Howto Wikimedia articles to MNF mca POI files

1. Introduction

This HowTo gives a brief overview of how to create mca files for MNF using wikimedia dumps and/or wikivoyage dumps and extracting the articles from those dumps. They will become POI files as any other POI file but with longer texts.
We also use te externallinks dump sql files to retrieve coordinates which we need for our POIs.

Note that all the manual steps are not “labour intensive” or time consuming. However, the data processing steps can be extremely time consuming. The steps (download the dump, parse the dump, import into the database) can take hours as you work with files ranging from 100+ MBytes up to 12GB, and these are compressed file sizes, containing up to over 100.000.000 lines (English). Uncompressed they are about 4-6 times bigger.
*Note: I started on a linux box but due to circumstances I had to switch to windows using cygwin, so this howto is cygwin based. For linux / *bsd / Mac OS X folks this shouldn’t be an issue.*

1.1 Requirements and Installation

windows:

- install cygwin (preferably*)
    - install wget, zcat, gzip, gunzip, python, bzip2, sqlite3
- or install http://unxutils.sourceforge.net/
    - install python 2.7 or 3.4
    - install sqlite3

linux/bsd/Mac OS X/windows/windows cygwin:

- Install sqlite3

*: I mention “cygwin (preferably)” as this whole concept is based on UTF-8 encoding. Windows still uses it’s own codepages and windows encoding, and even though it supports utf-8 this is still not 100%. Cygwin is an “encapsulated” 100% utf-8 compliant environment.

1.2 Language versus country

Wikipedia works per language(!), not per country. It is categorized by 2-digit iso-639-1[(1)] code. This means that there isn’t a wikipedia for Austria or Germany or Switzerland. There is a wikipedia in the German language. Note also that you therefore need to use the iso language code and not the iso country code (although they are the same in many cases).

Also the reverse can be true: having multiple languages for one country.
For the rest of this Howto I will refer to "nl" as language code in the examples as I'm Dutch.

1.3 Wikipedia and Wikivoyage

Wikipedia gets more and more articles where geographic coordinates are added (when applicable). Wikivoyage is a spin-off of wikipedia where it only deals about "places of interest / places to go". Unfortunately not all articles have coordinates.
This Howto deals about the creation of mca POI files of both types of articles.

2. SQlite databases and way of working

2.1 Create sqlite database

In this Howto sqlite is used for the simple reason that it is the most simple installable database system on a great number of platforms. Next to that it doesn't require administrator or su(do) rights, although that is always preferred. Sqlite databases are simply files containing one or more tables/indexes/views.
(Of course you can also use mysql/mariadb, postgresql, oracle, ms-sql or any other database as the generated sql files are simple, clean sql files.)

To create a database simply type:
`sqlite3 <database name>`
This will create an empty database in the current folder. (Remember that).
You will end up on the sqlite shell command prompt. type `.exit` or `.quit` to close the program.

2.2 Structure of sqlite databases

In this case we will create an sqlite3 database per language being "<language_code>_wikipedia.db, so for the Dutch language the database "nl_wikipedia.db" holding tables for "nl_externallinks", "nl_wikipedia" and "nl_wikivoyage". You could of course get rid of the language code once you are in a "<language_code>_wikipedia.db", but it gives some extra insight working with your tables and forces you to do really everything by language_code.
All the data is downloaded from dumps.wikimedia.org following the 2-digit country code convention like dumps.wikimedia.org/nlwiki/latest/<all kind of files>

2.3 Our <language_code>_wikipedia.db database

As mentioned: we will have up to 3 tables per language wikipedia database.

- table "nl_externallinks" containing the title, coordinates and some other info.

- table “nl_wikipedia” containing the title and text of the wikipedia articles with the same title as in “nl_externallinks”. Note that the text is limited to 600 characters (right now) due to the limitations in the MNF screen handling of POI note fields.
- table “nl_wikivoyage” containing the title and text of the wikivoyage[4] articles. Also text here is limited to 600 characters.

2.4 Folder structure

We create a folder structure having some sub folders to do our work.
We have a main folder as “top level” folder which we call wikipedia (as example, but you can call it anything you like). Below this folder we have several subfolders:

wikipedia	*Top level folder*
/dumps	*This is where we download our wikipedia dump files*
/images	*This where the icons for our mca files reside*
/output	*This is where our exported csv files get saved*
/scripts	*This is where our shell and python scripts reside*
/sqlite	*This is where our databases <code>_wikipedia.db resides*

3. Downloading, importing and parsing wikipedia and wikivoyage data

3.1 Downloading and importing externallinks

We start by importing the “nl-latest-externallinks.sql.gz” in to our “nl_wikipedia.db” database.
The externallinks is a huge sql script containing all links for all pages for that specific language. For the Dutch pages it contains 6.7 million sql inserts.
Some of those inserts are “geohack” [2,3] inserts statements containing the geographic coordinates of the articles. We need those coordinates and only those links are imported and the coordinates, which we need for our articles, are extracted.
These externallinks import will be our reference table.
Due to some reason there are many duplicates. Maybe caused by all the internal linking that takes place.
(Note: I first tried to derive the coordinates from the articles itself, but they have a different notation per language and even different notations within the language. English has over 8 notations, French 3, German a couple, Tzech language only two notations and Dutch only one. I started with Dutch (of course) and did not forsee the huge complexity and and also the missing (or failing) standardization among and within the languages).

We can use two batchfiles to build the database(s) for the externallinks:

- download_parse_externallinks.sh
- download_parse_all_externallinks.sh

The first one “download_parse_externallinks.sh” is used to download one language and build the database for it. Use the script from the “dumps” sub folder.

Usage: `../scripts/download_parse_externallinks.sh nl`

It will download the ““nl-latest-externallinks.sql.gz”, create the database and populate the table nl_externallinks.
The latter “download_parse_all_externallinks.sh”, also to be run from the dumps folder, will simply download all languages (as far as I inventarized) and build the databases for all languages. (Better run this script overnight).

Both scripts will call the python script “externallinks.py”. In the heading of the “externallinks.py” you will find a few settings for the script.
As this script is based on sqlite you want to make sure that `CREATE_SQLITE = "YES"` is set and `SQLITE_DATABASE_PATH` is set to the correct path in the top of the script.

3.2 Download and import wikipedia dump

3.2.1 Download wikipedia dump

As mentioned: Wikipedia dumps are per language, not per country. The wikipedia dump for that language can cover articles from all over the world.
To download a dump for your language use the wikimediadownloader.py script by simply giving the iso-639-1 language code as paramete to that script.
Usage (from dumps folder): `python ../scripts/wikimediadownloader.py no`
for the wiki dump in the Norwegian language.

3.2.2 Import wikipedia dump into database

use the python script parse_wikidump.py with 2-digit language code.

Usage (from dumps folder): `python ../scripts/parse_wikidump.py de`

This will convert the German language version, which is of course for every German speaker/reader. Note that you need to have run the externallinks script first. Currently there is no error checking.

In the heading of the parse_wikidump.py you will find a few settings for the script.
As this script is based on sqlite you want to make sure that `CREATE_SQLITE = "YES"` is set and `SQLITE_DATABASE_PATH` is set to the correct path.

3.3 Download and import wikivoyage dump

This paragraph is almost 100% identical to 3.2

3.3.1 Download wikivoyage dump

Wikivoyage dumps are also per language, not per country. The wikivoyage dump for that language can cover articles from all over the world.

To download a dump for your language use the wikivoyagedownloader.py script by simply giving the iso-639-1 language code as paramete to that script.
Usage (from dumps folder): `python ../scripts/wikivoyagedownloader.py no`
for the wiki dump in the Norwegian language.
Or use the “download_all_wikivoyagedumps.sh” to download all dumps in one go. Also run this script from the dumps folder. Note that the wikivoyage dumps are muchs smaller than the the wikipedia dumps.

3.3.2 Import wikivoyage dump into database

use the python script parse_wikivoyagedump.py with 2-digit language code.

Usage (from dumps folder): `python ../scripts/parse_wikivoyagedump.py de`

This will convert the German language version, which is of course for every German speaker/reader. Note that you need to have run the externallinks script first. Currently there is no error checking.
You can also run the “parse_all_wikivoyagedumps.sh” script from the dumps folder to import all wikivoyage dumps in one go. This will take several hours.

In the heading of the parse_wikivoyagedump.py you will find a few settings for the script.
As this script is based on sqlite you want to make sure that `CREATE_SQLITE = "YES"` is set and `SQLITE_DATABASE_PATH` is set to the correct path.

4. Export our data to csv

4.1 Wikipedia

We export to csv from sqlite by issueing the script “export_wikidump.sh” or “export_all_wikidumps.sh”
The “export_all_wikidumps.sh” will export all wiki articles of all wiki databases to csv (That did surprise you, didn’t it?). These exported csv will will be available in the folder output.
The”export_wikidump.sh” will only export the csv for the language you specify.
Usage (from the scripts folder): `./export_wikidump.sh de`

4.2 Wikivoyage

We export to csv from sqlite by issueing the script “export_wikivoyagedump.sh” or “export_all_wikivoyagedumps.sh”
The “export_all_wikivoyagedumps.sh” will export all wikivoyage articles of all wiki databases to csv (That did surprise you, didn’t it?). These exported csv will will be available in the folder output.
The”export_wikivoyagedump.sh” will only export the csv for the language you specify.
Usage (from the scripts folder): `./export_wikidump.sh de`

4.3 Check csv files

To check whether the generated csv files are correct you can run “csv_filechecker.py” python script from the output folder using the full(!) csv filename like:

```
python ..\scripts\csv_filechecker.py dewikivoyage.csv
```

Or to check all csv files (cygwin/linux):

```
for i in *.csv; do python ../scripts/csv_filechecker.py $i; done
```

5. Create POI mca files

Now that we have our csv file(s) in the folder output we can create our POI mca files.We have the icon imagase for wikipedia and wikivoyage in the images folder. Of course you can use others if you want to.
Use diggerQT to create them. You can use the images as they are. No further “tweaking” is necessary.
Longitude and Latitude can easily be derived from the csv.
Use “Title” as “Name”.
Use “Remark” and “Content” (preferably in that order) as “Note”.

Appendix

Notes

- As mentioned. The manual work is really simple and takes only minutes of your time. Importing the externallinks (all links) takes up to 2 hours on an i7 with a fast SSD-disk. Importing the wikivoyage articles takes seconds to approximately 30 minutes (fr and de) to over an hour (en). ~~Importing the wikipedia files takes hours, to even days (de, fr, en).~~ Switching to “in memory” processing speeds up the proces by at least a factor 10. Still experimental (like all the rest by the way).
- MNF has some issues as well. The text is currently limited to 600 characters as MNF can’t display longer texts while at the same time showing the options for “Navigate”, “Add to favorites”, Show on Map” and the like. The “Note” display needs an (auto)scrollbar. (Tested on a 4.5” 1280x720 display).
- MNF has memory issues (my conclusion, not confirmed). If you search, you get a result (if available) and you can navigate or display on map. If you scroll long enough through the articles, or scroll long and then select, MNF crashes.

Tips

1. Using your files and sqlite database on an SSD disk will speed up your process by at least a factor 2.
2. Sqlite can also use “in memory” databases and tables. If you have enough ram (>6GB for smaller languages; >12GB for big languages) you can run the process from memory. This will definitely speed up the process by many, many factors (~~untested~~ Currently being tested)
3. ~~Alternatively if you have enough memory: Make a ram disk (disk in memory). This will also speed up the process enormously like in tip 2.~~
4. Try to work from compressed files as indicated in this Howto and as defaultly done in the scripts. It will save you a lot of disk space and also improves the performance of your process.

To be done (what’s wrong and what needs to be improved)

1. At this moment dumps with character sets like Farsi (fa), Greek (el), Chinese (zh), japanese (ja), korean (ko) and the like simply don’t work in the scripts (yet).
2. Better error control. The externallinks script needs to be run first. The consecutive parsing scripts for wikipedia/wikivoyage do not test if this is available, but assume that “everything” is there.

3. Wikipedia/Wikivoyage page parsing can be improved performance wise. However, these wikipedia and wikivoyage mca files only need to be generated 3-4 times per year. Simply let it run overnight (over the weekend) on a fast server/pc.
4. Wikipedia/Wikivoyage page parsing can be improved with regard to delivering “clean texts”.
5. Create a wrapper script to parse all wikipedia dumps in one go (even if this takes days to run)
6. I simply inventarized all languages available for wikipedia. wikivoyage covers not as meany languages but I don’t care for the time being. They error out and the script(s) continue.

References & Links:

(1): http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
(2): http://de.wikipedia.org/wiki/Wikipedia:Technik/Labs/Tools/geohack
(3): https://bitbucket.org/magnusmanske/geohack
(4): http://www.wikivoyage.org/ ; Check your own language