Libosmium Library

Saturday, October 12, 2019 7:25 PM

1. Introduction

Header-only Library

Osmium is a header-only library, so there is nothing to compile to build it. Just include the header files you need.

But there are some tests and examples that can be build. Libosmium uses cmake:

```
>> mkdir build
>> cd build
>> cmake ..
>> make
```

The `osmium` Namespace

Everything in the Osmium library is in the `osmium` namespace or in sub-namespaces.
You'll likely encounter the **`osmium::io`** namespace for everything related to file input and output and the **`osmium::geom`** namespace for geometry-related functionality, but there are some more.

2. Dependencies:

1-CMake and Make : some tests and examples that can be build.
2-Utf8cpp : needed for the OPL output format.
3-Expat : needed for parsing OSM XML files.
4-zlib : needed for reading and writing OSM PBF files and for GZip support when reading and writing XML files.
5-bz2lib : needed for BZip2 support when reading and writing OSM XML files.
6-Boost Iterator is used for Tag filters, and for the Object Pointer Collection.
7-Extra dependencies for Extra Functions.

3. Input and Output:

For PBF input and output you need several libraries and threading enabled.
For version 2.3.0 and above you don't need much:

- Dependencies: Zlib
- Link with: libz, ws2_32 (Windows only), enable multithreading
- Classes: osmium::io::Reader , osmium::io::Writer
- Include files: osmium/io/any_input.hpp , osmium/io/any_output.hpp , osmium/io/pbf_input.hpp , osmium/io/pbf_output.hpp

Libosmium can read several different OSM file formats.
Whenever you want to use Osmium to access OSM files you need to include the right header files and link your program to the right libraries.

```
#include <osmium/io/pbf_input.hpp>
#include <osmium/io/pbf_output.hpp>
```

PBF Support:
To build with PBF support you have to compile with threads and need libz:
-pthread -lz

4. Detailed Usage:

a. Reading and Writing OSM Files with Osmium

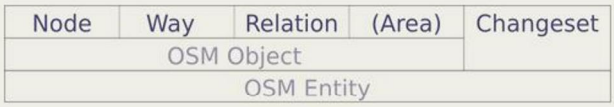
The osmium::io::File class

Before reading from or writing to an OSM file, you have to instantiate an object of class osmium::io::File. It encapsulates the file name as well as any information about the format of the file. In the simplest case the File class can derive the file format from the file name:

```
osmium::io::File input_file{"Le_Creusot.osm.pbf"} // PBF format
osmium::io::Reader reader{"Le_Creusot.osm.pbf"};
```

OSM Objects, and Entities:

OSM Object is illustrated in this photo, and each entity of this photo is described in details in the attached link. (Detailed Explanation of the Usage of the Namespaces, classes, functions and etc.)



b. Accessing Data in Buffers

Buffers containing OSM entities support the usual begin(), end(), cbegin(), and cend() functions:

You read the OSM entities from the file using the read() which returns a buffer with the data:

```
while (osmium::memory::Buffer buffer = reader.read()) {
   ...
}
```

```
osmium::io::Reader reader{"input.osm.pbf", osmium::osm_entity_bits::way};
```

You can set the following flags:

Flag	Description
`osmium::osm_entity_bits::nothing`	Do not ready any entities at all (useful if you are only interested in the file header)
`osmium::osm_entity_bits::node`	Read nodes
`osmium::osm_entity_bits::way`	Read ways
`osmium::osm_entity_bits::relation`	Read relations
`osmium::osm_entity_bits::changeset`	Read changesets
`osmium::osm_entity_bits::all`	Read all of the above

You can also “or” several flags together if needed.

```
osmium::memory::Buffer buffer = ...;
auto it = buffer.begin();
auto end = buffer.end();
for (; it != end; ++it) {
   std::cout << it->type() << "\n";
}
```

c. Handling Data of the Map:

If you process OSM data with libosmium to do something (e.g. convert to a different format, import into a database, build a routing graph), you will usually create one or more ***handlers***.

https://docs.osmcode.org/libosmium/v2.15.2/namespaceosmium.html

From <https://osmcode.org/libosmium/manual.html>

5. Conclusion:

I see that this Library will be the core library which we will use on our project (with necessary dependencies).

libosmpbfreader

Saturday, October 12, 2019 8:56 PM

1. A simple C++ library to read OpenStreetMap binary files.
2. Compared to https://github.com/joto/osmium libosmpbfreader is way less ambitious(Where Osmium provides a whole framework for parsing PBF files, it only provides a scaffold for reading those files.)
3. The main focus was to provide a library that can be used in 10 minutes.

From <https://github.com/CanalTP/libosmpbfreader>

Conclusion:

I don't think that we will use this Library, because it looks useless for our Project, but it maybe useful for smaller projects !