

Taho.exe 4.12

31.8.19

Inhaltsverzeichnis

1	Vorwort	2
1.1	Vorwort zur Version 3	2
1.2	Vorwort zur Version 3.08	2
1.3	Vorwort zur Version 4.00ß	2
1.4	Vorwort zur Version 4.08	2
2	Installation	3
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	3
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	5
3.6.c	Zoom-Stufen	5
3.6.d	File-Typ	5
3.6.e	Kalibratgithionsfiles	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)	6
3.10	Optionen (Menü Bearbeiten)	6
3.10.a	Anzahl Threads	6
3.10.b	Kartenquelle	6
3.10.c	Cache Gültigkeit	7
3.10.d	Packer	7
3.11	Einstellungen (Menü Datei)	8
4	Fileformat Taho-Source	8
5	Erfahrungsberichte	8
6	Rechtliches und Quellcode	8
6.1	Lizenz	8
6.2	Quellcode	9
7	Änderungen seit Version 3.00	9
8	Bekannte Probleme und Aussicht	12
9	Andere Programme	12
9.1	Maperitive	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.

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 nicht mehr, 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.

1.3 Vorwort zur Version 4.00 β

Hier gilt i.W. Das gleiche wie zur Version 3.08 nur noch stärker, denn diese Version ist fast eine Neuprogrammierung, nicht mehr mit den Microsoft Tools sondern mit QT-Creator entwickelt. Bei so massiven Änderungen können natürlich auch Fehler hinzugekommen sein. Berichtet sie mir und wenn eine Korrektur zu lange dauert probiert die 3.09 und ggf. die 3.07. Letztere könnte vor allem auf alten Windowsversionen die beste sein.

1.4 Vorwort zur Version 4.08

Intern hat sich wieder einiges getan, Ich habe ein Update von QT gemacht, und gebe jetzt sowohl eine 32bit als auch eine 64bit Version raus. Seltsamerweise gibt es den MinGW Compiler für das aktuelle QT 5.12 nur in der 64 Bit Version aber für 5.11 nur in der 32Bit Version. Also haben die beiden Versionen unterschiedliche QT-Libs, aber das dürfte egal sein. Da es inzwischen einen weiteren Mitstreiter, Jan Kovich, gibt werden die Quellcodes ab jetzt per [Github](#) verteilt.

2 Installation

Der Installer sollte alles nötige erledigen. Je nach gewähltem Paket wird eine 32 oder 64bit Version installiert. Wechselt man von einer zur anderen muss man die alte ggf per Hand deinstallieren. 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.

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. Seit Version 4 benutzt Taho die QT-Tools zur Übersetzung. Falls jemand eine weitere Sprache hinzufügen will steht die relevante Anleitung [hier](#) Allerdings auf Englisch. Die nötigen ts Files befinden sich im Quellcodepaket. Derzeit sind dies taho_en.ts, taho_fr.ts und taho_xx.ts Die beiden ersten enthalten die Übersetzung ins Französische und Englische, letztere nur die Originalsprache (Deutsch). Will also jemand Fehler in fr oder en korrigieren benutzt er die entsprechende Datei. Will jemand eine neue Sprache hinzufügen benenne man letztere entsprechend um, z.B. taho_es.ts für Spanisch. Das Resultat mir dann bitte zuschicken. Für kleinere Korrekturen ist es natürlich einfacher mir eine Mail zu schicken. Es gibt zwei Besonderheiten:

- „de“ ist die Abkürzung der jeweiligen Sprache. In taho_en.ts steht als Übersetzung also „en“. Dies sollte auch gleich sein zum Kürzel im Dateinamen.
- „Liesmich.pdf“ muß durch den Dateinamen mit der jeweiligen Hilfe übersetzt werden.

3.2 Koordinaten

Durch einen Klick auf den Button „Bbox Tool“ öffnet sich ein Dialogfenster mit einer kurzen Anleitung und der Standardbrowser mit der Seite <<http://www.oche.de/~junker/OSM/bbox-tool/bbox.html>>. 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 Klickt man dann im o.g. Dialogfenster auf OK wird dies analysiert und die Koordinaten eingetragen

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. Eine Sonderrolle spielt: „lokal Dir“ hier dient ein lokales Verzeichnis als Quelle. Wenn man also mit Taho oder z.B. [Maperitive](#) Tiles heruntergeladen hat oder selber, z.B mit [Maperitive](#), gerendert hat kann man diese Verwenden. Wählt man „lokal Dir“ öffnet sich eine Fileauswahl mit der man ein Tile auswählen kann. Also nicht etwa das Verzeichnis sondern ein Tile, dies ist weniger Fehleranfällig.

Bei einigen Quellservern wie [Thunderforest](#) und [Maptiler](#) muß man sich anmelden. Die beiden genannten haben kostenlose Varianten. Die URL enthält dann eine persönliche ID. Wählt man so eine Quelle aus wird geprüft ob die ID bereits bekannt ist, falls nicht wird der User danach gefragt. Bricht man dies ab wird die Auswahl der Quelle rückgängig gemacht.

Es gibt auch Server bei denen man sich mit Username und Password anmelden muss. Derzeit ist davon aber noch keine im defsrcP.Taho. Wer aber so eine Seite nutzen will kann dies. URL usw werden ganz normal in eines der zwei taho files eingetragen. Die nötigen Login Daten fragt Taho bei Bedarf ab und speichert sie dann in mydefsrc.taho. Bei der Eingabe muss man in einer Zeile Name und Passwort hintereinander durch Doppelpunkt getrennt eingeben. Dies sollte kein Problem sein, da dies ähnlich im html-Standard definiert ist.

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 vier 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 zusammengebaut. Es werden also nur die Kacheln geladen. Es gibt Software, die damit arbeiten kann. Kalibrationsfiles sind dabei nicht nötig.
3. QTOffline: Offline Karten für [QTLocation](#) Nutzt man dies zum ersten Mal wird ggf abgefragt wo der Ordner für QTLocation Karten ist.

4. 256*256 eigentlich keine Besonderheit, aber es erscheint unnötig, da ja auch die Tiles bei „keine“ (s. vorletzter 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 Kalibratgithionsfiles

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 mehrere 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 Falle 2 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.)
- Einstellungen zur Kartenquelle vornehmen
- Verzeichnis für QTLocation Offline Karten.
- den Packer und seine Syntax für die Kommandozeile bestimmen.

3.10.a 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 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.\)](#)

Default mäßig nutzt Tahoe so viele Threads wie es Kerne gibt es sei denn hier wird etwas anderes eingestellt oder für eine Quelle gibt es Beschränkungen.

3.10.b Kartenquelle

Da sich die URLs ab und zu mal ändern oder jemand neue findet, kann man sie nachladen. Die mir derzeit bekannten Karten sind in defsSrcP.taho enthalten. Außerdem wird versucht mydefsSrc.taho nachzuladen, dies ist der Ort für persönliche Quellen und Ids. 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 den entsprechenden Block aus defsSrcP.taho in mydefsSrc.taho kopiert. Zur Syntax des Tahoe-Source Files [s. 4.\)](#)

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 defsSrcP.taho aus dem www zu laden. Ich lege eine aktuelle Version unter:

<http://www.dimitri-junker.de/defsrcP.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 defsrcP.taho. Wer also sein geändertes defsrcP.taho allen zugänglich machen will, kopiere es irgendwohin und setze den Link auf o.g. Seite entsprechend.

Damit man einfach defsrcP.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](#) gab 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.~~

Leider gibt es die Seite nicht mehr falls jemand eine entsprechende neue Quelle findet teilt mir dies bitte mit!

3.10.c Cache Gültigkeit

Die OSM-Kacheln (256*256 Pixelkarten) im 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.d 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](#)).

4 Fileformat Taho-Source

In einem *.taho File sind die Kartenquellen im Block <mapallsrc> bzw <mapPubSrc>. Für jede Quelle gibt es einen Block <src>.

Ausführlicher ist dies in einer gesonderten [Doku](#) beschrieben.

5 Erfahrungsberichte

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

6 Rechtliches und Quellcode:

6.1 Lizenz

Dieses Programm steht unter der [GPL V3 Lizenz](#).

Eine Kopie liegt dem Programm bei. Ältere Versionen standen unter der CreativeCommons, da in die Version 2 aber viel von taho.pl übernommen wurde und dies unter der GPL steht, habe ich diesen Wechsel vorgenommen. Außerdem wird es ab Version 4 mit [QT-Creator](#) entwickelt und dynamisch gegen die [QT-Bibliotheken](#) (V5.x) gelinkt. Diese stehen unter der [LGPL V2.1](#)

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.

6.2 Quellcode

Da es inzwischen einen weiteren Mitstreiter, Jan Kovis, gibt werden die Quellcodes ab jetzt per [Github](#) verteilt.

Auf der obersten Ebene gibt es dyj.pro und taho.pro und die Verzeichnisse Taho und myLibsQT. In beiden Verzeichnissen jeweils auch ein entsprechendes .pro Die erstgenannten sind derzeit noch identisch, falls aber in Zukunft nach der Quellcode für [Dyjtrack](#) oder andere Programme hinzukommen würde dyj.pro alle Programme erzeugen und Taho.Pro nur das eine.

Im Ordner installation befinden sich die beiden iss Files für [InnoSetup](#) um die 32 bzw 64Bit Installationsfiles zu erzeugen.

6.2.a Nomenklatur der Branches/Comits

Zu jeder veröffentlichten Binary Version sollte es ein korrespondierendes Comit geben, also z.B. 4.8. Da man nicht für jede kleine Änderung gleich eine neue Version rausgeben wird schlage ich vor an die Versionsnummer jeweils pro Änderungspunkt der Versionsnummer einen Buschstaben anzuhängen, Also die 4.8a für die erste, dann 4.8b,... und am Ende eben 4.8

Nach dem Linken muß man noch die nötigen dlls... zu Taho exe kopieren, dies geschieht am einfachsten mit dem Tool windeployqt.exe Dies befindet sich bei mir z.B. in: C:\Qt\5.12.0\mingw73_64\bin\windeployqt.exe Der genaue Pfad hängt natürlich vom Installationsort von QT und dem gewählten Kit ab.

7 Änderungen seit Version 3.00

4.12 vom 31.8.2019

- Deutsche Texte aus meinen eigenen Librarys wurden immer noch nicht übersetzt.

4.11 vom 30.8.2019

- Taho kann jetzt auch Tiles von Servern laden die ein login verlangen, s. [hier](#)
- Deutsche Texte aus meinen eigenen Librarys wurden nicht übersetzt.

4.10 vom 25.4.2019

- Downloads z.B. von defsrcP.taho funktionierten nicht wenn das Zieldirectory nicht existierte. Jetzt wird es ggf. erzeugt.

4.09 vom 16.3.2019

- Der Aufruf von bboxtool funktionierte nicht mehr

4.08 vom 1.3.2019

- Als neue Karten-“Größe“ gibt es jetzt QTOffline
- Die allgemeinen Einstellungen (Bearbeiten/Optionen) werden jetzt in der Registry abgelegt, in das Taho-File kommen nur noch die Projekteigenschaften (alles aus dem Hauptfenster)
- Im Zusammenhang mit dem vorigen einiges an der Speicherung der Eigenschaften überarbeitet.
- Da Taho seit längerem per Installer eingerichtet wird machten die Optionen „Relativer Pfad“ usw. keinen Sinn mehr und flogen jetzt endlich raus.
- Verbesserungen der Oberfläche, z.B. sind die Fenster jetzt in der Größe veränderbar.

4.07 vom 6.1.2019

- Als Default Pfad für das Erzeugen von Karten wird jetzt einer unter "Dokumente" verwendet, da im Programmpfad evtl. keine Schreibrechte existieren.
- Beim Default Path gab es Probleme mit dem letzten Backslash.
- Platzhalter in Quell URLs erweitert, statt \$x funktioniert jetzt auch {x} und \${x}, * Groß/Kleinschreibung ist auch egal. Außerdem gibt es einen neuen Platzhalter für IDs die man bei Quellen bekommt bei denen man sich anmelden muss, wie thunderforest und maptiler.
- Die QuellURLs werden jetzt nur noch in den Files defsrcP.taho und mydefsrc.taho gespeichert, nicht mehr im Programm. Dabei merkt sich Taho welche Quellen aus welcher Datei gelesen wurde, so kann defsrcP.taho bei einem Update einfach überschrieben werden ohne mydefsrc.taho zu ändern. Damit ist auch "Export Src" obsolet.
- Deshalb muß jetzt auch im Taho File markiert werden welche Quelle als Standard verwendet

werden soll. Dazu dient Typ=100

- Konnte ein Tile nicht geladen werden wurde bisher versucht es durch ein Tile der Standardquelle zu ersetzen, dies brachte aber nicht viel und war nur aufwendig.
- viele URLs die früher mit http begannen beginnen jetzt mit https
- Seit Tahoe mit einem Installer installiert wird machen die Optionen zu den Pfaden keinen Sinn mehr, also habe ich sie gelöscht.
- Auch die öffentliche Version von Tahoe schreibt jetzt ein Log File im %temp% Ordner und zeigt dieses bei Downloadproblemen an. Bisher machte dies nur die Debug-Version
- andere kleine Bugs behoben

4.06 vom 10.6.2018

- Das Erzeugen von 8Bit Karten funktionierte nicht, dies ist ein Problem der verwendeten QT-Bibliothek. Es gibt aber eine Funktion mit der es funktioniert, also ab jetzt klappt es.

4.05 vom 8.6.2018

- Es gibt 2 URLs die in Tahoe benutzt werden, diese waren:
<http://www.oche.de/~junker/OSM/taho/tna.png>
<http://www.oche.de/~junker/OSM/bbox-tool/bbox.html>
Da die Oche.de mit Inkrafttreten der DSGVO abgeschaltet wurde mußten diese URLs angepasst werden, sie liegen jetzt unter:
<https://dimitrijunker.lima-city.de/OSM/> Diese URL ist ab jetzt im taho-File konfigurierbar.

4.04 vom 4.8.2016

- Einige Quellen wurden abgeschaltet, andere verschoben und neue gibt es auch.
- Die Verwaltung der obsoleten Quellen überarbeitet, sie stehen jetzt auch im *.taho File

4.03 vom 8.10.2015

- Fehlerausgabe erweitert, z.B. wird jetzt angezeigt wenn das Bild nicht erzeugt werden kann, z.B. weil es zu groß ist
- Damit man wenn „frei“ als Größe nicht funktioniert (weil es zu groß wäre) man nicht nur die Alternative 4096*4096 hat habe ich noch 2 Größen hinzugefügt (8k und 16k). Da die maximale Seitenlänge der verwendeten Routinen 32k-1 ist klappt 32k leider nicht mehr.
- Bei der Übernahme der koordinaten aus dem bbox Tool wurden die Nord- und Südgrenzen vertauscht. Dank der Fehlertoleranz der nachfolgenden Routinen funktionierte es zwar war aber unschön ;-)

4.02 vom 31.8.2015

- Man kann jetzt ein lokales Dir als Quelle verwenden, sinnvoll z.B. zur Kombination mit Maperitive
- Ab jetzt mit Installer

4.01 V2 vom 20.8.2015

- Im Binary-Paket fehlte mal wieder eine dll. Am Programm oder dem Source Paket hat sich nichts geändert.

4.01 vom 23.7.2015

- Logfileausgabe überarbeitet
- Fehler beim Tile-Download wurden nicht richtig angezeigt
- Die lonvia Tiles sind umgezogen, um so etwas besser zu behandeln habe ich den Adressen Timestamps gegeben, damit Tahoe entscheiden kann welche Adresse aktuell ist (exe oder defsrc)

- Absturz durch Zugriff auf nicht existierenden Fortschrittsdialog behoben, dieser spezielle existiert nur in der Debugversion

4.00ß V2 vom 6.12.2014

- Im Binary-Paket fehlte eine dll. Am Programm oder dem Source Paket hat sich nichts geändert.

4.00ß vom 1.12.2014

- Erste mit QT-Creator erzeugte Version, deshalb intern fast ein neues Programm, für den User hat sich aber wenig geändert. Es ist aber noch recht wenig getestet, deshalb das ß.

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.

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 Tahoe-Files (Kartenquellen) sollten besser editierbar werden als per Editor)
- Die Auswahl des Kartenbereichs sollte direkt in Tahoe geschehen, nicht über Bbox-Tool
- Bei den Vektorkarten ist sicher noch viel machbar, u.a. deren direkte Erzeugung aus der OSM-Datenbank.

9 Andere Programme

9.1 [Maperitive](#)

Dies ist ein Programm, das ähnliches kann wie Tahoe. Beide Programme haben ihre Stärken und Schwächen im Vergleich zum anderen. Mapperitive ist auf der Quellseite besser, es kann Tiles laden die Tahoe seit der Umstellung auf QT nicht mehr laden kann und als ganz großes Plus kann es auch selber Tiles rendern. Tahoe bietet dagegen auf der Ausgaben-Seite mehr. Also können sie sich gut ergänzen. Als Schnittstelle kann man in Mapperitive Tiles lokal speichern, und diese kann man dann als Quelle für Tahoe verwenden („[lokales Dir](#)“).

Dimitri Junker