Taho file format 2.Jan.2019

Contents

1 Preface

Taho files are used to save the settings of Taho. This includes the sources for OSM-tiles. Since I need them also for other programs this sources are saved in a file of them self. But all could be saved in one file.

2 Basics

Taho-files are in XML-Format. They include one Element by the name <taho> and one or more sub elements described below:

3 <options>

Here everything you can set under Edit/options is saved. There is no real need to change this part by hand, so I want explain the details here, but it is not difficult to understand if you look into it.

4 <params>

Here is everything for one project, so all you can set in the main Taho window.

5 <mapPubSrc> and <mapallsrc>

This element contains the information for the map sources. As said before this is usually in a separate file defsrcP.taho and mydefsrc.taho. The first one contains the sources published by me, the other one is for you to edit. This way defsrcP can be updated without loosing sources you found. To make it easier for the program to know which are private and which are public defsrcP contains <mapPubSrc> and mydefsrc contains <mapallsrc>, else they are the same.
For each source there is a <src> Block. In this the there are the following elements:

5.1 <name>

The name is used for the selection and for the folder names. So it might only contain characters allowed for folder-names

5.2 <prefix>

The map files will get this as a prefix in there names. Since the overlays are not saved individually there is no prefix needed for them.

5.3 <type>

0: Base pixel map; 1: Overlays; 2: Vector maps.; (X)Api Query; -2: obsolete

5.4 <maxThreads>

How many Threads may be used when downloading from this source. Is the value 0 or not set at all than it is unlimited.

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

5.5.a Pixel maps (Base and Overlay) only <url>

The base map has <type>0</type> and the Overlays <type>1</type>. The default base map can be set to <type>100</type>
The URL can be set in two ways. All OSM - servers I know so far save for example tile (x=1,y=2,zoom=3) under
BASEURL/3/1/2.png
in this case you just need to set the BASEURL, alternatively you could also use place-holders to define the same source as above as:
BASEURL/$Z/$X/$Y.png
but this makes only sense if the syntax differs from the usual OSM syntax. If such a URL contains a $ sign you have to double it. Instead of „$X“... you can also use „{x}“ or „${x}“ as placeholder. For Ids of sources where you need to register the Url might contain a private ID, like this:
https://tile.thunderforest.com/cycle/{z}/{x}/{y}.png?apikey=myID
To be able to include this sources in defsrcP the ID can be replaced by a placeholder:
https://tile.thunderforest.com/cycle/{Z}/{X}/{Y}.png?apikey={ID_Thunderforest}
The placeholder has to start with „{ID_“ and end with „}“. In all URLs with the same ID the same Placeholder should be used.
Especially if you want to download maps of other sources than OSM, you have to respect the terms of use. Besides the URL and the maximum zoom-level a prefix is saved for every source which will be used for the file-names. Most sources use tiles in the png format, but not all, therefore you can define the file format in <ext>, if this is not defined png is used by default.

5.5.b Vector maps

These have <type>2</type>. Until now I know two kinds of URLs:
http://openstreetmap.teddynetz.de/latest/img/63273/63273621.img.gz
http://osm.smash-net.org/srtm/53273621.img.gz
both start with a base url saved in <url> so:
<url>http://openstreetmap.teddynetz.de/latest/img</url>
<url>http://osm.smash-net.org/srtm</url>
At the first this is followed by a subfolder, each for 1000 maps, at the second all maps are in one folder. If the subfolder has the number (map_number)/Number_in_folder you need to set <mapspf> to this Number_in_folder. If there are no sub-folders you can either ignore this tag or set it to 0, so for the two examples:
<mapspf>1000</mapspf>
<mapspf>0</mapspf>
The map number is calculated by:

```
Number = ((int)((lat + 90) / ksize) + (int)((lon + 180) / ksize) * (int)(180 /
ksize)) + offset
```

In addition to the tags above ksize and offset need to be defined. For the examples ksize=1 deg:
<ksize>1.000000</ksize>
and offset 63240001 or 53240001
<offset>63240001</offset>
<offset>53240001</offset>
In all:

```
<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

These have <type>3</type>.
In the program there is no separation between vector maps and (X)API downloads. The URL usually contains place-holders, for example:
<url>http://xapi.openstreetmap.org/api/0.6/map?bbox=$W,$S,$E,$N</url>
The ($W,$S,$E,$N) will be replaced by the area. This would be all if there there would not be a size limit for some API query. For the map query above this limit is 100 square degrees, so you need to set
<ksize>100.000000</ksize>
If there is no limit set it to 0 ore not at all. For more information see:
S.:http://wiki.openstreetmap.org/wiki/API_v0.6 and http://wiki.openstreetmap.org/wiki/Xapi

6 <OsmIds>

Here Taho are some other program saves the Ids necessary for some sources where you need to register to use them. Since this is definitely not public it is saved in mdefsrc.taho. In the URL of the source, which might be in defsrcP a placeholder is used instead of the ID. The program then looks up here for the ID to replace the placeholder with. So for each ID the is one Element:

6.1 <ID>

This element contains tho following two elements.

6.1.a <name>

This placeholder has to start with „ID_“ but else you are free to name it in any way.

6.1.b <IDV>

The private ID

Dimitri Junker