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 the externallinks dump sql files to retrieve coordinates which we need for our POIs.

Note that the manual steps to perform are not “labour intensive” or time consuming. However, the data processing steps can be 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.*
Note2: This manual is in English english, not American english. Hence categorised instead of categorized, etcetera.

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 code pages 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 categorised 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.

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 / where to go / what to see/ where to eat", etcetera. Unfortunately not all of these wikivoyage articles have coordinates.
This Howto deals about the creation of mca POI files for 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, mssql or any other database as the generated sql files are simple, clean sql files.)

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" and optionally "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 language 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. This one is always created and necessary
- 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. This table is optional.

- table “nl_wikivoyage” containing the title and text of the wikivoyage[4] articles. Also text here is limited to 600 characters. This table is optional.

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 wikiscripts as it is the same folder if you do a git clone from github or download a zip (but you can call this top level folder anything you like). Below this folder we have several subfolders:

wikiscripts	*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 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] insert statements containing the geographic coordinates of the articles. We need those coordinates. Only the links with these “geohack” coordinates are imported as we need them for the articles in the wikipedia and wikivoyage dumps.
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, Czech language only two notations and Dutch only one. I started with Dutch (of course) and did not foresee the huge complexity and also the missing (or failing) standardization among and within the languages).

We can use two shell bash files 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 sub folder “dumps”.

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 inventorised) and build the databases for all languages. (Better run this script overnight).

Both scripts will call the python script "externallinks.py" to import the downloaded file into the sqlite database. 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.

4. Downloading, parsing wikipedia and wikivoyage data

4.1 Download, parse and optionally import wikipedia dump

4.1.1 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 folder dumps): `../scripts/wikipediadownloader.sh no`
for (only) the wiki dump in the Norwegian language, or:

```
../scripts/wikipediadownloader.sh no en nl de fa it
```

for the wikipedia dumps for multiple languages.

4.1.2 Parse wikipedia dump to csv

Use the python script parse_wikidump.py with wikitype, in this case wikipedia, and the 2-digit language code.
Usage (from folder dumps): `python ../scripts/parse_wikidump.py wikipedia de`
This will convert the German wikipedia 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 you want to have a csv to create your mca from, you need to make sure that `CREATE_CSV = "YES"` is set. Other options are optional.

4.1.3 Import wikipedia dump into sqlite

If you also want to import the parsed dump into sqlite, make sure the CREATE_SQLITE = “YES” is set and SQLITE_DATABASE_PATH is set to the correct path and ends with a forward slash (/).

4.2 Download, parse and optionally import wikivoyage dump

This paragraph is almost 100% identical to 4.1

4.2.1 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.
As there are not so many wikivoyage dumps and as they are really small compared to wikipedia dumps, a script is used to download them all. If you don’t want that simply modify the script.
Usage (from folder dumps): `../scripts/download_all_wikivoyagedumps.sh`

4.2.2 Parse wikivoyage dump to csv

use the python script parse_wikidump.py with wikitype, in this case wikivoyage, and the 2-digit language code.
Usage (from folder dumps): `python ../scripts/parse_wikidump.py wikivoyage 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 about an hour on an i-7 with ssd disk.

In the heading of the parse_wikidump.py you will find a few settings for the script.
As you want to have a csv to create your mca from, you need to make sure that `CREATE_CSV = “YES”` is set. Other options are optional.

4.2.3 Import wikipedia dump into sqlite

If you also want to import the parsed dump into sqlite, make sure the CREATE_SQLITE = “YES” is set and SQLITE_DATABASE_PATH is set to the correct path and ends with a slash (/).

5. Export our data to csv

This chapter is optional. You only need it if you import your parsed pages into sqlite and did not write a csv file.

5.1 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 folder scripts): `./export_wikidump.sh de`

5.2 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 folder scripts): `./export_wikidump.sh de`

6. Check and clean 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
```

Also, **and this is actually a “must do”**, you need to “clean” the csv files. There are still some remnants from HTML code which can make MNF crash when not removed.

Run the “clean_csv.sh” script from the output folder where your csv files are.

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

to clean one csv file. To clean multiple csv files simply type (from output folder):

```
../scripts/clean_csv.sh nl de no en fr
```

7. 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 images 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 *(note that the csv has them in the “default” order Latitude, Longitude while digger expects Longitude, Latitude)*.

Use “Title” as “Name”.

Use “Remark” and “Content” (preferably in that order) as “Note”.

Appendix

Notes

- 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. The note itself is not limited, it really has to do with displaying on the screen (Tested on a 4.5" 1280x720 display).

Tips

1. Using your files and sqlite database on an SSD disk will speed up your process by at least a factor 2.

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

1. 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.
2. Wikipedia/Wikivoyage page parsing can be improved with regard to delivering "clean texts".
3. I simply inventorised all languages available for wikipedia but maybe there are more. Wikivoyage currently only covers a subset of languages.

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 if your own language is available)*