

Format des fichiers Tahoe 2.00 28.8.19

Table des matières

1	Préface	1
2	structure de base	1
3	<options>	1
4	<params>	1
5	<mapPubSrc> et <mapallsrc>	2
5.1	<name>	2
5.2	<prefix>	2
5.3	<type>	2
5.4	<maxThreads>	2
5.5	<url>, <ext>, <offset>, <ksize> et <mapspf>	2
5.5.a) carte de base matricielle (base et overlay) seulement <url>	2
5.5.b) Cartes vectorielles	3
5.5.c) (X)API	3
6	<OsmIds>	4
6.1	<ID>	4
6.1.a) <nom>	4
6.1.b) <IDV>	4
7	<OsmLogins>	4
7.1	<LOGIN>	4
7.1.a) <Domain>	4
7.1.b) <name>	4
7.1.c) <pwd>	4

1 Préface

Les fichiers Tahoe sont utilisés pour enregistrer les paramètres du programme Tahoe. Cela inclut également les sources de téléchargement pour les tuiles OSM, etc. Comme je l'utilise également pour d'autres programmes, celles-ci sont externalisées dans un fichier supplémentaire. Tahoe pouvait aussi tout lire d'un fichier.

2 structure de base

C'est un fichier au format XML. Il contient un élément nommé "taho" et un ou plusieurs sous-éléments. Voici les éléments principaux:

3 <options>

Ici tout a été sauvegardé ce que vous pouvez ajuster sous Edition / Options. Cela a entre-temps erré dans le registre, donc cela ne devrait être trouvé que dans les anciens fichiers taho.

4 <params>

Voici les paramètres d'un projet, c'est-à-dire tout ce que vous pouvez rentrer dans la fenêtre principale. Sinon, la même chose s'applique ici que avec <options>

5 <mapPubSrc> et <mapallsrc>

Cet élément contient les informations sur les sources des carte. Par défaut, ils se trouvent dans les fichiers defsrcP.taho et mydefsrc.taho. DefsrcP est uniquement lu avec l'élément <mapPubSrc> et ne devrait normalement pas être modifié. Ceci peut donc être facilement remplacé via la fonction de mise à jour sans perte de modifications personnelles. Mydefsrc avec l'élément <mapallsrc> est principalement lu, mais si vous La saisie d'un identifiant réécrira le fichier. Il contient ensuite toutes les sources lues depuis un <mapallsrc> et bien sûr les identifiants. Pour chaque source, il existe un élément <src>. Dans ceci les éléments suivants se produisent:

5.1 <name>

Ce nom est besoin pour la sélection et pour les nom de dossiers. Alors il doit seulement contenir des lettres autoriser pour les dossiers.

5.2 <prefix>

Les cartes créer recevrons des noms commencent avec ce préfixe.

5.3 <type>

0: carte de base matricielle; 1: Overlays; 2: cartes vectorielles; 3 (X)Api ; 100: carte de base matricielle de défaut

5.4 <maxThreads>

Le nombre maximale de Threads si on charges des cartes de cette source. Si ce nombre est 0 ou n'est pas définit il n'y a pas de limite.

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

5.5.a carte de base matricielle (base et overlay) seulement <url>

La carte de base a <type>0</type> et les Overlays <type>1</type>.

La URL peut être définit en deux version. Tous les OSM-server que je connais sauvent par exemple la tuiles (x=1,y=2,zoom=3) sous

URL_DE_BASE/3/1/2.png

pour ceux là il suffit de définir la URL_DE_BASE, alternativement vous pouvez utiliser des Métacaractère, alors pour l'exemple dernier:

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

ceci est seulement raisonnable si la syntaxe est différent du standard OSM. Si le URL contienne un \$ il faut le doubler. Au lieu de "\$X" en tant que Métacaractère, vous pouvez également utiliser "{X}" ou "\${X}", où la casse est ignorée. La même chose vaut pour \$Y et \$Z, bien sûr.

Pour les serveurs source dotés d'un identifiant personnel, l'URL ressemble par exemple à comme ça:

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

Pour pouvoir enregistrer ces sources dans defsrcP, j'ai utilisé un Métacaractère pour l'ID, alors le URL ressemble à ceci:

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

L'important est que cela commence par "{ID_" et se termine par "}". Bien entendu, le même Métacaractère doit être utilisé pour toutes les URL portant le même identifiant.

Si vous voulez utiliser ce programme pour charger des cartes autre que de OSM vous devrez respecter les conditions d'utilisation. La plus part des source donne les tailles en format png, si non on peut définir le format dans <ext>

5.5.b Cartes vectorielles

Ceci ont <type>2</type>.

Pour le moment je connais deux sortes de URL:

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

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

les deux commence avec une URL de base qui est sauver dans <url> :

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

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

dans le premier exemple il y a encore un sous-dossier pour 1000 cartes, dans le deuxième tous sont dans le dossier principale. Si le sous dossier a le Numéro_de_cartes/nombre_de_cartes_par_dossier il faut sauver ce nombre_de_cartes_par_dossier dans <mapspf>. Si il y a pas de sous dossier il faut mettre 0.

Alors pour les deux source d'avant:

<mapspf>1000</mapspf>

<mapspf>0</mapspf>

Le numéro de cartes se calcule par:

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

Alors il faut encore sauver ksize et offset. Pour les exemple d'avant ksize=1 degré:

<ksize>1.000000</ksize>

et les offset 63240001 ou bien 53240001

<offset>63240001</offset>

<offset>53240001</offset>

Alors en tout:

```
<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

Pour celui ci il faut mettre <type>3</type>.

Dans le programme les carte vectorielles et les accès API ne sont pas séparées. Bien sûr il faut définir la URL, par exemple

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

Les garde-place (\$W,\$S,\$E,\$N) serrant remplacer par les limites de la zone a charger. Ci il y a une limite de la grandeur de la zone de 100 degrés carres alors il faut mettre encore:

<ksize>100.000000</ksize>

Si il n'y a pas de limite mètres ksize=0. Bien sûr on peut mettre tous les URL ici qui contiennent aux plus les paramètres (\$W,\$S,\$E,\$N). Principalement les query de (X)API. voir.:http://wiki.openstreetmap.org/wiki/API_v0.6 et <http://wiki.openstreetmap.org/wiki/FR:Xapi>

6 <OsmIds>

Ici, Tahoe ou un autre programme dépose les identifiants pour les sources où vous devez vous inscrire. Comme ils sont pas publics, ils appartiennent à mydefsrc.taho. Dans l'URL de la source, un Métacaractère avec le nom de l'ID est entré, qui est ensuite remplacé par l'ID du programme.

6.1 <ID>

Il y a un élément avec deux entrées pour chaque ID:

6.1.a <nom>

Cela doit commencer par ID_.

6.1.b <IDV>

L'identifiant personnel

7 <OsmLogins>

Ici, Tahoe ou un autre programme dépose les informations de login au serveur de tuiles dont il a besoin. Comme ils sont tout sauf publics, ils sont sauver dans mydefsrc.taho

7.1 <LOGIN>

Pour chaque serveur nécessitant une login, il existe un élément avec trois entrées:

7.1.a <Domain>

Ceci est automatiquement formé à partir de l'URL. La partie est utilisée entre la deuxième et la troisième, / ' alors sans le "http: //" et, plus important encore, sans le numero de tuiles, sinon il faudrait entrer à nouveau pour chaque tuile les données. Ainsi, à partir de http://privat.karten.de/5/4/3.png, privat.karten.de serait utilisé. Comme cela se produit automatiquement, vous ne devriez rien changer. S'il y a des problèmes, par exemple un serveur proposant plusieurs cartes, par exemple Ces URL ont: http://privat.karten.de/KarteA/5/4/3.png et http://privat.karten.de/KarteB/5/4/3.png et il demanderait des différentes login pour chacun, veuillez alors signaler à moi, ou changer le code source de Tahoe ou similaire vous-même.

7.1.b <name>

Le nom de Login

7.1.c <pwd>

Le mot de passe de login

Dimitri Junker