Tricks
with
mkgmap styles
2

ideas to enhance your maps

(c) N Willink 2012/14

latest edition : 24/10/2014

Introduction 3
How to use ‘continue’.....3
How to display names of 'unknown' tags 4
How to always render buildings 5
How change the fontsize of your place names 6
How to give inner polygons an outline.....6
How to create Country specific icons (bus stops) 7
BOUNDARIES 8
How to create Country specific highways.....9
How to create Country specific polygons.....9

Introduction

Any one not familiar with Perl , may find style sheets quite daunting and puzzling.

Initially it may be helpful to use the default sheets which come with mkgmap.

The following links may be also be useful:

http://wiki.openstreetmap.org/wiki/Mkgmap/help/style_rules

It is not our intention explain why snippets of code are working or not. However, we feel that after a while you may wish to add some extra features which are not included in the default style sheets. After all, you ultimately want to design your own maps!

This is an on going document . All suggestions are welcome (osm at pinns . co . uk).

How to use 'continue'

Many are confused by the use of continue . Consider these 2 examples

example1:

```
highway = primary & lit = yes [0xYYYYY resolution 24
continue] highway = primary [0xXXXXX resolution 24]
```

example2:

```
highway = primary [0xXXXXX resolution 24]
highway = primary & lit = yes [0xYYYYY resolution 24 continue]
```

Continue creates more than one line for the same highway if the same condition is met. If highway=primary then create a line.

In our first example a highway with lit=yes is created ***followed by*** a primary highway.

In our second example, only the highway=primary gets plotted . NOT the highway=primary & lit=yes.

If there is no continue , mkgmap will stop parsing the lines file, and move onto another file

How to display names of 'unknown' tags

If you want to show say all amenities not found in your style sheet you use

amenity=* [0xXXX resolution 24] # where XXX is a hexadecimal number

Make sure you place this at the end of your amenities list.
However, this will not tell you what the amenity was .
When you hover your mouse over it in Basecamp etc you will see 'amenity=*'
This is pretty useless as we already know that.
To make it show the type of amenity , you need to add the following code before the [.

{set name='${amenity}'} ' this means use the amenity tag as a name

ie `amenity=* {set name='${amenity}'} [ 0xXXX  resolution 24]`

The same applied to other tags,ie

```
man_made=* {set name='${man_made}'} [ 0xXXX  resolution 24]

cuisine=* {set name='${cuisine}'} [ 0xXXX  resolution 24]
```

How to always render buildings

Even if your buildings have the highest draworder in your typ file, they still may not appear!

```
building = yes [0xXXXXX resolution 24]
```

(This means that buildings only get plotted at zoom level
24. It does not mean than buildings will be on top !
Nor does it mean that buildings will always be plotted)

Here is an example where buildings may not be plotted !

Supposing in OSM a school building was plotted as

Amenity=school
Building=yes

In this example , the building will ***not*** be plotted:

```
amenity=school [0x16010 resolution 24 ]

building=yes [0x1081F resolution 24 ]
```

In the following example , the building will be plotted:

```
amenity=school [0x16010 resolution 24 continue]

building=yes [0x1081F resolution 24 ]
```

However, its not recommended to put continue after amenity, as it could plot other amenities as well , ie if you have 'amenity = *' further down .

Solution

```
building=yes [0x1081F resolution 24 CONTINUE]

amenity=school [0x16010 resolution 24 ]
```

We recommend

```
building=yes [0x1081F resolution 24 continue]
```

to be inserted almost as the first line of your polygon file.

How change the fontsize of your place names

Before TYP files were introduced by Garmin, the only way to change the fontsize of your towns,cities etc was by selecting a particular type number:

	Fontsize	Font Style
0 - 500	Big	**BOLD**
600 - A00	Big	Normal
B00 - D00	Small	Normal
E00 - 1100	Big	Normal

Unfortunately, on Basecamp/Mapsource & many GPS devices, these settings are hard wired and cannot be overwritten by the font settings of your TYP file.

From the above you can see that if you want to have your own town in bold & large you should select , say 0x500.

How to give inner polygons an outline

In relations:

```
type=multipolygon  {apply { add multipol=yes;  }}
```

Whenever it meets a multipolygon relation, it sets multipol to yes

In lines:

```
multipol=yes  [0x11002  resolution 24 continue ]
```

You must add continue as else the polygon won't be plotted, just any inner 'holes'

Remember:

Lines are plotted BEFORE polygons

How to create Country specific icons (bus stops)

If you live on or near the border of another country you may wish mkgmap to create icons specific to the various countries.

For this you need the option : mkgmap:country

Example

```
highway=bus_stop & mkgmap:country =LUX [0x10209 resolution 24]
highway=bus_stop & mkgmap:country =FRA [0x1020A resolution 24]
highway=bus_stop & mkgmap:country =DEU [0x10208 resolution 24]
highway=bus_stop [0x10207 resolution 24] # your country
```

However, this won't work unless

a) you have included a way of determining the boundaries of various countries
b) you have specified the countries

using the **--bounds** command - see below

BOUNDARIES

These can be created yourself or downloaded from an excellent site:

www.navmaps.eu Look for bounds.zip – it's quite a meaty file.

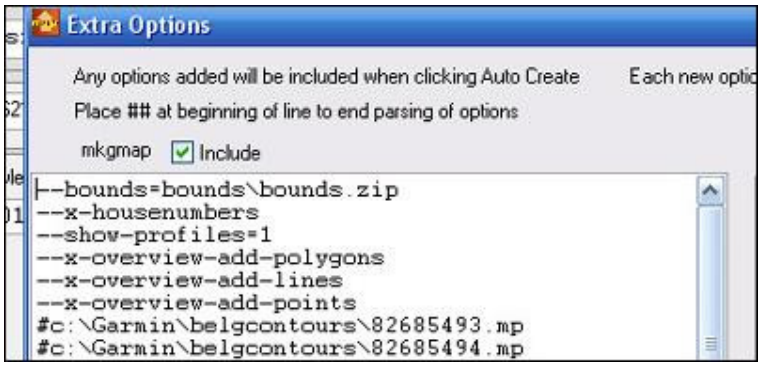
This needs to be placed in a subfile of where mkgmap.jar is

Using mapuploader2 create a folder called 'bounds' where you find mapuploader2.exe and place the zip file inside this folder – DO NOT UNZIP

Add the following command to your mkgmap list:

--bounds=bounds\bounds.zip

If you are using mapuploader go Preference Extra Options



This will examine your osm map and automatically allocate the correct countries/regions etc.

Note : This has nothing to do with the Countries added by individual OSM contributors, which often do not conform to the required shortcut , ie GB in staed if GBR or Luxembourg instead of LUX - see below,

The boundary file does not have to be unzipped - mkgmap will automatically find the correct data. You may find that in certain circumstances the boundaries are not quite correct.

How to create Country specific highways

```
highway=motorway  & mkgmap:country =LUX [0x10209 resolution 24]
highway=motorway  & mkgmap:country =DEU [0x1020A resolution 24]
```

How to create Country specific polygons

```
building=yes  & mkgmap:country = NLD [0x10009 resolution 24]
building=yes  & mkgmap:country = DEU [0x1000A resolution 24]
```

This gives the German and Dutch buildings a different colour.

COUNTRIES

These are specified using country code below

The following code needs to be added to the points file if you are using points.

If you are using then mkgmap:country option to differentiate between the motorways of different countries, say , it needs to be added to the lines file.

Again, if you want your buildings to have a different colour to another country's buildings , it has to be added to the polygons file.

Because of this . it's much easier to use the 'include' option and have the file in a subfile: I

have the 'admin code' in a file called address located in a subfolder called 'inc'

Add the following to your lines,points & polygons file

```
include 'inc/address';
```

Save the following code as 'address' and place it in the 'inc' folder

```
# first set the country code
```

```
mkgmap:country!=* & mkgmap:admin_level2=* {
set mkgmap:country='${mkgmap:admin_level2}' }
mkgmap:country!=* & addr:country=* { set mkgmap:country='${addr:country}' }
mkgmap:country!=* & is_in:country=* { set mkgmap:country='${is_in:country}' }

mkgmap:country=LUX & mkgmap:region!=* & mkgmap:admin_level6=* { set
mkgmap:region='${mkgmap:admin_level6}' }
mkgmap:country=BEL & mkgmap:region!=* & mkgmap:admin_level6=* { set
mkgmap:region='${mkgmap:admin_level6}' }
mkgmap:region!=* & mkgmap:admin_level4=* { set
mkgmap:region='${mkgmap:admin_level4}' }
mkgmap:region!=* & mkgmap:admin_level5=* { set
mkgmap:region='${mkgmap:admin_level5}' }
mkgmap:region!=* & mkgmap:admin_level6=* { set
mkgmap:region='${mkgmap:admin_level6}' }
mkgmap:region!=* & is_in:county=* { set mkgmap:region='${is_in:county}' }

# Germany = DEU cities
mkgmap:country=DEU & mkgmap:city!=* & mkgmap:admin_level4=Hamburg {set
mkgmap:city='${mkgmap:admin_level4}' }
mkgmap:country=DEU & mkgmap:city!=* & mkgmap:admin_level4=Berlin {set
mkgmap:city='${mkgmap:admin_level4}' }
mkgmap:country=DEU & mkgmap:region!=* & mkgmap:admin_level4=* { set
mkgmap:region='${mkgmap:admin_level4}' }
mkgmap:country=DEU & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8|subst:Gemeinde |subst:Stadt}' }
mkgmap:country=DEU & mkgmap:city!=* & mkgmap:admin_level7=* { set
mkgmap:city='${mkgmap:admin_level7|subst:Gemeinde |subst:Stadt}' }
mkgmap:country=DEU & mkgmap:city!=* & mkgmap:admin_level6=* { set
mkgmap:city='${mkgmap:admin_level6|subst:Gemeinde |subst:Stadt}' }
mkgmap:country=DEU & mkgmap:city!=* & mkgmap:admin_level9=* { set
mkgmap:city='${mkgmap:admin_level9|subst:Gemeinde |subst:Stadt}' }
mkgmap:country=DEU & mkgmap:city!=* & mkgmap:admin_level10=* { set
mkgmap:city='${mkgmap:admin_level10|subst:Gemeinde |subst:Stadt}' }

# Austria = AUT
mkgmap:country=AUT & mkgmap:city!=* & mkgmap:admin_level10=* { set
mkgmap:city='${mkgmap:admin_level10|subst:Gemeinde |subst:Stadt }' }
mkgmap:country=AUT & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8|subst:Gemeinde |subst:Stadt }' }

mkgmap:country=NLD & mkgmap:city!=* & mkgmap:admin_level10=* { set
mkgmap:city='${mkgmap:admin_level10}' }
mkgmap:country=BEL & mkgmap:city!=* & mkgmap:admin_level9=* { set
mkgmap:city='${mkgmap:admin_level9}' }
mkgmap:country=CZE & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
mkgmap:country=CZE & mkgmap:city!=* & mkgmap:admin_level7=* { set
mkgmap:city='${mkgmap:admin_level7}' }
mkgmap:country=DNK & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
mkgmap:country=DNK & mkgmap:city!=* & mkgmap:admin_level7=* { set
mkgmap:city='${mkgmap:admin_level7}' }
mkgmap:country=FIN & mkgmap:city!=* & mkgmap:admin_level9=* { set
mkgmap:city='${mkgmap:admin_level9}' }
mkgmap:country=FIN & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
mkgmap:country=FRA & mkgmap:city!=* & mkgmap:admin_level9=* { set
mkgmap:city='${mkgmap:admin_level9}' }
mkgmap:country=FRA & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
mkgmap:country=ISL & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
```

```
mkgmap:country=ITA & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
mkgmap:country=NOR & mkgmap:city!=* & mkgmap:admin_level9=* { set
mkgmap:city='${mkgmap:admin_level9}' }
mkgmap:country=POL & mkgmap:city!=* & mkgmap:admin_level10=* { set
mkgmap:city='${mkgmap:admin_level10}' }
mkgmap:country=POL & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
mkgmap:country=PRT & mkgmap:city!=* & mkgmap:admin_level9=* { set
mkgmap:city='${mkgmap:admin_level9}' }
mkgmap:country=PRT & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
mkgmap:country=SVN & mkgmap:city!=* & mkgmap:admin_level10=* { set
mkgmap:city='${mkgmap:admin_level10}' }
mkgmap:country=ESP & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
mkgmap:country=SWE & mkgmap:city!=* & mkgmap:admin_level9=* { set
mkgmap:city='${mkgmap:admin_level9}' }
mkgmap:country=SWE & mkgmap:city!=* & mkgmap:admin_level7=* { set
mkgmap:city='${mkgmap:admin_level7}' }
mkgmap:country=CHE & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }

# Ecuador = ECU
mkgmap:country=ECU & mkgmap:region!=* & mkgmap:admin_level4=* { set
mkgmap:region='${mkgmap:admin_level4}' }
mkgmap:country=ECU & mkgmap:city!=* & mkgmap:admin_level6=* { set
mkgmap:city='${mkgmap:admin_level6}' }
mkgmap:country=ECU & mkgmap:city!=* & mkgmap:admin_level7=* { set
mkgmap:city='${mkgmap:admin_level7}' }
mkgmap:country=ECU & mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }

# common rules for all the rest of countries
mkgmap:city!=* & mkgmap:admin_level8=* { set
mkgmap:city='${mkgmap:admin_level8}' }
mkgmap:city!=* & mkgmap:admin_level9=* { set
mkgmap:city='${mkgmap:admin_level9}' }
mkgmap:city!=* & mkgmap:admin_level10=* {
set mkgmap:city='${mkgmap:admin_level10}' }
mkgmap:city!=* & is_in:city=* { set mkgmap:city='${is_in:city}' }
mkgmap:city!=* & addr:city=* { set mkgmap:city='${addr:city}' }

mkgmap:postal_code!=* & mkgmap:postcode=* {
set mkgmap:postal_code='${mkgmap:postcode}' }
mkgmap:postal_code!=* & addr:postcode=* { set
mkgmap:postal_code='${addr:postcode}' }

mkgmap:street!=* & addr:street=* { set mkgmap:street='${addr:street}' }
mkgmap:street!=* & addr:housename=* { set mkgmap:street='${addr:housename}' }

mkgmap:phone!=* & phone=* { set mkgmap:phone='${phone}' }

mkgmap:is_in!=* & is_in=* { set mkgmap:is_in='${is_in}' }

mkgmap:housenumber!=* & addr:housenumber=* {
set mkgmap:housenumber='${addr:housenumber}' }
#addr:housenumber=* { set name='${addr:street} ${addr:housenumber}'
| '${addr:housenumber}' } [0x2900 resolution 24 continue]
```