

# SM to SQLite for Mission-X



<b>Preface</b>	<b>3</b>
<b>Pre-Requisites</b>	<b>3</b>
<b>Download OSM data</b>	<b>4</b>
<b>Search Open Scenery Data on Map</b>	<b>5</b>
<b>Converting and Filtering</b>	<b>6</b>
<b>Converting OSM to SQLite</b>	<b>8</b>
<b>Installing Perl On Linux</b>	<b>10</b>
<b>Installing Perl On Windows</b>	<b>11</b>
<b>Changelog</b>	<b>12</b>

## Preface

Laminar Research uses OpenStreetMap (OSM) data to draw their network and other information. Unfortunately there is no API or easy way to parse this information and use it in X-Plane.

In order for the plugin to use the same data, I have worked on a “simple” workflow to:

- **Step 1:** Download, convert and filter out OSM data into an XML OSM format.  
In this step we are using the original OSM tools to do the job.  
Depending on the amount of the information this can take from few seconds to few minutes.
- **Step 2:** Reduce the OSM file into SQLite database, so Mission-X plugin will be able to read from it.  
This step produces two databases, one holds “nodes” and the other holds the “ways” which we need for Mission-X.  
The nodes database we can delete at the end of the process, unless you want to re-run the script and thus skip this step for better performance (depends on the script flags).

Once these steps are done, you should copy the “ways” database into “{missionx plugin folder}/db”.

**Please remember that using OSM data is only for Medevac based helicopters missions.**

## Pre-Requisites

OSM tools binaries should be available from the OSM site.  
Prefer the 64bit version, if you can, over the 32bit one.

“Osmconvert”: <https://wiki.openstreetmap.org/wiki/Osmconvert>

“Osmfilter”: <https://wiki.openstreetmap.org/wiki/Osmfilter>

On Linux OSes you might have a package that you could just install from.  
On my Linux Mint distribution, you could install the **osmctools** using “apt install” command.

## Download OSM data

The ability for Mission-X to use the OSM data depends on the quality of the data and the area it covers. If the area is small, it might not cover the area Mission-X needs to look for, therefore it will fallback to none OSM target search.

A good place to download OSM data can be found in: [geofabrik site](#).

The file format we should download is the “**.pbf**” and not the “osm” since we will create the “osm” manually while hopefully reducing its size.

I suggest starting with small regions and not continents. If you must, then only download the country/state you will conduct your medevac adventures.

Remember, the larger the file the more time and size your database will be.

GEOFABRIK downloads

### OpenStreetMap Data Extracts

The OpenStreetMap data files provided on this server do **not** contain the user names, user IDs and changeset IDs of the OSM objects because these fields are assumed to contain personal information about the OpenStreetMap contributors and are therefore subject to data protection regulations in the European Union.

[Extracts with full metadata](#) are available to OpenStreetMap contributors only.

Welcome to Geofabrik's free download server. This server has data extracts from the [OpenStreetMap project](#) which are normally updated every day. Select your continent and then your country of interest from the list below. (If you have been directed to this page from elsewhere and are not familiar with OpenStreetMap, we highly recommend that you read up on OSM before you use the data.) This open data download service is offered free of charge by Geofabrik GmbH.

Willkommen auf dem Geofabrik-Downloadserver. Hier gibt es Daten-Auszüge aus dem [OpenStreetMap-Projekt](#), die normalerweise täglich aktualisiert werden. Wählen Sie aus dem Verzeichnis unten den Kontinent und ggf. das Land, für die Sie Daten benötigen. (Wenn Sie von anderswo auf dieser Seite gelandet sind und von OpenStreetMap nichts wissen, dann ist es empfehlenswert, sich mit dem Projekt vertraut zu machen, bevor Sie mit den Daten arbeiten.) Diese Downloads werden von der Geofabrik GmbH kostenlos angeboten.

Click on the region name to see the overview page for that region, or select one of the file extension links for quick access.

Sub Region	Quick Links		
	.osm.pbf	.shp.zip	.osm.bz2
<a href="#">Africa</a>	<a href="#">[.osm.pbf]</a> (3.8 GB)	✗	<a href="#">[.osm.bz2]</a>
<a href="#">Antarctica</a>	<a href="#">[.osm.pbf]</a> (29.0 MB)	<a href="#">[.shp.zip]</a>	<a href="#">[.osm.bz2]</a>
<a href="#">Asia</a>	<a href="#">[.osm.pbf]</a> (8.2 GB)	✗	<a href="#">[.osm.bz2]</a>
<a href="#">Australia and Oceania</a>	<a href="#">[.osm.pbf]</a> (783 MB)	✗	<a href="#">[.osm.bz2]</a>
<a href="#">Central America</a>	<a href="#">[.osm.pbf]</a> (380 MB)	✗	<a href="#">[.osm.bz2]</a>
<a href="#">Europe</a>	<a href="#">[.osm.pbf]</a> (21.6 GB)	✗	<a href="#">[.osm.bz2]</a>
<a href="#">North America</a>	<a href="#">[.osm.pbf]</a> (9.3 GB)	✗	<a href="#">[.osm.bz2]</a>
<a href="#">South America</a>	<a href="#">[.osm.pbf]</a> (2.0 GB)	✗	<a href="#">[.osm.bz2]</a>

🇬🇧 Not what you were looking for? Geofabrik is a consulting and software development firm based in Karlsruhe, Germany specializing in OpenStreetMap services. We're happy to help you with data preparation, processing, server setup and the like. [Check out our web site](#) and contact us if we can be of service.

🇩🇪 Nicht das Richtige dabei? Die Geofabrik ist ein auf OpenStreetMap spezialisiertes Beratungs- und Softwareentwicklungsunternehmen in Karlsruhe. Gern helfen wir Ihnen bei der Datenaufbereitung, Datenkonvertierung, Serverinstallation und ähnlichen Aufgaben. [Besuchen Sie unsere Webseite](#) und sprechen Sie mit uns, wenn wir Ihnen helfen können.

You can drill down into regions and countries and download only the area you are interested in.

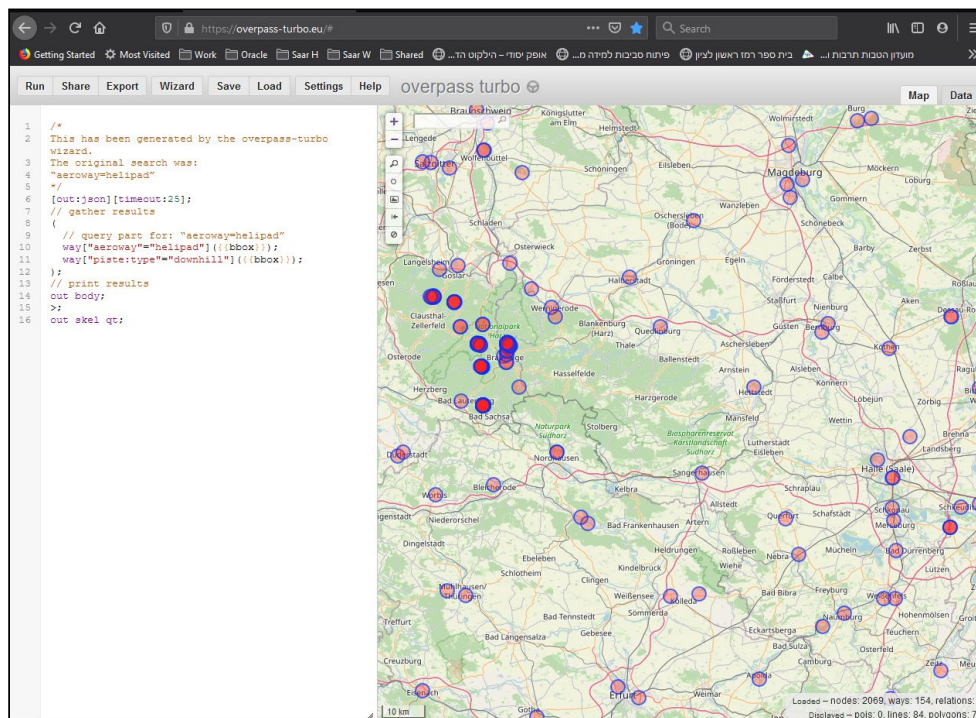
I believe that there are sites that allow you to custom define the OSM region but I haven't checked those.

# Search Open Scenery Data on Map

Site to query information: <https://overpass-turbo.eu/#>

Thanks to “@Daikan” for this tip.

The overpass-turbo site allows you to pick an area based on a map and then query for specific key=value. This can assist in building a better OSM database and your sqlite queries.



```

/*
This has been generated by the overpass-turbo wizard.
The original search was:
"aeroway=helipad"
*/
[out:json][timeout:25];
// gather results
(
  // query part for: "aeroway=helipad"
  way["aeroway"="helipad"]({{bbox}});
  way["piste:type"="downhill"]({{bbox}});
);
// print results
out body;
>;
out skel qt;
  
```

You can modify this example or use the wizard to search for relevant information. This can also assist in building better “sqlite” database files from the OSM and queries to fetch the information from your templates.

## Converting and Filtering

Once you have your “.pbk” file of the area at hand, we should start the optimization process. Place the file in a disk where you have enough space in a temporary folder. Main reason, the OSM file is much larger than the “pbk” one because of different file formats.

I’ll provide here a few commands and flags that I use. You are more than welcome to modify these commands to filter out as many metadata as you can which we do not need. Since there is a huge diversity of metadata attributes it is hard to filter them all out. The ones that we most care about related to the “highway” key attribute and “name” but I could not figure out how to just pick those.

**Step 1: convert the “pbk” file to an “o5m” format so osmfilter will do his work on it.**

In this example I use the “windows” binaries and on “Washington state pbk”:

```
osmconvert64-0.8.8p.exe washington-latest.osm.pbk --drop-relations
--drop-author --drop-version -o="washington-latest.osm.o5m"
```

**Step 2: Filter out as many attributes as you want/need to have smaller OSM file**

```
osmfilter.exe washington-latest.osm.o5m --drop-ways="natural=coastline"
--keep="aeroway= amenity= faa= icao= int_name= highway=primary =primary
=tertiary =service =secondary =track =residential =path =living_street
=trunk =motorway_link =tertiary_link =trunk_link =road =secondary_link
=pedestrian =primary_link =abandoned =planned =raceway =bridleway
piste*= waterway= " --drop-node-way-tags="created_by= source*= oneway=
lanes= horse= surface= maxspeed= ref= hgv*= sidewalk= name*= old_name*=
isolation= prominence= wiki* alt*= attrib*= geobase*= tiger*= gnis*=
zoom*= ele= note= intermittent= website= layer= boat= ship= lit= _*=
abutters= access*= acres= addr*= aerialway*= aerodrome*= air*= alcohol=
animal*= area*= alcohol= art*= basketball= bathroom*= beds= bicycle*=
bin= blind= blood*= brand*= building:= bus*= button_operated=
capacity=* car*= change:= charge= cinema*= city= climbing:= N*= F*=
R*= TO*= boundary= bridge:= building= cuisine*= cycleway*= fixme= foot=
from*= gener*= golf= lan*= le*= max=* motor*= old*= op*= parking*= pay*=
pos*= pro*= rec*= ref*= rel*= roo*= sac*= sea*= ser*= shop*= sid*= smo*=
sport*= super*= take*= targ*= to_add*= toi*= traf*= tur*= twi*= ve*=
vol*= wal*= web*= whee*= wi*= zi*= " --drop="access=no" >
washington-latest.osm
```

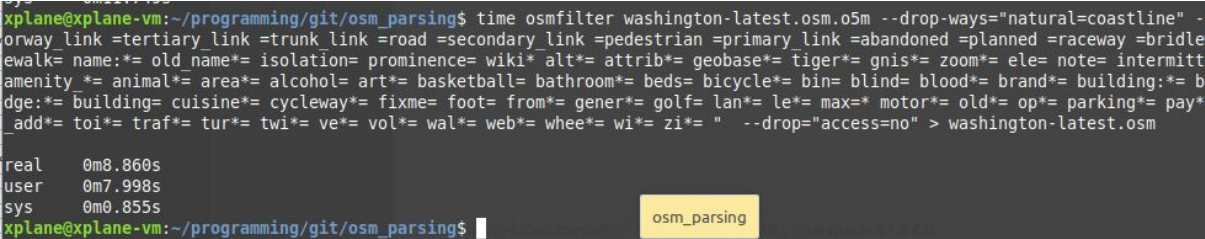
The command is quite long because I want to filter out as many attributes as I can. This will reduce the size of the database we will create later on. You could ignore the

`--drop-node-way-tags` flag, you will just have a bigger OSM file and database, nothing wrong with that.

### Here is the same command only in non format for easier copy

```
osmfilter {filename}.osm --drop-ways="natural=coastline" --keep="aeroway= amenity= faa=
icao= int_name= highway=primary =primary =tertiary =service =secondary =track =residential
=path =living_street =trunk =motorway_link =tertiary_link =trunk_link =road =secondary_link
=pedestrian =primary_link =abandoned =planned =raceway =bridleway piste*= waterway= "
--drop-node-way-tags="created_by= source*= oneway= lanes= horse= surface= maxspeed= ref=
hgv*= sidewalk= name*= old_name*= isolation= prominence= wiki* alt*= attrib*= geobase*=
tiger*= gnis*= zoom*= ele= note= intermittent= website= layer= boat= ship= lit= _*=
abutters= access*= acres= addr*= aerialway*= aerodrome*= air*= alcohol= animal*= area*=
alcohol= art*= basketball= bathroom*= beds= bicycle*= bin= blind= blood*= brand*=
building:= bus*= button_operated= capacity=* car*= change:= charge= cinema*= city=
climbing:= N*= F*= R*= TO*= boundary= bridge:= building= cuisine*= cycleway*= fixme=
foot= from*= gener*= golf= lan*= le*= max=* motor*= old*= op*= parking*= pay*= pos*= pro*=
rec*= ref*= rel*= roo*= sac*= sea*= ser*= shop*= sid*= smo*= sport_*= super*= take*= targ*=
to_add*= toi*= traf*= tur*= twi*= ve*= vol*= wal*= web*= whee*= wi*= zi*= "
--drop="access=no" > {filename}.osm
```

Feel free to modify this command if you want to keep some of the attributes, and add attributes that are specific to your OSM region.



```
xplane@xplane-vm:~/programming/git/osm_parsing$ time osmfilter washington-latest.osm --drop-ways="natural=coastline" -
orway_link =tertiary_link =trunk_link =road =secondary_link =pedestrian =primary link =abandoned =planned =raceway =bridle
ewalk= name:= old name*= isolation= prominence= wiki* alt*= attrib*= geobase*= tiger*= gnis*= zoom*= ele= note= intermitt
amenity *= animal*= area*= alcohol= art*= basketball= bathroom*= beds= bicycle*= bin= blind= blood*= brand*= building:= b
dge:= building= cuisine*= cycleway*= fixme= foot= from*= gener*= golf= lan*= le*= max=* motor*= old*= op*= parking*= pay*
_add*= toi*= traf*= tur*= twi*= ve*= vol*= wal*= web*= whee*= wi*= zi*= " --drop="access=no" > washington-latest.osm

real    0m8.860s
user    0m7.998s
sys     0m0.855s
xplane@xplane-vm:~/programming/git/osm_parsing$
```

Here is an example from a Linux VM. You can see it only took ~10 seconds to convert to OSM file format. On windows it can take much more, depending on the number of filter flags you define.



## Converting OSM to SQLite

The script that reduces the OSM file into SQLite database is a custom built, and was written in perl. Therefore it needs to have certain libraries installed with it.

I will explain the perl preparation after this topic.

Place the script inside the temp folder or in other folder and execute the following, keep in mind that we can modify the files names according to the ones we have and want as output:

```
$ perl mx_osm_to_sqlite.pl file=washington-latest.osm task=all
outfile=osm_washington.db
```

The script needs flags to work. You have to define the:

“file” flag, which is the input “osm” file.

“task” flag which can be “all,node or way”.

Optionally the “outfile” flag which I highly recommend to configure since it allows you to define a custom output name for your database file and thus distinguish it from the others.

The output file should be copied into the “*{Mission-X plugin}/db*” folder.

This process can take some time, depending on the amount of data you have.

To see the script options, you should execute:

```
perl mx_osm_to_sqlite.pl
```

or

```
perl mx_osm_to_sqlite.pl help=yes
```

```
xplane@xplane-vm:~/programming/git/osm_parsing$ perl mx_osm_to_sqlite.pl help=yes
This script was written to convert an OSM <node> and <way> elements into tables and rows of information.
This should allow for faster search of data. It is strongly advised to prepare an OSM of specific area or region you would like to query, and not a whole continent.

Syntax:
=====
{script name} *file={filename} *task=[all|node|way] rebuild_node_data=[yes|no] [outfile={file name}] [nodefile={file name}] rebuild_node_tag_data=[yes|no] rebuild_ways=[yes|no]

Example:
* = mandatory
{script name} file=alaska_filtered.osm task=all          Will parse and rebuild all <node> and <way> tags in file. This includes table rebuild.
{script name} file=alaska_filtered.osm task=node        Will parse and rebuild only <node> tags in file. It won't rebuild node tables
{script name} file=alaska_filtered.osm task=node rebuild_node_data=yes Will parse and rebuild only <node> tags in file. It will rebuild node tables but won't pro
xplane@xplane-vm:~/programming/git/osm_parsing$
```

Here is an example for a script output. You can see that it provides some timing stats for every 500K of elements processed and the time it took it to run.



```

After rebuild tables.
Doing <node>, task: all.....500K - 18.32 sec
.....500K - 36.67 sec
.....500K - 55.59 sec
.....500K - 74.40 sec
.....500K - 93.44 sec
.....500K - 112.93 sec
.....500K - 132.67 sec
.....500K - 152.54 sec
.....500K - 171.66 sec
.....500K - 191.86 sec
.....500K - 211.24 sec
.....500K - 230.49 sec
.....500K - 249.64 sec
.....500K - 268.83 sec
.....500K - 287.88 sec
.....500K - 307.40 sec
.....500K - 328.38 sec
.....500K - 348.73 sec
.....
Doing <way>, task: all.....500K - 532.13 sec
.....500K - 674.78 sec
.....drop table if exists bounds
Dropping bounds table.
create table if not exists bounds as select min (t1.lat) as min_lat, max(t1.lat) as max_lat, min(t1.lon) as min_lon, max(t1.lon) as max_lon
Building bounds table.
create index id_way_street_idx on way_street_node_data(id)
Building id_way_street_idx index.
create index id_way_tag_idx on way_tag_data(id)
Building id_way_tag_idx index.

Elapsed: 685.76
=====

Overall: 826354 tags read.
Worked on: 10097116 node Lines
Worked on <node>s: 9270762
Worked on <way>s: 270243
    match_ref counter:: 4466360

real    11m25.826s
user    11m15.049s
sys     0m10.080s

```

This test was done on Mint 19.3 VM. I find it faster than the Native Windows part.

## Installing Perl On Linux

Most Linux distributions come with perl installed on them but not all come with all the libraries perl has to offer and the script uses. In windows the situation is different since their installation includes many common libraries that you won't find in Linux.

Here are the commands I used to make "`mx_osm_to_sqlite.pl`" run on Ubuntu 20.04 (desktop addition).

When you first run the script, you might receive the following:

```
$ perl mx_osm_to_sqlite.pl help=y
```

```
Can't locate DBI.pm in @INC (you may need to install the DBI module) (@INC contains:
/etc/perl /usr/local/lib/x86_64-linux-gnu/perl/5.30.0 /usr/local/share/perl/5.30.0
/usr/lib/x86_64-linux-gnu/perl5/5.30 /usr/share/perl5
/usr/lib/x86_64-linux-gnu/perl/5.30 /usr/share/perl/5.30 /usr/local/lib/site_perl
/usr/lib/x86_64-linux-gnu/perl-base) at mx_osm_to_sqlite.pl line 3.
BEGIN failed--compilation aborted at mx_osm_to_sqlite.pl line 3.
```

```
Other missing librtaries:
Can't locate XML/LibXML.pm
```

This means you need to install a missing library. In this case the DBI which is the database interface library.

You could install it through the "apt" command or through CPAN. Just search the error on the internet and you should find ways to install it (quite easy I may say).

```
Can't locate DBI.pm
```

```
$ sudo apt-get install libdbi-perl
```

```
Can't locate XML/LibXML.pm
```

```
$ sudo apt-get install libxml-libxml-perl or
```

```
$ sudo yum install "perl(XML::LibXML)" or
```

```
$ [root] perl -MCPAN -e shell
```

```
# at prompt type in : install XML::LibXML
```

The "CPAN" option seems to compile the library, so you should do it on the root account instead of the user account (my opinion).

I think the easiest way is to just download the binary from the OS packages.

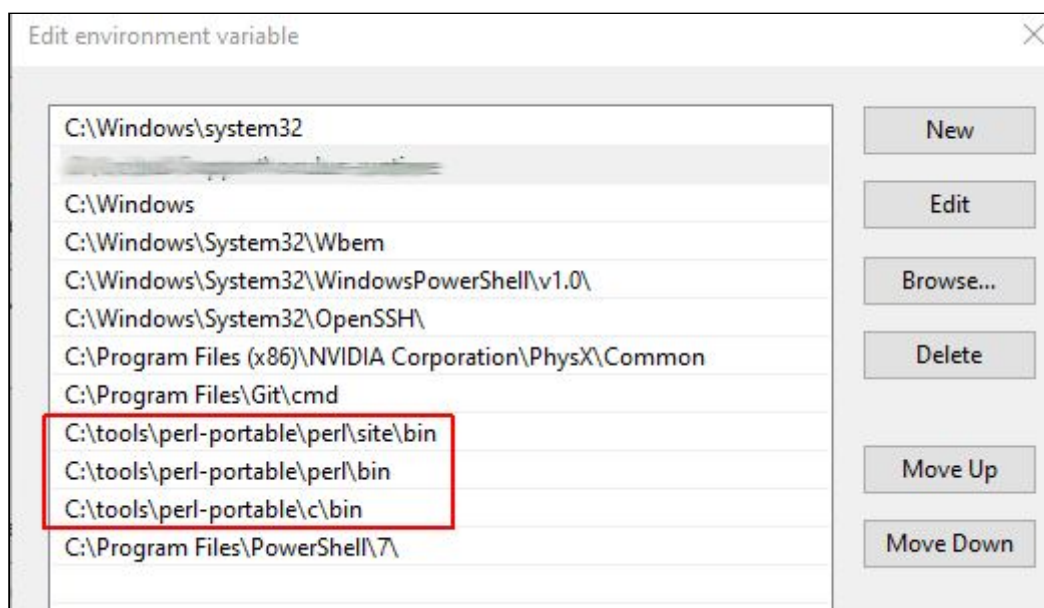
Once you receive an output from the script, you should be good to go.

## Installing Perl On Windows

On windows I just download perl from: <https://www.perl.org/get.html#win32>

I preferred to use the “[Strawberry Perl](#)” since it is 100% Open Source and community driven, and for our script we do not need more.

I did the Portable installation, but you can go ahead and just install the “\*msi” version. In my case, all libraries were present for the script to run, I just had to add the “perl” binary folder to the “PATH” environment variable:



## Changelog

Date	revision	Detail
19-jun-2020	1.1	* Tuned filter to have more metadata relate to airways. Helps with helipad queries. * Added overpass-turbo site
14-jun-2020	1.0.1	Minor modification to the osmfilter flag. Will keep piste* key attributes.