Importing Aires database as POIs into OsmAnd

Richard R. Goodwin (Morphology) Email: richard@morphology.co.uk

Disclaimer

This guide it simply that – a ***guide*** – it is not intended to be a definitive reference work, and no claims are made as to its accuracy or fitness for any purpose. Some of the concepts described herein can seem complex if you have a limited understanding of computers. Whilst I have tried to describe the steps required with as little computer jargon as possible it is inevitable that techno-bollox will creep in from time to time.

If you are unable to make the processes described in this guide work for you then I'm afraid that there will be little that I can do to help. It will never be possible for me to create a one-size-fits-all document that covers every feasible combination of devices and software and it would be wrong for me to try.

Although Osmand+ runs on the Android operating system on tablets and smartphones, most of the data manipulation we need to carry out in order to create the Points Of Interest files in a format that Osmand+ can understand are carried out using software that runs on a PC running Windows. There may be alternatives or equivalents that work under IOS on Macs or various flavours of Linux, but you're on your own.

Background

I put this guide together as an aid for like-minded motorhome owners, whose aim is to get the best out of the Osmand+ Android Application.

Osmand (Open Street Map Automated Navigation Directions) is an application that enables users of Android tablets and smart-phones to download Open Street Map files and store them on the device, the obvious benefit of which is that whilst travelling abroad it is no longer necessary to find a data connection or pay mobile data roaming charges in order to access maps or plan routes. New maps and updates to existing maps are also free of charge, which makes a pleasant change from updating traditional Sat-Nav devices from the likes of Tom Tom and Garmin.

There are two versions of Osmand. The free version is limited to downloading 16 offline maps whereas the paid-for version (known as Osmand+ and currently costing £4.99) allows unlimited map downloads. As far as I am concerned, the two are interchangeable. The free version may lag a revision or two behind the paid-for version but, other than the map download limits, I believe them to be identical. For that reason, I'll simply refer to the app as Osmand.

Open Street Map (OSM) is a free map of the whole world built by a community of contributors (as of January 2013 there were in excess of 1 million registered users). It was started by Steve Coast in the UK in 2004 initially as a way of breaking the hold that government-funded mapping companies such as Ordnance Survey have over their proprietary and copyrighted mapping data.

OSM has since expanded to cover the whole world and has subsumed other open-source mapping datasets to become the pre-eminent Open Source Map. Its coverage of Western Europe and North America is easily on a par with, say, Google Maps, though its coverage of less-populated corners of the planet can sometimes be sketchy (though feel free to update it where you find omissions or inaccuracies!).

Osmand is a work in progress. What started as a simple app to render OSM data on Android devices

has been driven by a community of developers and users who are adding navigation and routing algorithms, Points of Interest, topographical features, hill shading and so on.

The point where Osmand+ started to become of interest to the motorhoming community was when the developers added the ability to import waypoints and Points of Interest (POI) files.

Notes about OSMAnd Map Storage

Open Street Map files in .OBJ format are ***huge*** (England is around 800Mb, for example) so it's a good idea to give some thought to where you are going to store these on your Android device, especially if you intend to carry all of Western Europe around with you.

I have always purchased phones & tablets with 16Gb of internal storage but ***with the ability to add a micro SD card***. My reasoning is simple: The price difference between a device with 16Gb of on-board storage and one with, say 32Gb can often be tens if not a hundred pounds, whereas a 16Gb Micro SD Card will cost around £5 and 32Gb ones are now around £12.

Some versions of the Android operating system enable applications themselves to be moved to an SD Card, though this is not universally possible – it depends on the device manufacturer building in support for it. For example, first revisions of Android for the Samsung Galaxy Tab 3 didn't include the 'move application to SD Card' feature, though later revisions did.

Fortunately, Osmand is one of those well-behaved apps which lets you specify where it should store its data. So, for example, I have the App installed on my Phones internal memory, but the map data stored on the SD card.

To do this, start Osmand and tap on ***Settings*** then ***General***. Scroll down until you see ***Storage Directory***. Mine is set to: */mnt/sdcard/external_sd* which is the 'mount' point for the external SD card in my particular device.

How this is described may very between devices and versions of Android. To find out what yours is called, I recommend downloading the free 'My Files' app from the Google Play Store. This is a handy directory and file explorer (similar to windows explorer).

Scroll down the list of directories and, if your external SD card has been mounted correctly and recognised by the Android operating system, you should recognise one of the directories as being your SD Card. Tap on it, which will open the 'root' directory of the SD card and, at the top of the display, the My Files app will show the current path.

Mine shows ***/sdcard/external_sd*** but to get the full path you need to prepend ***/mnt/*** to it, which is the default Android storage 'mount point'.

Importing POI files to Osmand - General

Osmand will index additional POI files in made available to it in .OBF format. The trick is to create the .OBF files and drop them into the 'root' directory of Osmand's storage directory before starting the program.

Note: It is sometimes necessary to force the application to halt [go to Settings->Apps->Osmand+ and tap FORCE STOP] before re-starting Osmand in order to force the application to re-index any .OBF files.

Prerequisites

- In order to convert between various different type of POI file, you will need to download and install GPSBabel. Available from here: http://www.gpsbabel.org/ this will let you take POI files in various different formats, and convert them into Open Street Map format (.OSM). ***Note:*** this is a Windows application, so needs to be installed on a PC.
- In order to convert .OSM files into .OBF you will need **OsmAnd Map Creator**. This is a Java console application so unfortunately, you will need to install Java in order to be able to run it (see below). As far as I am aware, this is the only way of creating .OBF files from .OSM files. If you know different, please let me know.

Examples: Converting CampingCar-Infos POI file for use in OsmAnd

First you need to download the ASCII (ie plain text) version of the POI files for the particular country you are interested in.

From http://www.campingcar-infos.com/index1.htm click on "Cliquez pour entrer" *(Click to Enter)* then "Telechargement GPS" *(Download GPS)*

Click on ASCII / Tom Tom (they both go to the same place)

Click on the **Ascii** icon next to the country you want Eg. France – this will download a .ZIP file called Aires_CCinfos_FRANCE_asc.zip unzip this file into a suitable place on your PC – you will end up with several different files:

AA_CCI.bmp
AA_CCI.asc
AC_CCI.bmp
AC_CCI.asc
...etc

the .BMP files are the bitmap Icons used by the satnav to indicate the type of POI. Unfortunately OsmAnd cannot currently display these, so they are of no use to us.

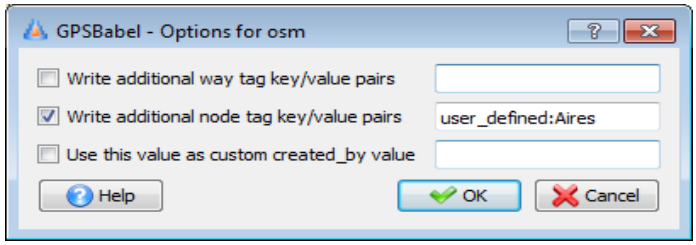
The asc files contain the co-ordinates and descriptions of the Aires themselves. These are split up into a number of different types:

AA_CCI – Aires on Motorways
AC_CCI – Campsite that accepts Motorhomes
ACF_CCI – Aire on a Farm
ACS_CCI – Campsite which allows use of service point
AP_CCI – Parking tolerated during the day (no service point)
APCC_CCI – Dedicated Motorhome parking (no service point)
APN_CCI – Parking Tolerated Day & Night
AS_CCI – Service point only
ASN_CCI – Service point with overnight parking
ATOTALES_CCI – All the above in one file

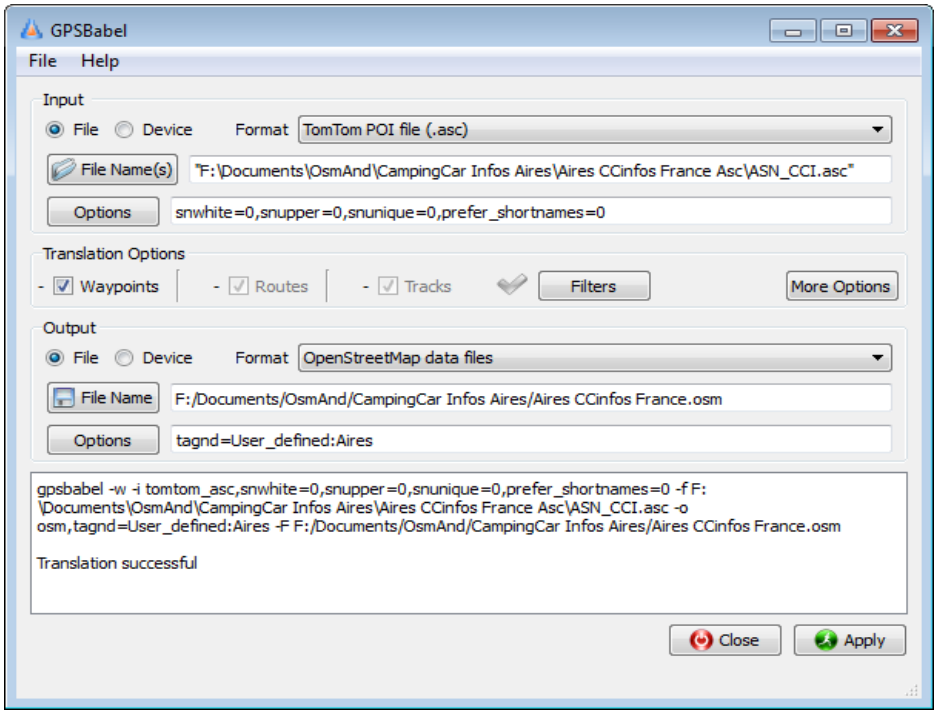
For overnight stops with a service point, the ASN_CCI.asc file is probably the most useful.

Convert the .asc file into a .osm (Open Street Map) file, using GPS Babel

- Launch GPSBabel
- Select Tom Tom POI file (.asc) as the input file format
- Click on the [File Name(s)] button to navigate to the appropriate source file (ASN_CCI.asc) we downloaded above.
- Select output format: OpenStreetMap data files
- click the [File Name] button and give the output file a suitable name (Aires CCInfos France.osm for example)
- Click the [Options] button and enter “user_defined:Aires” as the Additional node Tag/Key value pair:



- Click on the [Apply] button to perform the translation:

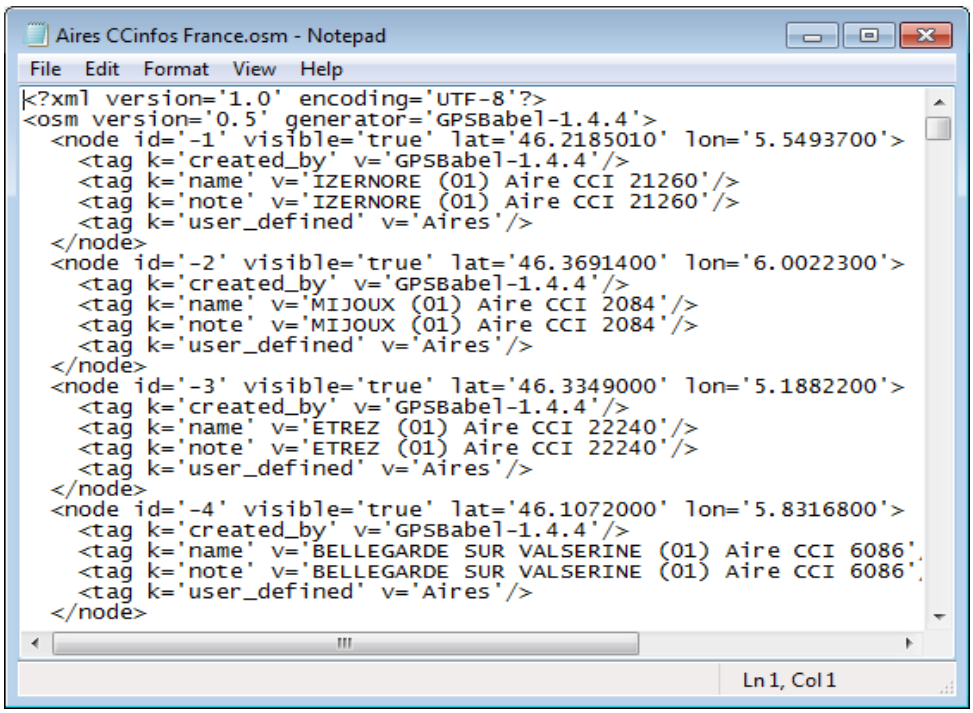


If all is well, you'll get a 'Translation Sucessful' Message.

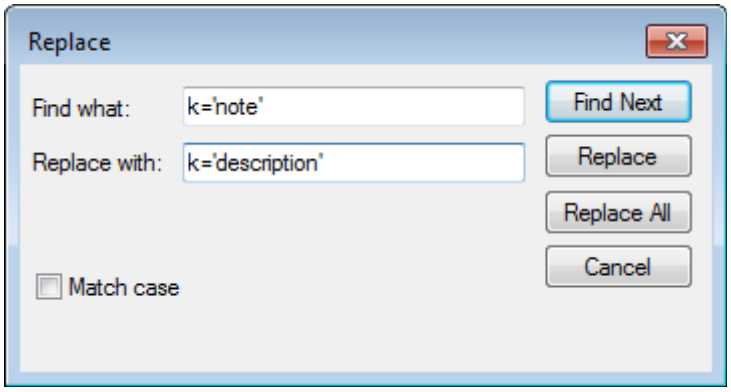
Unfortunately, GPSBabel creates OSM files containing nodes with a 'note' tag, whereas Osmand looks for nodes with a 'Description' tag.

So, we need to perform a search and replace on the file.

Open it in Notepad, and it will look something like this:



- Click on **Edit->Replace**
- Find What: k='note'
- Replace With: k='description'
- Click on [Replace All]



It's pretty pointless doing this with the CampingCar Infos files, as the 'note' tag is only ever the same as the 'name' tag, so this step is more important with POI files from other sources.

The next stage is to convert this .OSM file into .OBF format using OsmMapCreator

OsmAnd Map Creator

This is a Java Console app, which can be downloaded from the code.google source pages.

Unzip it all into a directory such as **c:\program files\OsmMapCreator**

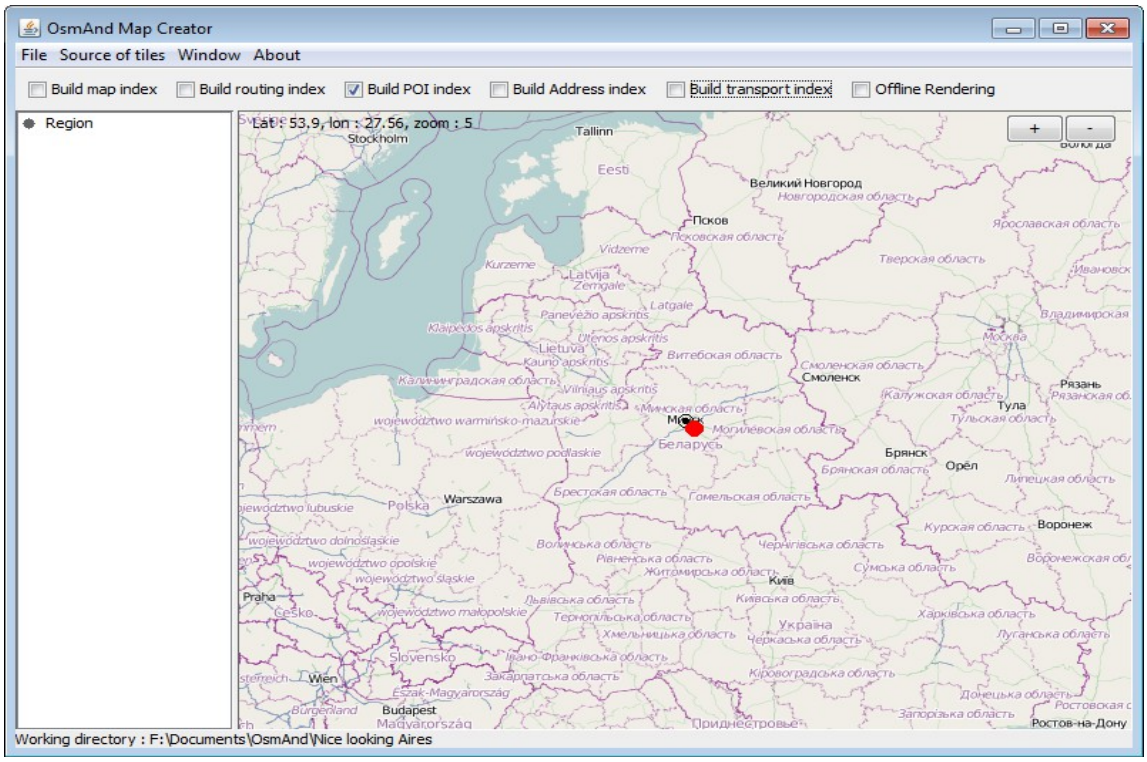
Start it by running **OsmMapCreator.bat** (which you will find in the root of the directory you created) If it complains that it cannot find **javaw.exe** this means that either you don't have the Java Runtime Environment installed, or else windows cannot find it in the %PATH% environment variable.

If the latter, then go to Computer->Properties->Advanced and Environment Variables, and stick the path to the JRE on to the end of the PATH environment variable.

Mine was in **c:\Program** Files (x86)\Java\jre7\bin

If you don't have Java installed, it can be downloaded from **www.java.com**

When you run it, you'll see a screen similar to this:



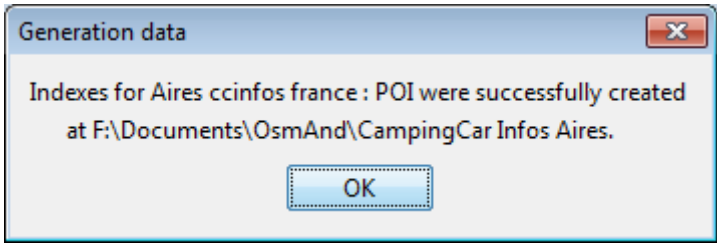
Make sure you have the latest version. Mine is currently Version 1.1.0 Build #2582M.

First click on **File->Specify Working Directory** this lets OsmMapCreator know where you would like it to drop the .OBF file once it has created it.

Untick everything on the top line ***except*** Build POI Index

Click on **File->Create .OBF from osm file** navigate to where you stored your OSM file and click **Open**

Hopefully, once it has done its stuff you'll get a message to say that it has created the indexes successfully.



One point to bear in mind is that OsmMapCreator is case sensitive, so it will only look for files with the extension .osm **NOT** .OSM !!

The resulting file should be dropped into the Working Directory you specified above. You need to find this file and copy it into the root directory of the Osmand installation.

Make sure the Osmand application is completely stopped – it is worth going into Settings-Apps finding the Osmand app in the list and clicking on “Force Stop”

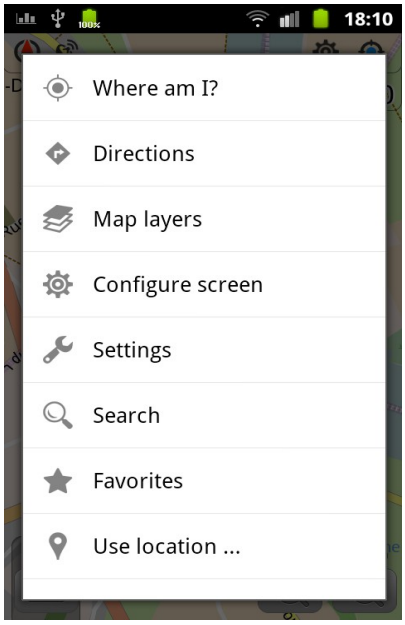
Re-start Osmand, which will force it to find and index any .OBJ files in its root directory

Finding and displaying POIs on Osmand+

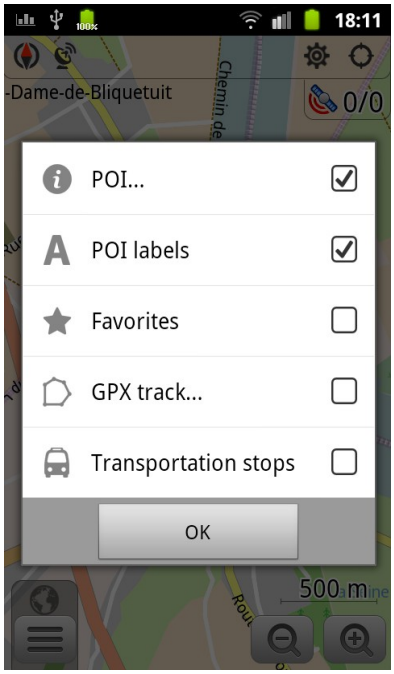
The screenshots below were taken from a Samsung Galaxy W phone, but the principle and menus should be reasonably similar on other Android tablets and phones.

To display your User Defined POIs on Osmand

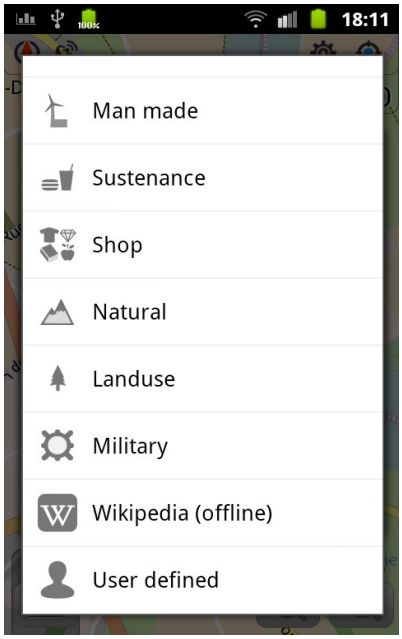
- from within the application tap 'Menu', which will bring up a screen similar to this:



- Tap on 'Map Layers', which will bring up a screen similar to this:



- Make sure POI Labels is checked
- If POI is already Checked, un-check it, then re-check it
- This will bring up a list of 'in-built' POIs. If you scroll to the bottom of the list you'll see User Defined



- Tap on this, and your POIs will then be displayed on the Osmand map once you zoom in above a certain magnification. The screenshot below shows some aires in the Calais area

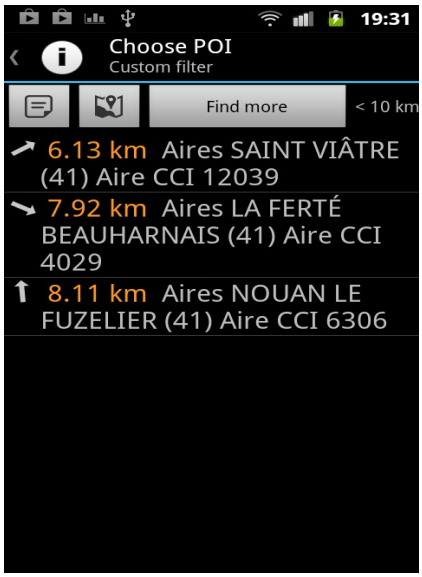
If you hold a finger on one, it will bring up a sub-menu where you can choose to display the description which, for a CampingCar Infos aire looks something like this:

Not very exciting, but other POI files may contain more interesting 'description' fields.

Finding the nearest Aire

This is where POI files on Osmand+ really start to come into their own. Suppose for a moment you are travelling down the A71 south of Orleans, and want to find the closest Aire...

- Tap on Menu then 'Map Layers'
- Un-check POI if it is already checked, tap on Menu then 'Map Layers' again
- Tap on 'Custom Filter'
- Scroll to the bottom of the list and select 'User Defined' then tap the [Select All] button
- Click on 'Filter' and it will show you any User Defined POIs within 1km of your current position
- If none are displayed, tap 'FIND MORE' and it will start extending the search to 2km, 3km, 5km, 10km etc until it starts displaying a list of POIs along with their distance and direction



Tap on the one you are interested in (Eg. Nouan Le Fuzelier) and a Pop-up menu will be displayed giving you a choice of

- Directions
- Set as Destination
- Show on Map

Notes about .OSM files

- It is possible to include Web links in POIs, by including the following tag:

  <tag k='website' v="www.google.com"/>

- This could be useful when importing POI sets from sources that include links to photos etc.