

# Style Mapnik pour OpenStreetMap sous QGIS



2016 – cartocite.fr  
CC-BY-SA Carto'Cité

---

## Présentation

Il est possible d'utiliser des données OpenStreetMap dans un SIG tel que QGIS. Un projet QGIS impliquant des données OpenStreetMap particulières nécessitera généralement de **convertir** les données à plusieurs reprises pour intégrer des **clés** au fur et à mesure de leur prise en compte. Pour réaliser cette conversion nous pouvons utiliser l'outil **ogr2ogr** accompagné d'un fichier de configuration **osmconf.ini**.

En résumé, des données OSM au format **.osm** ou **.pbk** vont être converties par l'outil **ogr2ogr** en données SIG au format **SpatiaLite**. Et la conversion est paramétrée grâce au fichier de configuration **osmconf.ini**.

Le paramétrage d'**ogr2ogr** via **osmconf.ini** permet principalement de configurer quelles **clés** seront converties en **champs**. Il permet d'autres configurations ; par exemple, il permet de définir quelles **clés** doivent être considérées comme un **polygone** si le **way** est fermé.

L'approche **itérative** (ajout des clés au fur et à mesure) est d'autant plus simple que le fichier de configuration peut être placé dans le dossier de travail avec les données OSM et du fichier **SpatiaLite**. Ainsi, l'intérêt d'utiliser un **fichier osmconf.ini à part** réside essentiellement dans la capacité à l'éditer facilement et d'avoir au final une configuration dédiée à un projet.

Pourquoi **SpatiaLite** ? C'est un format moderne performant et pérenne. Certains projets impliquent des données « mobiles » ; dans ce cas un fichier **SpatiaLite** contenant toutes les couches nécessaires s'avère pratique. Si **SpatiaLite** ne convient pas c'est probablement que PostGis est la solution et dans ce cas la démarche est proche.

**Cette méthode est nécessaire pour utiliser les styles QGIS proposés par Carto'Cité.**

Fonctionnement de **ogr2ogr** : <http://www.gdal.org/ogr2ogr.html>

Paramétrer **osmconf.ini** : [http://www.gdal.org/drv\\_osm.html](http://www.gdal.org/drv_osm.html)

---

## Objectifs

- Disposer de données OpenStreetMap dans QGIS (ou un autre SIG).
- Choisir les clé qui seront transformées en champs.
- Utiliser les styles QGIS proposés par Carto'Cit .

---

## Pr requis

- Comprendre le fonctionnement g n ral d'OpenStreetMap et plus particuli rement le stockage des donn es attributaires cl /valeur.
- Savoir r cup rer des donn es OpenStreetMap au format *.osm* ou *.pbf*
- QGIS  $\geq 2.14$
- OGR/GDAL (  priori install  avec QGIS syst matiquement)
-  diteur de texte (notepad++, etc.)

---

## Ligne de commande

### Le terminal

*ogr2ogr* est un outil en ligne de commande, il s'utilise donc dans un *terminal*. On parle de **terminal**, de **shell** ou de **console** g n ralement pour d signer l'outil qui permet d'utiliser des programmes avec du texte appel  ligne de commande.

Si vous n' tes pas habitu s   utiliser un terminal ; il ne faut pas reculer devant cette exp rience, les op rations sont relativement simples.

### Windows/Linux

Sous Linux, il suffit de lancer un terminal pour pouvoir utiliser *ogr2ogr* (lorsqu'il est install ).

Sous Windows il existe diff rents cas de figure, aussi pour avoir un terminal fonctionnel, il faut d'abord se reporter   la partie d di    Windows : Pr paration de *ogr2ogr* sous Windows.

---

## Op rations

### Pr paration

Nous avons besoin d'un fichier de donn es OSM et d'un fichier de configuration *osmconf.ini*. Nous allons placer ces fichiers dans un dossier que nous nommerons « *Projet osm* » ; il faut donc cr er ce dossier, sur le bureau par exemple.

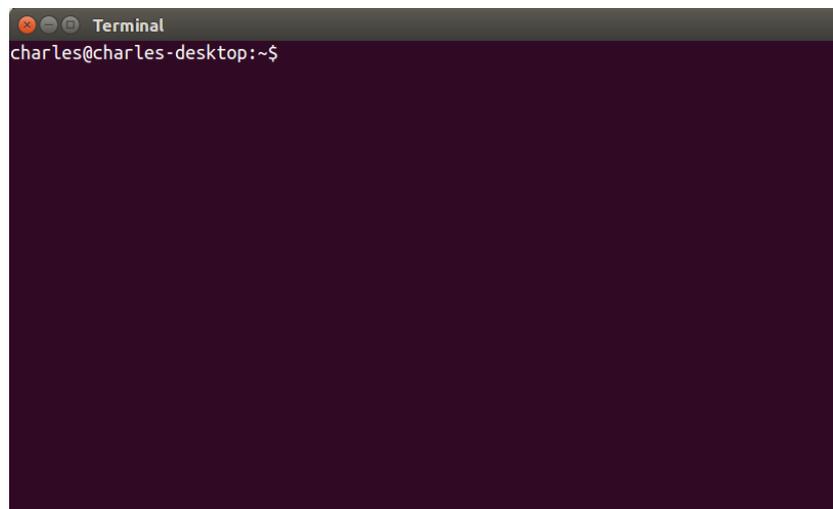
Prenons les données OSM de la ville de Tours au format *.pbf* pour avoir un fichier relativement léger : [Mapzen – Metro Extract](#). Renommer le fichier en *tours.pbf*

Pour le fichier de configuration nous allons utiliser celui qui est adapté pour les styles « Mapnik » de Carto'Cit . Penser   faire *Fichier > Enregistrer sous* ou un simple CTRL+S pour sauvegarder le fichier *osmconf.ini* dans le dossier *Projet osm* : [https://raw.githubusercontent.com/charlesmillet/qgis\\_osm\\_styles/master/osm\\_mapnik/osmconf.ini](https://raw.githubusercontent.com/charlesmillet/qgis_osm_styles/master/osm_mapnik/osmconf.ini)

  titre d'information, le fichier *osmconf.ini* par d faut se trouve ici : <https://svn.osgeo.org/gdal/trunk/gdal/data/osmconf.ini>

## Se placer dans le dossier de travail

Ouvrir un terminal.



Dans le terminal il faut tout d'abord se placer dans notre dossier de travail, celui o  se trouvent les fichiers *nantes.pbf* et *osmconf.ini*. La commande **cd** (*current directory*) fonctionne sous Windows et Linux :

Exemple sous Windows :

```
cd 'c:/users/charles/Bureau/Projet osm'
```

Exemple sous Linux :

```
cd '/home/charles/Desktop/Projet osm'
```

Il faut penser   utiliser des guillemets (aussi appel es double *cote*) ou des apostrophes (simple *cote*) comme dans les exemples si l'adresse du dossier contient des **espaces**.

Pensez  galement   utiliser la touche tabulation en cours de saisie pour profiter de l'auto compl tion.

```
Terminal
charles@charles-desktop:~$ cd '/home/charles/Desktop/Projet osm/'
charles@charles-desktop:~/Desktop/Projet osm$
```

## Conversion

Dans le terminal, une fois placé dans le bon dossier, il faut rédiger la ligne de commande *ogr2ogr*. La ligne de commande de base est :

```
ogr2ogr -f "SQLite" -dsco SPATIALITE=YES
tours.sqlite tours.osm
```

Cette commande ne prend pas en compte le fichier de configuration spécial que nous avons téléchargé. Pour le prendre en compte il faut utiliser l'option **--config OSM\_CONFIG\_FILE** et pointer vers le fichier **osmconf.ini**. La commande se présente alors comme ceci :

```
ogr2ogr -overwrite -f "SQLite" -dsco SPATIALITE=YES
tours.sqlite tours.osm --config
OSM_CONFIG_FILE osmconf.ini
```

En cas d'erreur il faut essayer d'ajouter le paramètre **--config OSM\_USE\_CUSTOM\_INDEXING NO**

```
ogr2ogr -overwrite -f "SQLite" -dsco SPATIALITE=YES
tours.sqlite tours.osm --config
OSM_CONFIG_FILE osmconf.ini --config
OSM_USE_CUSTOM_INDEXING NO
```

Il est possible d'accélérer la conversion sur un PC qui ne date pas trop avec le paramètre **--config OGR\_SQLITE\_SYNCHRONOUS OFF**

```
ogr2ogr -overwrite -f "SQLite" -dsco SPATIALITE=YES
tours.sqlite tours.pbf --config
OSM_CONFIG_FILE osmconf.ini --config
OSM_USE_CUSTOM_INDEXING NO --config
OGR_SQLITE_SYNCHRONOUS OFF
```

```
Terminal
charles@charles-desktop:~$ cd '/home/charles/Desktop/Projet osm/'
charles@charles-desktop:~/Desktop/Projet osm$ ogr2ogr -overwrite -f "SQLite" -ds
co SPATIALITE=YES tours.sqlite tours.pbf --config OSM_CONFIG_FILE osmconf.ini --
config OSM_USE_CUSTOM_INDEXING NO --config OGR_SQLITE_SYNCHRONOUS OFF
Warning 1: organizePolygons() received an unexpected geometry.
Either a polygon with interior rings, or a polygon with less than 4 points,
or a non-Polygon geometry. Return arguments as a collection.
charles@charles-desktop:~/Desktop/Projet osm$
```

Tout s'est bien passé, il n'y a qu'un polygone qui a eu un problème de conversion mais rien d'alarmant.

---

## Appliquer les styles

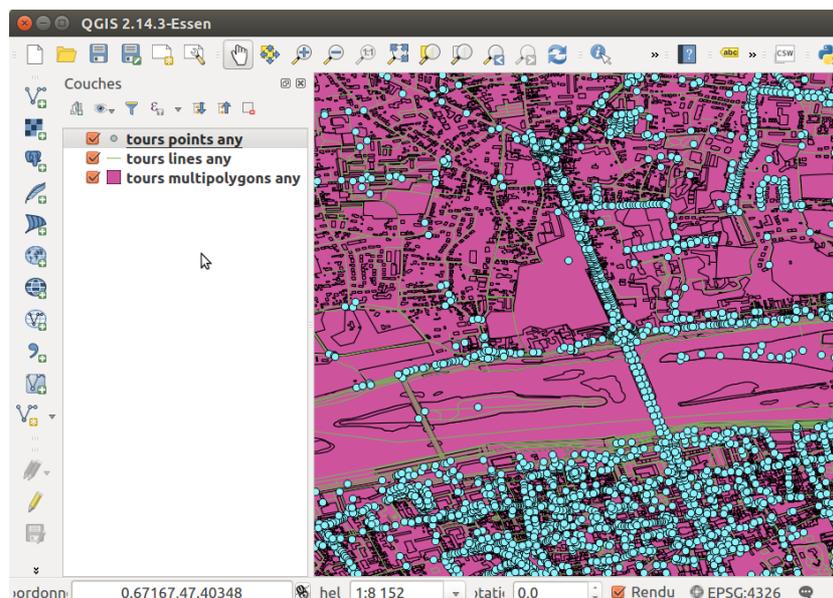
### Ajouter les données dans QGIS

Maintenant que nous disposons d'un fichiers *SpatiaLite* avec les données OpenStreetMap, nous allons l'ajouter à QGIS. Pour l'ajouter nous pouvons faire un simple glisser/déposer du fichier ou passer par le menu ou la barre d'outil de QGIS pour l'ajout de couches vectorielles.

Il faut ajouter les trois couches qui correspondent aux *points*, *polylignes* et *multi-polygones* :

Identifiant de la couche	Nom de la couche	Nombre d'entités	Type de géométrie
0	points	35799	Point
1	lines	26110	LineString
2	multilinestrings	298	MultiLineString
3	multipolygons	153599	MultiPolygon
4	other_relations	2005	GeometryCollection

On observe la multitude de données OSM. On peut également en profiter pour activer la re-projection à la volée en Lambert 93 pour éviter d'avoir la déformation due au WGS 84.



## Récupérer les styles

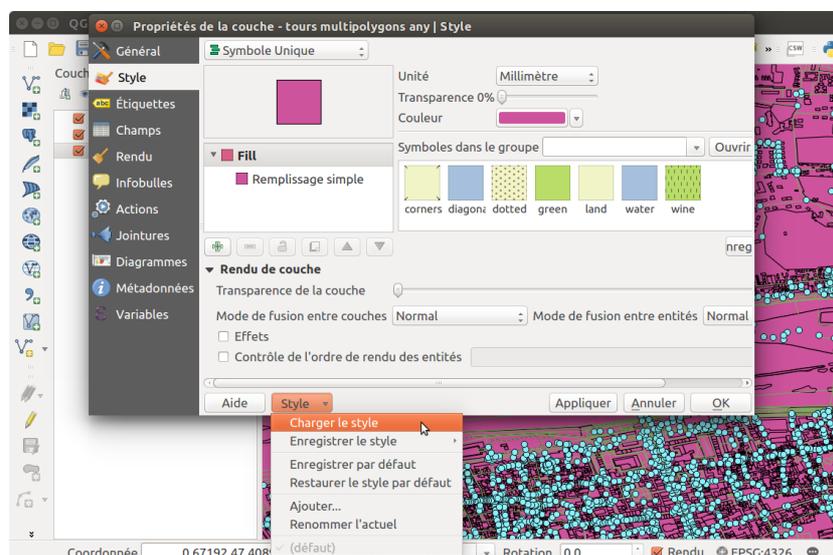
Les styles « Mapnik » sont en cours de développement et sont loin d'être complets. Par contre, ils représentent une bonne base pour développer ses propres styles. Pour les télécharger, depuis le [dépôt github](#), il faut récupérer les 3 fichiers de style :

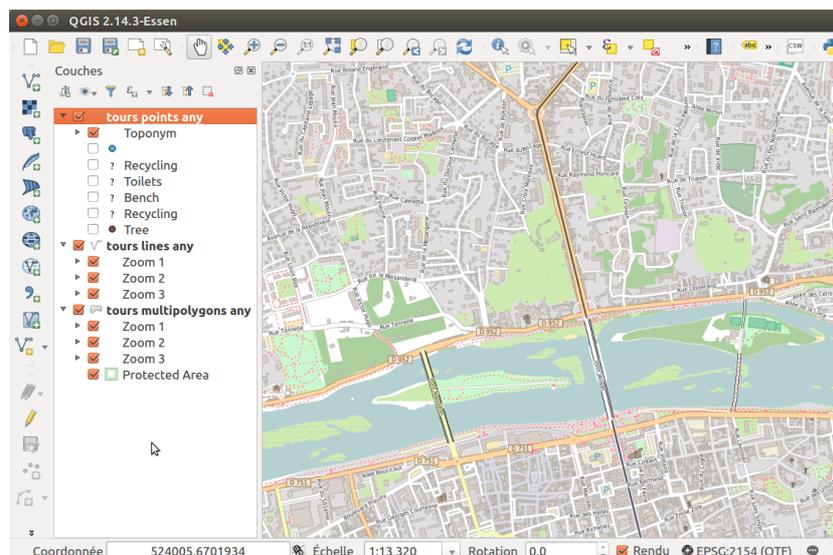
- [osm\\_mapnik\\_line.qml](#)
- [osm\\_mapnik\\_point.qml](#)
- [osm\\_mapnik\\_polygon.qml](#)

Pensez à cliquer sur le bouton **raw** pour afficher le contenu du fichier et pouvoir ensuite télécharger le fichier de style (CTRL+S).

## Appliquer les styles

Pour appliquer les styles, il faut se rendre dans les *Propriétés* de chaque couche une par une, aller dans l'onglet *Style*, puis sur le bouton *Style > Charger le style* et sélectionner le styles téléchargé :





À la fin des opérations on se retrouve avec ces fichiers :

- osmconf.ini
- osm\_mapnik\_line.qml
- osm\_mapnik\_point.qml
- osm\_mapnik\_polygon.qml
- tours.pbf
- tours.sqlite

---

## Préparation de ogr2ogr sous Windows

### Cas numéro 1

Vous avez installé QGIS via **OSGeo4W** ou vous avez installé **FWTools**. Dans ce cas vous pouvez utiliser le *shell* OSGeo4W ou le *shell* FWTools directement.

Si vous ne trouvez rien dans le menu *Démarrer*, cherchez directement sur le disque dur à l'aide de l'outil de recherche. À titre d'exemple, OSGeo4W s'installe généralement dans le dossier C:\OSGeo4W64 et le terminal se lance avec le fichier *OSGeo4W.bat*.

### Cas numéro 2

Vous avez installé QGIS à partir de l'exécutable, dans ce cas nous allons utiliser le **terminal de Windows**. Pour pouvoir utiliser confortablement le terminal nous allons **référencer le chemin vers les outils**. Et pour référencer le chemin vers les outils, nous allons justement utiliser le terminal. Donc la première chose à savoir faire est de lancer un terminal sous Windows.

Le terminal Windows peut se lancer avec l'exécutable **cmd.exe**. Il faut donc lancer **cmd.exe**. Il y a différents moyens de le faire sous Windows, par exemple sous Windows 7, on peut appuyer sur la touche Windows du clavier ce qui va lancer le *lanceur (launcher)* de Windows, puis taper **cmd.exe**. On peut aussi utiliser la fonction exécuter : touches **WIN+R** puis taper **cmd.exe**.

Il faut maintenant trouver où est installé l'outil *ogr2ogr.exe*. Pour cela on peut faire une recherche sous Windows. Trouvez l'emplacement de *ogr2ogr.exe* et copier/coller l'adresse complète du **dossier** où se trouve l'exécutable *ogr2ogr.exe*. Par exemple, il est généralement sous **c:/OSGeo4W/bin** lorsque l'on est passé par l'installateur OSGeo4W ; si on a installé QGIS avec l'installateur classique on le trouvera dans le dossier **C:/Program Files/QGIS Essen/bin** – Il faut changer **QGIS Essen** par la version de QGIS que vous avez installée.

Nous allons définir des **variables d'environnement** ; en quelque sorte nous allons dire à Windows où se trouve le logiciel qui nous intéresse.

La commande **setx** permet de définir de manière permanente une variable à la différence de la commande **set** qui ne la définit que temporairement – ce qui peut être utile lorsque l'on ne possède pas les droits d'administration :

```
setx PATH "%PATH%;C:/Program Files/QGIS Essen/bin"  
setx GDAL_DATA "C:/Program Files/QGIS  
Essen/share/gdal"
```

On peut ensuite vérifier que Windows sait maintenant bien où se trouve *ogr2ogr* en exécutant la commande suivante :

```
ogr2ogr --version
```