

# Tricks with mkgmap styles 3

dealing with strings and  
other 'things'

*ideas to enhance your maps*

(c) N Willink 2012/15

latest edition : 13/1/2015

Introduction .....	4
Abbreviations.....	5
How to replace Road with Rd , Street with St.....	5
How to replace Rue with R. and Boulevard with BVD.....	6
How to replace the French 'des' / 'de' with d.....	6
If a name contains several words, how do I show only the first 2?.....	6
How to show abbreviations & still search for unabridged names .....	7
How to replace Park with prk ,Wood with wd .....	7
Naming .....	8
Left\$.....	8
Mid\$.....	8
Road shields and routes .....	9
How can I fit more than 2 route refs into a shield? .....	9
<i>What if 2 routes have 4 letters each?</i> .....	10
How to make Toilets searchable on your Garmin .....	10
FIXMES.....	11
How can I show all the fixmes on my map?.....	11



## ***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](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 .

In this volume we are examining some of the string filter tools available:

- subst
- part
- substring

We recommend you download the latest mkgmap & mapuploader3.

All suggestions are welcome (osm at pinns . co . uk).

# Abbreviations

## How to replace Road with Rd , Street with St

Not only does it save space in an img , it also makes your map less cluttered.

For this we use the subst: filter - see mkgmap's manual

Illustration shows roads (Rd) , drives (Dr) , Close (Cls)



*The following applies to countries which place 'street' at the end of a name.*

### example1:

```
name~'.*Road' & highway=* { set name='${name|subst:"Road=>RD"}' }
```

Place this in lines before the highways are parsed

It appears 'name~'.\*Road' ' has to be inserted else if a name does not contain 'road' it gets deleted using the namelsubst filter.

All Roads now appear as Rd ( not RD - this is a Garmin issue.

The only way to make them appear as RD is to place a space between R and D ie

```
name~'.*Road' & highway=* { set name='${name|subst:"Road=>R D"}' }
```

### example2:

```
name~'.*Crescent' & highway=* { set name='${name|subst:"Crescent=>Cres"}' }
```

```
name~'.*Crescent' & highway=* { set name='${name|subst:"Crescent=>Cres"}' }
```

## ***How to replace Rue with R. and Boulevard with BVD***

Where 'road' is placed at the beginning of a name the code is slightly different:

```
name~'Boulevard.*' & highway=* { set name='${name|subst:"Boulevard=>BVD"}' }
```

```
name~'Rue.*' & highway=* { set name='${name|subst:"Rue=>R."}' }
```

## ***How to replace the French 'des' / 'de' with d.***

The following line of code does NOT work

```
name~'.*Des.*' & highway=* { set name='${name|subst:"Des=>D."}' } #Does Not WORK!
```

Instead we've opted to use the 'part' function to split a street name in to 3 parts.

'Part' splits the street name using 'space' as a separator. `${name|part:" :2"}`

```
highway=* & name=* {add 2ndwrdd='${name|part:" :2"}'} # check for existence of  
more than one word in sentence separated by space
```

```
highway=* & 2ndwrdd==* {set name='${name|part:" :1"} d. ${name|part:" :3"}' }
```

There is one problem - the above only caters for street names with 3 words ;

The above code would replace ' Rue de la gare ' with 'Rue d. la' .

However with some ingenuity it can be modified to cater for 4 words if the third word='la'.

To be honest , general mkgmap's string manipulation extremely cumbersome.

## ***If a name contains several words, how do I show only the first 2?***

I am indebted to Barry Bickerton who says:

'I have "National Byway (County Durham and Northumberland)" plastered all over my map so I have used the filter :

```
(ncn=yes | rcn=yes) { name '${rcnname|part:" <3"}' } [0x10018 resolution 22  
continue with_actions]
```

to extract only the first two words from the name, there seem to be quite a few routes which have the start and finish incorporated in their names.'

## ***How to show abbreviations & still search for unabridged names***

Showing abbreviations ( High St) is fine but you still want to be able to search for 'High Street'

Before you abbreviate the street names use mkgmap:label:

```
highway=residential & mkgmap:label:2!=* {set mkgmap:label:2='${name} (${ref})' | '${ref}' | '${name}'}
```

The manual states that label:3&4 are used for searching but it seems label:2 does the same

## ***How to replace Park with prk ,Wood with wd***

```
name~'.*Park { set name='${name|subst:"Park=>Prk"}'}
```

```
name~'.*Wood { set name='${name|subst:"Wood=>Wd"}'}
```

This time the above code is placed in the Polygons file.

However , it should be improved as Car Park will also be rendered as Car Pk

```
leisure=* & name~'.*Park { set name='${name|subst:"Park=>Prk"}'}
```

```
natural=wood & name~'.*Wood { set name='${name|subst:"Wood=>Wd"}'}
```

# Naming

In basic:

```
name$= ref$+"."+operator$
```

In mkgmap styles :

- a) check if ref\$ or operator\$ exists
- b) if so then set name to ref , add "." and operator

```
ref=* { set name='${ref}.${operator}' }
```

## Left\$

```
name="1234567"
```

```
Basic: name=left$(name,3) - ie 123
```

```
mkgmap: name=* { set name='${name|substring:0:3}' }
```

## Mid\$

```
name="abcdefg"
```

```
Basic: name=Mid$(name,3) , ie cdefg
```

```
mkgmap: name=* { set name='${name|substring:2}' }
```

```
name="abcdefg"
```

```
Basic: name=Mid$(name,3,3) , ie cde
```

```
mkgmap: name=* { set name='${name|substring:2:5}' }
```

Explanation :

first number sets 'start' of string , second number 'end' of string

substring:2:5 means from 3rd letter to letter before 6th letter - very confusing ; 0 is used as the first letter.

You can see the effect if you replace

```
name=* { name '${name}' } (found in default lines style file)
```

with

```
name=* { name '${namelsubstring:2:5}' } # 3,4,5
```

# Road shields and routes

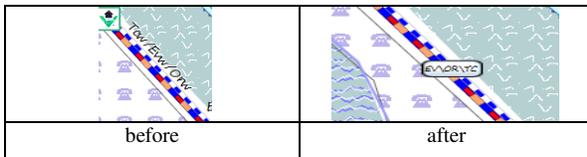
## How can I fit more than 2 route refs into a shield?

If the length of the ref is bigger than 8 then Garmin will not show the ref within a shield.  
For Oregon's the max length is 5!

So Evw/Tcw/Orw is shown separately as text.

It is most unlikely that more than 3 routes share the same highway. For this purpose we limit ourselves to a maximum of 3 routes.

With three routes you need two separators (/), so the maximum length for all refs is 6.



This means that you can play it safe by only taking the first 2 letters of each ref chunk.

- 1) check if a *third* ref chunk exists
- 2) if it does, then reduce each ref chunk to 2 letters
- 3) finally recreate the ref.

in relations, the totref (ie total refs) gets created for each highway  
ie

```
(route=foot | route=hiking)
{
    apply {
        set ref='${ref}';
set oref='${oref}/${ref}' | '${ref}';
    }
}
```

In lines we add the following - the code has not been maximised

```
oref=* {add part1='${oref|part:":1"}'}
oref=* {add part2='${oref|part:":2"}'}
oref=* {add part3='${oref|part:":3"}'}
# make them only 2 letters
part1=* & part3=* {set part1='${part1 | substring:"0:2"}'}
part1=* & part3=* {set part2='${part2 | substring:"0:2"}'}
part1=* & part3=* {set part3='${part3 | substring:"0:2"}'}
# now put the parts together
oref=* & part3=* {set oref='${part1}\${part2}\${part3} ' }
```

## **What if 2 routes have 4 letters each?**

Supposing two route each with a 4 letter ref share the same highway?

Unfortunately there is no len() function to determine the length of a ref so we have to check if there is a 4th letter:

```
part1=* {set p14='${part1|substring:3:4}'}  
part2=* {set p24='${part2|substring:3:4}'}
```

also we need a variable which is set only if there are two routes only (ie p2='two\_only')

```
p2='two_only' & p14=* & p24=* {  
  set part1='${part1 | substring:"0:3"}' ;  
  set part2='${part2 | substring:"0:3"}' ;  
  set oref='${(part1)\$(part2)}' ;  
}
```

## **How to make Toilets searchable on your Garmin**

There have been many a request to show how to enable a search for the nearest public convenience on your Garmin.

Garmin , unfortunately, doesn't make it easy and presumes we are all travelling by car.

Toilets are listed under Auto services !!!

So , search Auto Services / Rest Area / Tourist info

More than likely your osm toilets don't appear .

*If they don't , change your amenity=toilet type number to 2FOC !*

Alternatively, select your own category and check which pois appear - then change your toilets to the type number of one of the pois that is listed.

# FIXMES

## *How can I show all the fixmes on my map?*

If you are a osm plotter you may like to show any fixmes AND display what has to be fixed.

- Create a small icon in your TYP file and insert the following in your points.
- Click the POI line option in mapuploader3
- Place the following in your points file

```
(fixme=* | mkgmap:line2poi=true & fixme=*){ set name= '${fixme}' }[0x6117  
resolution 24]
```

Picture below showing fixmes ( 'incomplete', 'rough' & some unspecified)

