

Taho.exe

23.12.13

Inhaltsverzeichnis

1	Vorwort	2
1.1	Vorwort zur Version 3	2
1.2	Vorwort zur Version 3.08	2
2	Installation	2
3	Benutzung	3
3.1	Sprache	3
3.2	Koordinaten	3
3.3	Ausgabe	3
3.3.a)	Namensgebung	3
3.3.b)	Ausgabe Verzeichnis	4
3.4	Pixel/Vektorkarten	4
3.5	Kartenquelle	4
3.5.a)	Basispixelkarte	4
3.5.b)	Overlays für Pixelkarte	4
3.5.c)	Vektorkarten	4
3.6	Weitere Einstellungen für Pixelkarten	4
3.6.a)	Größe	4
3.6.b)	Bits/Pixel	4
3.6.c)	Zoom-Stufen	5
3.6.d)	File-Typ	5
3.6.e)	Kalibrationsfiles	5
3.6.f)	ohne Kalibrationfiles in Unterordnern	5
3.7	Karten erstellen (Button oder Menü Bearbeiten)	5
3.8	OSM-Bugs laden (Menü Bearbeiten)	5
3.9	KMZ erstellen (Menü Bearbeiten)	5
3.10	Optionen (Menü Bearbeiten)	6
3.10.a)	User Agent ID	6
3.10.b)	Anzahl Threads	6
3.10.c)	Kartenquelle	6
3.10.d)	Cache Gültigkeit	7
3.10.e)	Packer	7
3.11	Einstellungen (Menü Datei)	8
4	Fileformat Taho-Source	8
4.1	<name>	8
4.2	<prefix>	8
4.3	<type>	8
4.4	<maxThreads>	8
4.5	<url>, <offset>, <ksize> und <mapspf>	8
4.5.a)	Pixelkarten (Basis und Overlay) nur <url>	8
4.5.b)	Vektorkarten	9
4.5.c)	(X)API	9
5	Erfahrungsberichte	10
6	Rechtliches und Quelle	10
7	Änderungen seit Version 2.06	10
8	Bekannte Probleme und Aussicht	12

1 Vorwort

Taho dient dazu, aus den 256*256 Pixel großen Kacheln der OSM-Karten größere Karten zusammenzustellen, so dass diese von GPS-Programmen wie Glopus benutzt werden können. Je nach Programm sind unterschiedliche Größen sinnvoll und damit das Programm weiß, welchen Bereich die Karte darstellt, werden sog. Kalibrationsfiles für jedes Grafikfile benötigt. Auch diese sind je nach GPS-Software unterschiedlich. Es kann aber auch Vektorkarten laden, hier dient es praktisch nur als Downloader.

Taho.exe war ursprünglich ein grafisches Frontend für taho.pl. Es diente als Verbindung von diesem mit <http://www.openstreetmap.org/export/>. Die neue Version kommt aber ohne taho.pl aus, ist also deutlich einfacher zu installieren und flexibler. Mit etwas Glück wird taho.pl bald in die o.g. Export-Seite integriert, so dass dieses Programm unnötig wird.

Ab Version 2.02 können wie bei taho.pl auch 8Bit PNGs erzeugt werden. Damit sollte es auch wieder mit OZI funktionieren.

Durch den Umstieg auf Android benutze ich das Programm inzwischen selber sehr selten, bin also umso mehr auf [Erfahrungsberichte](#) angewiesen.

1.1 Vorwort zur Version 3

Mit Vektorkarten habe ich noch sehr wenig eigene Erfahrungen und keine Erfahrungsberichte anderer. Dieser Teil sollte also als β angesehen werden. Bitte schickt mir also Verbesserungsvorschläge.

1.2 Vorwort zur Version 3.08

Dieses Programm entstand ursprünglich mit Visual C++ 6 unter Windows XP. Nach einem Wechsel zu Windows 8.1 stellte ich fest, dass weder dieses Programm noch die Programmierumgebung funktionierten. Also besorgte ich mir erst einmal die Testversion von Visual Studio 2013 und passte das Programm an. Dabei ist es möglich, dass ich Fehler übersehen habe oder neue hinzugekommen sind. Da ich derzeit alle meine Programme entsprechend anpassen muss kann ich sie nicht so ausgiebig testen um wirklich jeden Fehler zu finden. Falls Sie also Fehler finden nicht das Programm löschen sondern mir mitteilen.

2 Installation

Eine wirkliche Installation ist nicht nötig, einfach das ZIP-File irgendwo hin entpacken. Damit die Hilfe funktioniert, muss dieses File im Docu-Ordner erhalten bleiben. Will man KMZ Files oder Vektorkarten (img) erzeugen, wird ein Packer benötigt, der Zip-Files erstellen kann und als Parameter ein Listfile akzeptiert, in dem die zu packenden Files stehen. Getestet habe ich es mit der Kommandozeilenversionen von 7-Zip: [7za.exe](#) und [Winrar](#). Die nötigen Einstellungen findet man im Menü: Bearbeiten/Optionen. Installiert man Taho auf einem USB-Stick ist es sinnvoll das Default-File nicht im Ordner „Eigene Dateien“ sondern im Programmordner zu speichern. Außerdem sollten dann Pfade im *.taho-File relativ zu diesem gespeichert werden, da sonst bei einem neuen Laufwerksbuchstaben die Pfade nicht mehr stimmen. Im rechten Teil der entsprechenden Optionen kann man die Einstellungen einzeln verändern oder man benutzt einen der beiden Buttons links.

3 Benutzung

3.1 Sprache

Um die Sprache zu ändern, muss man ins Menü: Bearbeiten/Optionen gehen. Derzeit existieren Programm und Hilfetexte in Deutsch, Französisch und Englisch. Die verschiedensprachigen Texte stehen im File taho_lang.html. Will man einen Fehler korrigieren oder z.B. eine neue Sprache hinzufügen, kann man dieses File z.B. mit OpenOfficeWriter editieren. Zum Korrigieren einfach den Text verändern. Will man eine neue Sprache hinzufügen, muss man 5 Sachen machen:

1. eine neue Spalte hinzufügen
2. im obersten Feld dieser Spalte den Namen der Sprache in einer beliebigen Sprache einfügen.
3. im zweiten Feld die ID dieser Sprache eintragen, s. <[http://msdn.microsoft.com/en-us/library/dd318693\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd318693(v=VS.85).aspx)> Man kann entweder den Wert aus der Spalte „Prim.lang. Identifier“ oder den aus „Locale identifier“ benutzen. Letzteres ist nur sinnvoll, wenn man mehrere Länderversionen einer Sprache einfügt. Wichtig ist die ID nur zur automatischen Sprachauswahl, und 2 Spalten dürfen nicht die gleiche ID haben.
4. für die Sprachauswahl braucht man noch den Sprachnamen in allen bereits vorhandenen Sprachen. Angenommen Sie wollten als 4. Sprache Spanisch hinzufügen, so müsste die folgende Zeile eingefügt werden:

IDD_OPT(IDC_LANG):4	L	4	0x8f0456		Spanisch	Spanish	espagnol
---------------------	---	---	----------	--	----------	---------	----------

Das erste Feld ist dabei unwichtig. Die ID im 4. Feld könnte sich in Zukunft ändern, falls es also nicht klappt, mal die anderen Zeilen mit den Sprachen suchen und vergleichen.

5. in den restlichen Feldern der neuen Spalte die Texte übersetzen. Dabei ist die ID STRING_TABLE(IDS_HLPFILE) etwas Spezielles. Hier geht es nicht um die Übersetzung eines Textes, sondern um den Filenamen des Helpfiles. Hat man also neben der Übersetzung des Programms auch das Helpfile ins Spanische übersetzt und es z.B. ayuda.pdf genannt, so müsste dies hier eingetragen werden. Hat man es nicht übersetzt und will stattdessen die deutsche Hilfe angezeigt bekommen, trägt man deren Namen also liesmich.pdf ein.

3.2 Koordinaten

Startet man taho.exe, öffnet sich neben dem eigentlichen Programmfenster auch noch der Standardbrowser mit der Seite <<http://www.oche.de/~junker/OSM/bbox-tool/bbox.htm>>. Dort kann man den gewünschten Bereich auswählen. Um die so ermittelten Koordinaten in das andere Fenster zu übertragen, kann man sie einzeln abtippen, oder man kopiert den erzeugten Link (<bbox...>) unten links per <STRG><C> in die Zwischenablage und fügt ihn per <Strg> <V> in das letzte Eingabefeld im Bereich Kartenbereich ein. Klickt man dann auf „aus URL holen“, so wird versucht die Koordinaten zu ermitteln.

3.3 Ausgabe

3.3.a Namensgebung

Außer der Namensgebung mit Kartennummern (wie bei taho.pl) kann man jetzt auch Namen mit Koordinaten erzeugen. Ein weiterer Unterschied zu taho.pl ist, dass als Zoomlevel der benutzt wird, der den verwendeten OSM-Karten entspricht, unabhängig von der Kartengröße. Falls da unerwartet ein Programm Probleme mit hat, bitte melden. Außerdem gibt es die Einstellung „Ordner“. Hier wird ein Ordner pro Zoomlevel, darin je ein Ordner pro x-Koordinate erzeugt und die Files erhalten die y-Koordinate. Gedacht ist dieser Modus als Alternative zum Größenmodus „keine“ wenn man zusätzlich Overlays nutzen will. Auch wenn man diesen Modus mit allen Größen kombinieren kann werden wohl die GPS-Programme nur die Größe 256*256 akzeptieren.

3.3.b Ausgabe Verzeichnis

Ist „Auto“ ausgewählt, wird automatisch ein Ausgabe Verzeichnis ermittelt. Dabei wird für jeden Renderer (s.[3.5a](#)) ein eigenes Verzeichnis verwendet. Ansonsten kann man hier ein beliebiges Ziel vorgeben.

3.4 Pixel/Vektorkarten

Durch Auswahl eines der beiden Reiters wählt man aus welche Art Karten man laden will. Die meisten hiernach folgenden Einstellungen werden nur für Pixelkarten benötigt und sonst ausgeblendet. Die Erzeugung von Vektorkarten ist noch im β -Stadium.

3.5 Kartenquelle

3.5.a Basispixelkarte

Hier kann man aus verschiedene Kartenversionen auswählen.

Hier kann man auswählen welchen Kartenrenderer man will, das sind einfach unterschiedliche Kartenversionen, die Karten sehen also leicht unterschiedlich aus, sind aber sonst gleichwertig.

3.5.b Overlays für Pixelkarte

Über die Basiskarten können noch sog. Overlays gelegt werden, diese können z.B. Bojen (Seamark) oder Höheninfos (Topo, Land Shading,...),... enthalten. Um ein oder mehrere Overlays zu aktivieren/deaktivieren dessen Check-Box anklicken. Um die Liste der Quellen & Overlays zu bearbeiten s. [3.10a](#)

3.5.c Vektorkarten

Hier kann man die Download-quelle auswählen. Im Gegensatz zu den Pixelkarten ist hier keine Trennung in Quelle und Overlays nötig. Um ein oder mehrere Quellen zu wählen dessen Check-Box anklicken. Hinter dem Namen steht jeweils ob es eine *.img oder *.osm Quelle ist. *.img sind fertige Files die nur geladen werden. Bei den osm Downloads handelt es sich um direkte Zugriffe auf die OSM-Datenbank. Man erhält also die aktuellsten Daten, aber auch alle, entsprechend lange kann der Download dauern. Um die Liste der Quellen zu bearbeiten s. [3.10a](#)

3.6 Weitere Einstellungen für Pixelkarten

3.6.a Größe

Hier kann man die Größe der einzelnen Karten auswählen. Dabei gibt es drei Besonderheiten:

1. frei: hier wird der Bereich als eine Grafik erzeugt. Bis zu welcher Größe das wirklich klappt, dürfte vom Rechner abhängen. Dieser Modus ist für GPS-Programme nicht geeignet, aber wer einfach so eine Karte haben will, kann dies nutzen. Außerdem ist dieser Modus sinnvoll, um eine Karte für UI-View zu erstellen.
2. keine: hier werden keine Karten zusammgebaut. Es werden also nur die Kacheln geladen. Es gibt Software, die damit arbeiten kann. Kalibrationsfiles sind dabei nicht nötig.
3. 256*256 eigentlich keine Besonderheit, aber es erscheint unnötig, da ja auch die Tiles bei „keine“ (s. Voriger Punkt) Tiles dieser Größe erzeugt. Aber einerseits sind diese auf viele Unterordner verteilt, andererseits können dort keine Tiles mit Overlays kombiniert werden.

3.6.b Bits/Pixel

Bei Bit/Pixel kann man zwischen 8, 24 oder 32 Bit wählen. Ozi unterstützt nur 8 und 24 Bit. Da die Karten intern zuerst in 32Bit erzeugt werden, ist eine Ausgabe in diesem Format die schnellste Variante, vor allem auf langsamen Rechnern. Bei der Umrechnung 32->24 Bit ist der Rechenaufwand ebenfalls gering, dafür spart man bei 8Bit Speicherplatz.

3.6.c Zoom-Stufen

Taho.pl benutzte 2 unterschiedliche Zoom-Stufen. Taho.exe nur den, den man von den OSM-Karten kennt. Es können mehrere Zoom-Stufe ausgewählt werden.

3.6.d File-Typ

Hier kann man zwischen png und jpg wählen. Einige Garmin Geräte kommen wohl nur mit jpg zurecht, sonst scheint png OK zu sein. Da es aber auch Programme gibt die die Endung png.tile wünschen, wie den Osmroid und OSMTracker unter Android gibt es auch noch diese Einstellung. Bis auf die Endung sind die Files aber normale png.

3.6.e Kalibrationsfiles

Neben den eigentlichen Karten braucht man meist noch Kalibrationsfiles. Leider gibt es kein Standardformat. Hier kann man das gewünschte Format auswählen. Worldfiles gibt es wohl in mehreren Versionen, das hier erzeugte ist für "WGS 84 /World Mercator" EPSG 3395.

3.6.f ohne Kalibrationfiles in Unterordnern

Das meiste steht schon in [3.3a](#) und [3.6a](#), aber hier nochmal zusammengefaßt. Benutzt man ein Programm, welches die Files so erwartet wie sie auf den OSM-Servern liegen ist das einfachste als Größe „keine“ einzustellen, denn die Tiles werden ja eh so gecacht. Unterstützt das Programm aber keine Overlays, will man diese aber nutzen kann man alternativ folgendes einstellen

- Größe: 256*256
- Namensgebung: Ordner.
- Filetyp: png oder png.tile

3.7 Karten erstellen (Button oder Menü Bearbeiten)

hier werden endlich die Karten und ggf. Kalibrationsfiles erstellt. Um dies zu beschleunigen werden 4 Tasks parallel gestartet. Es wird zwar ein Fortschrittsbalken angezeigt, allerdings ist dabei der Download eines Files ein Schritt -> vor allem bei den meist sehr großen OSM-Files (z.B. Stadt Aachen etwa 30MB) scheint es so, als ob nichts passiert. Da vorher nicht bekannt ist wie groß das File ist sehe ich da auch keine Verbesserungsmöglichkeit.

3.8 OSM-Bugs laden (Menü Bearbeiten)

Hier wird für den ausgewählten Bereich die OSMBugs geladen. Dies kann man zwar auch über <http://openstreetbugs.appspot.com/>, allerdings existiert dort ein mir nicht erkennbarer Filter, der je nach Zoomlevel mehr oder weniger Bugs durchlässt. Ein weiteres Problem ist, dass die Texte oft sehr lang sind. Deshalb kann man hier wählen, dass die Texte auf eine Nummerierung gekürzt werden. Der eigentliche Text steht dann in einer Text-Datei. So bleiben die Karten lesbar. Auch können nicht alle Programme gpx-Files als POI-File nutzen. Deshalb kann man hier zwischen gpx und asc (z.B. für Glopus) wählen. Falls für ein anderes Programm noch ein anderes Format benötigt wird, sagt es mir.

3.9 KMZ erstellen (Menü Bearbeiten)

Die hier erstellten kmz-Files sind zip-Files aus einem doc.kml und einer oder mehreren Karten. Da man ggf. mehrere Download Durchläufe braucht um die nötigen Karten zu laden, habe ich das Erzeugen des kmz vom Karten erzeugen separiert.

Ruft man diesen Menüpunkt auf ohne vorher die nötigen Einstellungen zum Aufruf des Packers gemacht zu haben, wird man dazu aufgefordert, dies nachzuholen.

Um kmz-Files zu erzeugen braucht man also Karten mit kml-Files als Kalibrationfile. Dann wählt man im KMZ-Dialog die kml Files aus. Die Karten werden automatisch daraus ermittelt. Je nach Anwendung des kmz schienen mir 3 Optionen sinnvoll:

1. entweder erzeugt man für jede Karte ein eigenes kmz (also 1kml + 1Karte → 1kmz)
2. oder ein kmz pro Zoom-Stufe

3. oder ein kmz für alles.

Im Falle 1 erhält das kmz den gleichen Namen wie das kml. Für die anderen Fälle muss man einen Filenamen vorgeben. Im Falle2 wird noch automatisch die Zoom-stufe zum Namen angehängt. Hier funktioniert also nicht die Warnung falls die Files bereits existieren.

3.10 Optionen (Menü Bearbeiten)

hier kann man:

- die Sprache auswählen. [s.3.1](#))
- den User Agent ID festlegen (s.u.)
- die maximale Anzahl Threads (s.u.)
- einstellen wo Dateien gesucht werden und wie Pfade abgespeichert werden (s. [Installation](#))
- Einstellungen zur Kartenquelle vornehmen
- den Packer und seine Syntax für die Kommandozeile bestimmen.

3.10.a User Agent ID

Eine weitere Anpassung um den Lizenz-Bedingungen der <http://www.wanderreitkarte.de/> näher zu kommen. Diese fordert u.a. eine eigene User Agent Identifikation. Da der Betreiber derzeit nicht auf Anfragen reagiert (Urlaub?) ist noch nicht klar ob dies etwas bringt oder nicht. Standardmäßig meldet sich Tahoe als Browser, das ist so in Visual C++. Jetzt kann man aber auch einstellen, daß es sich als Tahoe mit oder ohne Versionsnummer meldet, also „Taho“ oder aktuell „Taho 3.06“ Falls für die Reit&Wanderkarte eine Freischaltung für jede ID nötig ist wäre wahrscheinlich „Taho“ das beste. Im Zweifelsfall also eine Mail an info@wanderreitkarte.de schreiben. Falls Du seine Tiles nicht nutzt lasse es bei Standard.

3.10.b Anzahl Threads

Die optimale Anzahl Threads hängt von vielen Dingen ab, u.a. von:

- der Internetverbindung, je schneller die ist je eher bringen mehr Threads etwas
- Anzahl Prozessorkerne: pro Kern ist mindestens ein Thread sinnvoll, aber da jeder Thread abwechselnd lädt (hoher Internet-Last, geringe CPU-Last) und rendert (keine Internet-Last aber hohe CPU-Last, vor allem wenn die Anzahl Farben reduziert wird) kann es auch sinnvoll sein mehr Threads als Kerne laufen zu haben.
- Kartenquelle: verschiedene Server reagieren unterschiedlich auf parallele Downloads. Extrembeispiel: die Reit&Wanderkarte blockt dann ganz. Deshalb kann man bei jeder Kartenquelle zusätzlich eine maximale Anzahl Threads einstellen, bei der Reit&Wanderkarte ist dies 1. [s. 4.](#))

3.10.c Kartenquelle

Da sich die URLs ab und zu mal ändern oder jemand neue findet, kann man sie nachladen. Die mir derzeit bekannten Karten sind fest in Tahoe enthalten. Beim Programmstart wird zusätzlich versucht, `defsrc.taho` und `mydefsrc.taho` nachzuladen. Hat eine der so geladenen Quellen den gleichen Namen wie eine bereits vorhandene, wird diese überschrieben. Will man also eine URL ändern, muss man den Namen genau übernehmen. Am einfachsten geht dies, indem man über den Export-Button zuerst den aktuellen Stand in ein File schreibt und dieses dann mit einem Editor verändert. Zur Syntax des Tahoe-Source Files [s. 4.](#))

Im Programm existiert eine schwarze Liste von alten Quellen die es nicht mehr gibt. Wird so eine Quelle in einem der taho-Files gefunden wird sie ignoriert. Derzeit stehen auf der Liste: „Osmarender“, „Cycle1“ und „Cycle2“.

Man kann solche auf dem Rechner vorhandenes taho-Files auch über den Menüpunkt „Datei/Einstellungen laden“ nachladen. Dies ist sinnvoll wenn man diese Quellen/Overlays nur gelegentlich nutzen will. Damit nicht jeder selber rausbekommen muss, wie die neue URL ist, gibt es im Optionen-Dialog „update Quell-URLs“. Hier wird versucht ein `defsrc.taho` aus dem www zu laden. Ich

lege eine aktuelle Version unter:

<http://www.dimitri-junker.de/defsrc.taho>

ab. Da ich aber öfter länger verreise und dann nicht in der Lage bin dies zu ändern, habe ich ein indirektes Verfahren für den Download eingebaut. Taho sucht auf der Seite:

http://wiki.openstreetmap.org/wiki/Taho#Tiles_sources

nach einem Link mit defsrc.taho. Wer also sein geändertes defsrc.taho allen zugänglich machen will, kopiere es irgendwohin und setze den Link auf o.g. Seite entsprechend.

Damit man einfach defsrc.taho updaten kann gibt es jetzt das mydefsrc.taho, so kann man eigene Quellen dort sichern ohne das sie bei einem Update verloren gehen

Auf [dieser Seite](#) gibt es etwa 275 Sprachversionen als Overlays, die mit der Basiskarte „No Label“ benutzt werden sollen. Um die Overlay-Auswahl nicht ganz unübersichtlich zu machen habe ich nur Deutsch, Englisch und Französisch aufgenommen. Man kann aber einfach weitere Sprachen hinzufügen. Dazu sucht man sich zuerst auf o.g. Seite das gewünschte Overlay aus, z.B. die spanische Version „osm-label-es“ dann editiert man das mydefsrc.taho. Dort gibt es z.B. schon:

```
<src>
    <name>osm-labels-de</name>
    <url>http://a.www.toolserver.org/tiles/osm-labels-de</url>
</src>
```

Fügt man entsprechend einen Block:

```
<src>
    <name>osm-labels-se</name>
    <url>http://a.www.toolserver.org/tiles/osm-labels-es</url>
</src>
```

hinzu kann man auch spanische Karten benutzen.

3.10.d Cache Gültigkeit

Die OSM-Kacheln (256*256 Pixelkarten) im tah-Unterverzeichnis werden gecacht, d.h. wenn dort das File bereits vorhanden und weniger als x Tage alt ist, wird es nicht neu geladen. Dies habe ich so von taho.pl übernommen. Dort waren 7 Tage fest vorgegeben, hier kann man dies einstellen.

3.10.e Packer

Wie unter [3.9](#)) beschrieben sind die kmz-Files eigentlich zip-Files. Das Packen erledigt taho nicht selber, sondern ruft einen bei den meisten Usern eh vorhandenen Packer auf. Diesen muss man hier auswählen. Außerdem sind die Vektorkarten als gz-Files vorhanden und müssen ausgepackt werden, auch dafür braucht man einen Packer. Neben dem Pfad wird auch noch die Form der Kommandozeilen benötigt, denn leider ähneln sich diese bei den verschiedenen Programmen zwar, sind aber nicht identisch. Da die Länge einer Kommandozeile beschränkt ist, aber evtl. sehr viele Files in ein kmz gepackt werden sollen, verwendet taho den List-File Modus. Es erzeugt also eine Liste aller zu packenden Files in einem List-File. An den Packer müssen also der Name des kmz-Files und des Listfiles übergeben werden, außerdem Befehlskürzel und Parameter. Für 3 Packer habe ich dies ermittelt, und wenn man einen dieser selektiert, wird die Kommandozeile automatisch angepasst, kann aber immer noch geändert werden. Für die Filenamen und Pfade müssen natürlich Platzhalter benutzt werden (\$Q,\$Z,\$L s.u.).

Zum Packen der kmz-Files sind die folgenden Kommandozeilen vorgegeben:

- 7-zip (bzw. seine Kommandozeilen Version 7za.exe): „a -tzip \$Z @\$L“
- Winrar: „a -afzip \$Z @\$L“
- Winzip: „-min -a \$Z @\$L“

Die benutzen Platzhalter sind: \$Z für das kmz-File und \$L für das Listfile.

In allen Fällen bedeutet das „a“ oder „-a“ add, also hinzufügen zu einem gepackten File. „-tzip“ bzw. „-afzip“ bedeutet, dass als Packverfahren zip benutzt werden soll, „-min“, dass das Fenster minimiert werden soll. Das @ vor dem \$L sagt dem Packer, dass danach nicht das zu packende File sondern eben ein Listfile folgt.

Zum Entpacken der gz-Files sind die folgenden Kommandozeilen vorgegeben:

- 7-zip (bzw. seine Kommandozeilen Version 7za.exe): „x \$Q -o\$Z“

- Winrar: „x \$Q \$Z“
- Winzip: „-min -e \$Q \$Z“

Die benutzen Platzhalter sind: \$Q für das Quellfile und \$Z für das Zielverzeichnis

In allen Fällen bedeutet das „x“ oder „-e“ extract, also herausholen aus einem gepackten File.

Im unwahrscheinlichen Fall, dass in der Kommandozeile ein \$ vorkommt, sollte man dies verdoppeln, vor allem wenn danach ein Z oder L folgt.

Die Filenamen setzt taho immer in Anführungszeichen, damit sollte es auch mit Leerstellen im Filenamen funktionieren. Winzip habe ich nicht getestet. Wer einen anderen Packer zur Kooperation bringt, teile mir doch bitte die nötige Einstellung mit, dann kann ich dies in Taho eintragen.

3.11 Einstellungen (Menü Datei)

Hier kann man die gemachten Einstellungen und Optionen speichern und wieder laden. Speichert man sie unter dem vom Programm angebotenen default-Namen, so werden diese Einstellungen automatisch beim Programmstart verwendet. Diese *.taho Files können auch als Parameter an das Programm verwendet werden. Stellt man ein, dass *.taho immer mit diesem Programm geöffnet werden sollen reicht also ein Doppelklick auf so ein File. Laden kann man sowohl Files, die Koordinaten, usw. enthalten als auch solche, die die Quell-URLs enthalten (s.3.10a). Um letztere zu speichern, muss man aber den Export-Button unter Quelle verwenden

4 Fileformat Taho-Source

In einem *.taho File sind die Kartenquellen im Block <mapallsrc>. Für jede Quelle gibt es einen Block <src>. In diesem kommen die folgenden Tags vor:

4.1 <name>

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

4.2 <prefix>

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

4.3 <type>

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

4.4 <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.

4.5 <url>, <offset>, <ksize> und <mapspf>

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

Die Basiskarte hat <type>0</type> 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. Im entsprechenden Taho-File (ab Vers2.10) gibt es zusätzlich für jede Quelle einen Prefix der für die Filenamen benutzt wird.

4.5.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 Kartenummer/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 Kartenummer berechnet sich nach:

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

4.5.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>

5 Erfahrungsberichte

Ich selber nutzte Glopus bin aber inzwischen auf ein Android-Smartphone umgestiegen, leider gibt es Glopus nicht dafür. Dort nutze ich Karten die sich selber die Karten holen, also brauche ich Tahoe nicht mehr. Also nicht über Fehler ärgern sondern sie mir mitteilen. Die Wahrscheinlichkeit, daß ich sie selber bemerke ist gering.

6 Rechtliches und Quelle:

Dieses Programm steht unter der GPL V3 Lizenz.

<http://www.gnu.de/documents/gpl.de.html>

Eine Kopie liegt dem Programm bei. Ältere Versionen standen unter der CreativeCommons, da in die aktuelle Version aber viel von taho.pl übernommen wurde und dies unter der GPL steht, habe ich diesen Wechsel vorgenommen.

Die Lizenz der mit diesem Programm geladenen Karten muss auf jeden Fall gewahrt werden. Werden z.B. OSM Karten veröffentlicht muss auf die Quelle hingewiesen werden. Für Details s.: [OSM-FAQ](#)

Bei anderen Quellen muss man selber ermitteln was man darf.

Dort, wo Sie dieses Programm her haben, sollte auch der Quellcode erhältlich sein. Eventuell aber erst ein paar Tage nach der Binary-Version. Falls nicht, gibt es ihn auf meiner HP, aber ggf. nicht in der aktuellen Version, falls jemand anders eine neue Version herausgegeben hat. Deshalb bitte ich alle, die an dem Programm arbeiten, mir dies mitzuteilen, damit es nur eine Version gibt.

<http://www.dimitri-junker.de/html/openstreetmap.html>

7 Änderungen seit Version 2.06

3.09 vom 23.12.2013

- Die 3.08 lief nicht unter Windows XP und auf Systemen ohne installiertes Visual Studio fehlte die mfc120.dll. Beides sollte mit dieser Version behoben sein.

3.08 vom 21.12.2013

- Da die OSM-Export-Seite nicht mehr für die Zwecke dieses Programms nutzbar war habe ich eine andere Seite angepasst und hier eingebaut.
- Anpassung an Visual Studio 2013 und Windows 8.1. S.a. [Vorwort zu Version 3.08](#)

3.07 vom 2.8.2013

- Kartenquellen aufgeräumt, statt Cycle1 und Cycle2 nur noch Cycle und Osmarender gibt es nicht mehr. Damit sie nicht über ein Tahoe File doch reinkommen sind diese Namen auf einer schwarzen Liste.
- Bei „nur Tiles laden wurden Overlay-Tiles ignoriert
- Interne Änderungen.

3.06 vom 9.8.2012

- Beim Umschalten zwischen Vektor und Pixelkarte wurde png.tile nicht versteckt.
- Eine weitere Änderung um den Lizenz-Bedingungen der <http://www.wanderreitkarte.de/> näher zu kommen.

Diese fordert u.a. eine eigene User Agent Identifikation: diese kann jetzt auf verschiedene Werte gesetzt werden s. [User Agent ID](#)

- Interne Änderungen (Lesen aus XML)

3.05 vom 8.5.2012

- durch eine Änderung in der 3.03 funktionierte die Größe 8192*8192 nicht mehr.
- Erweiterung des Fugawi-Kalibrationsfile um vp-Zeilen

3.04 vom 3.9.2011

- Ohne default.taho wurde die maximale Anzahl Threads auf 0 gesetzt, was natürlich nicht funktioniert.

3.03 vom 1.8.2011

- Die maximale Anzahl Threads kann jetzt eingestellt werden s. [3.10a](#)
- Karten können weiterhin als png oder jpg gespeichert werden, allerdings können erstere jetzt auch die Endung png.tile erhalten dies ist z.B. für den osmtracker unter Android nötig. [s. 3.6d](#)
- Man kann jetzt auch 256*256 Pixel Karten erzeugen. [s. 3.6a](#)
- Man kann jetzt auch explizit angeben, daß die Karten in Unterordnern sortiert werden sollen wie auf dem Server [s. 3.3a](#)

3.02 vom 22.5.2011

- Es wurden die falschen Kalibrationsfiles erzeugt

3.01 vom 16.2.2011

- „nur“ Fehlerkorrekturen.

3.00 vom 15.2.2011

- Vektorkarten und OSM-Rohdaten können jetzt zusätzlich geladen werden
- u.a. wegen der Vektorkarten habe ich die Oberfläche überarbeitet.

2.12 vom 22.1.2011

- Falls kein defsrc vorhanden war fand Taho keine Tiles, jetzt ist es wieder so wie in älteren Versionen, dass ein Set an Quellen fest im Programm gespeichert ist.
- Die Overlays werden jetzt alphabetisch geordnet angezeigt.
- Einige zusätzliche Sicherheitstests

2.11 vom 12.12.2010

- Die Anzahl der fest einprogrammierten Kartenquellen hat sich deutlich erhöht, außerdem können jetzt neben den Basiskarten auch noch Overlays verwendet werden.
- Leseroutinen für XML Files (z.B. html, taho) überarbeitet
- sonstige Bugfixes

2.10 vom 10.11.2010

- //CHG: TAHO 2.10b SG
Neues Kalibrationsfile für GPS-Tuner
- //CHG: TAHO 2.10d DYJ
Die Kalibrationsfiles waren bei nicht quadratischen Files falsch
- //CHG: TAHO 2.10e DYJ
Beim Laden eines *.taho wurde die Datenquelle (Mapnik,...) nicht übernommen.
- //CHG: TAHO 2.10f DYJ
Neue Optionen für den Fall, dass man Taho auf einem USB-Stick o.ä. nutzen will.
- //CHG: TAHO 2.10g DYJ
Die *.taho benutzten das falsche charset, was zur Folge hatte, dass Umlaute falsch angezeigt wurden wenn man es sich per Browser ansah.

- //CHG: TAHO 2.10h DYJ
„aus URL“ erkannte den Zoomlevel nicht mehr, da sich die URL auf der Export-Seite geändert hat.
- //CHG: TAHO 2.10i DYJ
Man kann jetzt jedem Renderer ein eigenes Prefix für die Filenamen geben, also nicht immer OSM_... Default-mäßig werden die 3 ersten Buchstaben des Namens benutzt.
- interne Änderungen s. „2.10a SG“, „2.10c DYJ“
Teils nur interessant für Programmierer, teils aber auch nötig um Abstürze zu eliminieren.

2.9 vom 12.8.2010

- Da das Formular überquoll, sind selten benötigte Einstellungen in einen Optionen Dialog verschoben worden. Außerdem hat das Hauptfenster ein Menü erhalten, wodurch die Anzahl der Buttons verringert werden konnte.
- Taho kann jetzt auch kml und kmz Dateien erzeugen, u.a für Google-Earth und einige Garmin Geräte.
- Die Quell-URLs können jetzt flexibler definiert werden → mehr Quellen nutzbar.

2.8 vom 12.7.10

- Taho ist jetzt mehrsprachig
- Fehlerkorrektur: Taho verweigerte den download mit dem höchsten Zoomlevel.
- Änderung: default.taho und defsrc.taho werden jetzt in „Eigene Dateien“ geschrieben, in einen Ordner „DYJ\OSM“. Falls dort noch nichts vorhanden ist, wird ggf. das entsprechende File aus dem Programmordner dorthin kopiert.

2.7 vom 22.4.10

- kleinere Änderungen des Formulars
- Neu: Export-Größe frei. S. [3.6](#).
Neu: Kalibrations-File für UI-View
- Änderung: beim Download im Modus Größe=keine werden die wirklichen Filenamen angezeigt.

8 Bekannte Probleme und Aussicht

- Evtl. gibt es noch Probleme mit Karten die den 180. Längengrad kreuzen oder in die Polregionen reichen (dort gibt es keine OSM-Karten)
- Die Taho-Files (Kartenquellen) sollten besser editierbar werden als per Editor)
- Die Auswahl des Kartenbereichs sollte direkt in Taho geschehen, nicht über Bbox-Tool
- Bei den Vektorkarten ist sicher noch viel machbar, u.a. deren direkte Erzeugung aus der OSM-Datenbank.

Dimitri Junker