
osm2pgsql

<https://osm2pgsql.org>

osm2pgsql is a tool for loading OpenStreetMap data into a PostgreSQL / PostGIS database suitable for applications like rendering into a map, geocoding with Nominatim, or general analysis.

See the documentation for instructions on how to install and run osm2pgsql.



Features

- Converts OSM files to a PostgreSQL DB
- Conversion of tags to columns is configurable in the style file
- Able to read .gz, .bz2, .pbz and .osm files directly
- Can apply diffs to keep the database up to date
- Support the choice of output projection
- Configurable table names
- Support for hstore field type to store the complete set of tags in one database field if desired

Installing

Most Linux distributions include osm2pgsql. It is available on macOS with Homebrew and Windows builds are also available. See <https://osm2pgsql.org/doc/install.html> for details.

Building

The latest source code is available in the osm2pgsql git repository on GitHub and can be downloaded as follows:

```
1 git clone https://github.com/openstreetmap/osm2pgsql.git
```

Osm2pgsql uses the cross-platform CMake build system to configure and build itself.

Required libraries are

- CLI11
- expat
- proj
- bzip2

-
- `zlib`
 - Boost libraries (for boost geometry)
 - `nlohmann/json`
 - OpenCV (Optional, for generalization only)
 - `potrace` (Optional, for generalization only)
 - PostgreSQL client libraries
 - Lua (Optional, used for Lua tag transforms and the flex output)
 - Python (only for running tests)
 - `Psycpg` (only for running tests)

The following libraries are included in the `contrib` directory. You can build with other versions of those libraries (set the `EXTERNAL_*libname*` option to `ON`) but make sure you are using a compatible version:

- `fmt` (`>= 7.1.3`)
- `libosmium` (`>= 2.17.0`)
- `protozero` (`>= 1.6.3`)

It also requires access to a database server running PostgreSQL 9.6+ and PostGIS 2.2+.

Make sure you have installed the development packages for the libraries mentioned in the requirements section and a C++ compiler which supports C++17. We officially support `gcc >= 8.0` and `clang >= 8`.

To rebuild the included man page you'll need the `pandoc` tool.

First install the dependencies.

On a Debian or Ubuntu system, this can be done with:

```
1 sudo apt-get install make cmake g++ libboost-dev \  
2   libexpat1-dev zlib1g-dev libpotrace-dev \  
3   libopencv-dev libbz2-dev libpq-dev libproj-dev lua5.3 liblua5.3-dev \  
4   pandoc nlohmann-json3-dev pyosmium
```

On a Fedora system, use

```
1 sudo dnf install cmake make gcc-c++ libtool boost-devel bzip2-devel \  
2   expat-devel fmt-devel json-devel libpq-devel lua-devel zlib-devel \  
3   potrace-devel opencv-devel python3-osmium \  
4   postgresql-devel proj-devel proj-epsg pandoc
```

On RedHat / CentOS first run `sudo yum install epel-release` then install dependencies with:

```
1 sudo yum install cmake make gcc-c++ boost-devel expat-devel zlib-devel
2 \
3 potrace-devel opencv-devel json-devel python3-osmium \
  bzip2-devel postgresql-devel proj-devel proj-epsg lua-devel pandoc
```

On a FreeBSD system, use

```
1 pkg install devel/cmake devel/boost-libs textproc/expat2 \
2 databases/postgresql94-client graphics/proj lang/lua52
```

On Alpine, use

```
1 apk --update-cache add cmake make g++ nlohmann-json \
2 postgresql-dev boost-dev expat-dev bzip2-dev zlib-dev \
3 libpq proj-dev lua5.3-dev luajit-dev
```

Once dependencies are installed, use CMake to build the Makefiles in a separate folder:

```
1 mkdir build && cd build
2 cmake ..
```

If some installed dependencies are not found by CMake, more options may need to be set. Typically, setting `CMAKE_PREFIX_PATH` to a list of appropriate paths is sufficient.

When the Makefiles have been successfully built, compile with

```
1 make
```

The man page can be rebuilt with:

```
1 make man
```

The compiled files can be installed with

```
1 sudo make install
```

To install the experimental `osm2pgsql-gen` binary use

```
1 sudo make install-gen
```

By default, the Release build with debug info is created and no tests are compiled. You can change that behavior by using additional options like following:

```
1 cmake .. -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug -DBUILD_TESTS=ON
```

Note that `Debug` builds will be much slower than release build. For production `Release` or `RelWithDebInfo` builds are recommended.

Using the PROJ library

Osm2pgsql has builtin support for the Latlong (WGS84, EPSG:4326) and the WebMercator (EPSG:3857) projection. Other projections are supported through the Proj library which is used by default. Set the CMake option `WITH_PROJ` to `OFF` to disable use of that library.

Using LuaJIT

To speed up Lua tag transformations, LuaJIT can be optionally enabled on supported platforms. This can speed up processing considerably.

On a Debian or Ubuntu system install the LuaJIT library:

```
1 sudo apt-get install libluajit-5.1-dev
```

Configuration parameter `WITH_LUAJIT=ON` needs to be added to enable LuaJIT. Otherwise make and installation steps are identical to the description above.

```
1 cmake -D WITH_LUAJIT=ON ..
```

Use `osm2pgsql --version` to verify that the build includes LuaJIT support. The output should show something like

```
1 Lua 5.1.4 (LuaJIT 2.1.0-beta3)
```

Generalization

There is some experimental support for data generalization. See <https://osm2pgsql.org/generalization/> for details.

Help/Support

If you have problems with osm2pgsql or want to report a bug, go to <https://osm2pgsql.org/support/>.

License

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Contributing

We welcome contributions to osm2pgsql. See CONTRIBUTING.md and <https://osm2pgsql.org/contribute/> for information on how to contribute.