\$W,\$S,\$E,\$N</url>`

Die Platzhalter (`$W,$S,$E,$N`) werden durch die entsprechenden Grenzen des Bereichs ersetzt. Eigentlich hätte man damit alles, aber es gibt eine Größenbeschränkung bei diesem Befehl auf 100 Quadratgrad.

Dies kann man mit

`<ksize>100.000000</ksize>`

setzen. Lässt man dies weg oder setzt es zu 0 ist der Bereich beliebig groß. Man kann hier jede URL einsetzen die maximal die o.g. Parameter benötigt. Also vor allem natürlich alle API und XAPI Zugriffe.

S.: http://wiki.openstreetmap.org/wiki/API_v0.6 und <http://wiki.openstreetmap.org/wiki/DE:Xapi>

6 <OsmIds>

Hier legt Taho oder ein anderes Programm die Ids für Quellen ab bei denen man sich anmelden muss. Da diese alles andere als öffentlich sind gehören sie in `mydefsrc.taho` In der URL der Quelle wird ein Platzhalter mit dem ID-Namen eingetragen der vom Programm dann durch die ID ersetzt wird

6.1 <ID>

Für jede ID gibt es so ein Element mit zwei Einträgen:

6.1.a <name>

Dieser muß mit `ID_` beginnen ist sonst aber beliebig.

6.1.b <IDV>

Die persönliche ID

Dimitri Junker