```
# Trekking-Poutnik profile, template for Trekking profile variants
# Template version 2.5.2  + no code chnage, Notepad++ folding comment blocks inserted
# See also https://github.com/poutnikl/Brouter-profiles/wiki
# and https://github.com/poutnikl/Trekking-Poutnik
#
# LEGEND
#
# Legend above is placeholder for generated comments of final profile
# See the profile bottom for changelogs and verbose *) comments

# bstart /global
---context:global
assign   isweet                 0 # 0 as default, *) flag for weather conditions

assign   consider_elevation     1 # 1 as default
assign   consider_smoothness    1 # 1 as default
assign   set_uphill_cost        1 # 1 as default
assign   set_downhill_cost      1 # 1 as default
assign   allow_steps            1 # 1 as default
assign   allow_ferries          1 # 1 as default
assign   allow_traffic_penalty  1 # 1 as default

assign   turnInstructionMode  = 1  # 0=none, 1=auto-choose, 2=locus-style, 3=osmand-style

assign   cycleroutes_pref       0.2 # **) costfactor penalty for not being cycleroute
assign   routelevel             2 # *****)  default =2, 1-icn only, 2 +ncn, 3 +rcn, 4 all

assign   MTB_factor             0.0 # default 0.0, see ****), see also https://github.com/poutnikl/Brouter-profiles/wiki/Trekking-MTB-Profiles---legend
                                    # major rework ! unified with granularity of smallpaved factor
assign   smallpaved_factor      0.0 # default 0.0, if <> 0.0 then it penalizes/promotes mainroads and unpaved roads for positive/negative values.
                                    # reworked

assign   hills                  1 # 1=default,                         0=trekking profile default,                1=try to avoid steep hills >3.0%,
                                  # 2=velomobile-like avoiding slopes, 3= simulates ascend/length time equiv.     4=valley mode

assign   valley_nonflat_multiplier = 2.0


assign   use_proposed_cn        1 # 0 as default, considers proposed cycle networks as unmarked but valid cycleroutes.
assign   avoid_unsafe           0 # 0 as default, gives penalty to road without bike friendly status.

assign   path_preference        0.0 # 0.0 as default, try 20.0 to penalize nonpath ways a/o paved ways



# Internal parameters

assign   uphillcostvalue        switch equal hills 1 70
                                switch equal hills 2 80
                                switch equal hills 3 60
                                switch equal hills 4 150
                                0
assign   uphillcutoffvalue      switch equal hills 1 3.0
                                switch equal hills 2 1.0
                                switch equal hills 3 0.5
                                switch equal hills 4 1.5
                                1.5
assign   downhillcutoffvalue    switch equal hills 1 1.5
                                switch equal hills 2 0.5
                                switch equal hills 3 1.5
                                switch equal hills 4 1.5
                                1.5
assign   downhillcostvalue      switch equal hills 1 60
                                switch equal hills 2 80
                                switch equal hills 3  0
                                switch equal hills 4 150
                                60

assign   validForBikes        1

assign positive_smallpaved_factor  or ( equal smallpaved_factor 0.0 ) ( greater smallpaved_factor 0.0 )

assign abs_smallpaved_factor  multiply ( switch positive_smallpaved_factor 1.0 -1.0 ) smallpaved_factor


assign   MTB_hillcostfactor   multiply -0.3333 ( max -3.0 ( multiply -1.0 ( max 0.0 MTB_factor ) ) )
                  # for MTBfactor <=0 is 0, for MTBfactor >=3 is 1, otherwise 0.3333 * MTBfactor
                  # progressively decreases hillcosts to be 0.0 at MTB_factor = 3.0
                  # if MTB_factor = 1 , then downhillcost decreases e.g. from 60 to 40

assign   downhillcost if ( and consider_elevation set_downhill_cost )  then
        ( multiply ( add 1.0 ( multiply MTB_hillcostfactor -1.0 ) ) downhillcostvalue ) else 0

assign   uphillcost   if ( and consider_elevation set_uphill_cost   )  then
        ( multiply ( add 1.0 ( multiply MTB_hillcostfactor -1.0 ) ) uphillcostvalue ) else 0

assign   uphillcutoff    if ( and consider_elevation set_uphill_cost )   then    uphillcutoffvalue else 1.5
assign   downhillcutoff  if ( and consider_elevation set_downhill_cost ) then  downhillcutoffvalue else 1.5

assign   elevationpenaltybuffer if ( equal hills 4 ) then 10 else 5   # 5 is trekking default
assign   elevationmaxbuffer     if ( equal hills 4 ) then 20 else 10  # 10 is trekking default
assign   elevationbufferreduce  if ( equal hills 4 ) then 0.0 else ( multiply 0.333 max uphillcutoff downhillcutoff ) # 0.0 is trekking default

assign   uphillCFshift     0.0 # experimental shifting of up/downhillcostfactors as alternative way of
assign   downhillCFshift   0.0 # prioritizing/penalizing of up/downhills, based on length, not elevation

assign    network_coef           add 1.0 ( multiply -1.0 cycleroutes_pref )  # 1 - cycleroutes_pref
assign    no_network_coef        add 1.0 cycleroutes_pref                    # 1 + cycleroutes_pref

assign    pass1coefficient 1.8
assign    pass2coefficient 0

assign   unsafe_penalty       1.0 # 1 - was 2 in Trekking, but this may cause U-like safe routes
assign   cost_of_unknown      2.0 # 2 as default, cost of highway=

# bend /global
# bstart /way
---context:way   # following code refers to way-tags

assign isicn route_bicycle_icn=yes
assign isncn or route_bicycle_ncn=yes ncn=yes
assign isrcn or route_bicycle_rcn=yes rcn=yes
assign islcn or route_bicycle_lcn=yes lcn=yes

assign any_cycleroute or route_bicycle_icn=yes or route_bicycle_ncn=yes or route_bicycle_rcn=yes route_bicycle_lcn=yes

assign any_cn_yes or lcn=yes or rcn=yes ncn=yes
assign any_cn_proposed or lcn=proposed or rcn=proposed ncn=proposed
assign any_cn or any_cn_yes switch use_proposed_cn any_cn_proposed 0

assign nodeaccessgranted or any_cycleroute any_cn

assign is_ldcr if      not nodeaccessgranted         then false
               else if equal cycleroutes_pref 0.0    then false
               else if equal routelevel 1 then isicn
               else if equal routelevel 2 then or isicn isncn
               else if equal routelevel 3 then or or isrcn isncn isicn
               else if equal routelevel 4 then any_cn
               else or isicn isncn


assign isbike or bicycle=yes|permissive|designated nodeaccessgranted
assign ispaved surface=paved|asphalt|concrete|paving_stones
assign isunpaved not or surface= or ispaved or surface=fine_gravel surface=cobblestone

assign probablyGood or ispaved  and isbike not isunpaved  #probably good surface - weather independent

assign istrack  highway=track|road|path|footway
assign ismainroad highway=motorway|motorway_link|trunk|trunk_link|primary|primary_link|secondary|secondary_link|tertiary|tertiary_link

assign ismuddy and or isunpaved surface=
               and iswet not surface=gravel|pebblestone   # ***)

assign turncost       if  ismainroad  then 90    else if ispaved then 60   else 30
# Turn cost is based solely on estimation of road speediness and eventual time cost of turning, partially promoting tracks )

assign initialclassifier
if route=ferry then 1
else if ( highway=motorway|motorway_link )       then 2
else if ( highway=trunk|trunk_link )             then 3
else if ( highway=primary|primary_link )         then 4
else if ( highway=secondary|secondary_link )     then 5
else if ( highway=tertiary|tertiary_link )       then 6
else if ( highway=unclassified )                 then 7
else if ( highway=residential   )                then 8
else if ( highway=living_street  )               then 9
else if ( highway=service   )                    then 10
else if ( highway=footway   )                    then 11
else if ( istrack )                              then 12
else if    ( highway=cycleway )                  then 13
                                                 else 14

assign initialcost
if route=ferry then 10000
else if ( highway=motorway|motorway_link|trunk|trunk_link ) then 500
else if ( highway=primary|primary_link )                    then 120
else if ( highway=secondary|secondary_link )                then 80
else if ( highway=tertiary|tertiary_link|unclassified )     then 50
else if ( highway=residential|living_street|service   )     then 30
                                                            else 0

assign defaultaccess =
       if access= then not motorroad=yes
       else if access=private|no then false
       else true


assign bikeaccess =
       if nodeaccessgranted then true
       else if bicycle= then
       (
         if vehicle= then defaultaccess
         else not vehicle=private|no
       )
       else not bicycle=private|no|dismount

assign footaccess =
       if bikeaccess then true
       else if bicycle=dismount then true
       else if foot= then defaultaccess
       else not foot=private|no

assign accesspenalty =
       if bikeaccess then 0
       else if footaccess then 4
       else 100000


#
# handle one-ways. On primary roads, wrong-oneways should
# be close to forbidden, while on other ways we just add
# 4 to the costfactor (making it at least 5 - you are allowed
# to push your bike)
#
assign badoneway =
       if reversedirection=yes then
         if oneway= then junction=roundabout else oneway=yes|true|1
       else oneway=-1


assign onewaypenalty =
       if ( badoneway ) then
       (
         if ( cycleway=opposite|opposite_lane|opposite_track ) then 0
         else if ( oneway:bicycle=no                         ) then 0
         else if ( highway=primary|primary_link              ) then 50
         else if ( highway=secondary|secondary_link          ) then 30
         else if ( highway=tertiary|tertiary_link            ) then 20
         else 4.0
       )
       else 0.0


# bstart /roughness_penalty
```

```
assign smoothness_penalty
# http://wiki.openstreetmap.org/wiki/Key:smoothness

if not consider_smoothness                                then 0.0 else
if smoothness=                                            then 0.0 else
if smoothness=excellent|very_good|good                    then 0.0 else
if smoothness=intermediate                                then 0.2 else
if smoothness=bad|robust_wheels                           then 0.5 else
if smoothness=very_bad|high_clearance                     then 1.0 else
if smoothness=horrible|off_road_wheels                    then 2.0 else
if smoothness=very_horrible                               then 4.0 else
if smoothness=impassable                                  then 10.0 else
0.0
# bmid /roughness_penalty
assign mtb_scale_penalty  if not consider_smoothness then 0.0 else

# Remember this is for trekking bikes, not MTB. There are MTB dedicated profiles.
# http://wiki.openstreetmap.org/wiki/Key:mtb:scale

if mtb:scale=        then 0.0 else   if mtb:scale=0-      then 0.0 else
if mtb:scale=0       then 0.0 else   if mtb:scale=0+      then 0.3 else
if mtb:scale=1-      then 0.7 else   if mtb:scale=1       then 1.2 else
if mtb:scale=1+      then 1.8 else   if mtb:scale=2-      then 2.5 else
if mtb:scale=2       then 4.0 else   if mtb:scale=2+      then 8.0 else
if mtb:scale=3       then 15.0 else
if mtb:scale=4|5|6   then 9000
                     else 0.0

assign roughness_penalty max smoothness_penalty mtb_scale_penalty
# bend /roughness_penalty

assign univ_factor_coef = (

         if ( highway=motorway|motorway_link|trunk|trunk_link ) then 1.0
    else if ( highway=primary|primary_link )                    then 0.9
    else if ( highway=secondary|secondary_link )                then 0.7
    else if ( highway=tertiary|tertiary_link )                  then 0.4
    else if ( highway=unclassified )                            then 0.25
    else if ( highway=residential|living_street|service|pedestrian|footway   )
                                                                then switch ispaved 0.25 switch isunpaved -0.4 0.0
    else if not highway=track|road|path                         then switch ispaved 0.25 switch isunpaved -0.4 0.0 #
    else if highway=track|road then (                           # not path
              if surface=concrete                               then -0.2
              else if and tracktype=grade1 ispaved              then 0.2
              else if or tracktype=grade1 ispaved               then 0.1
              else multiply ( add 1.0 ( multiply 0.33 roughness_penalty ) )
                ( if surface=cobblestone                        then -0.4
                  else if surface=fine_gravel                   then -0.4
                  else if surface=                              then switch tracktype=grade2|grade3 -0.4 -0.7
                  else if surface=compacted                     then -0.5
                  else if surface=grass                         then -0.8
                  else if surface=ground|gravel|pebblestone|mud then -1.0
                                                                else -1.0
                )
         )
         # is path
     else ( multiply ( add 1.0 ( multiply 0.33 roughness_penalty ) )
                   ( if surface=concrete                   then -0.3
                     else if ispaved                       then  0.1
                     else if surface=compacted|fine_gravel then -0.4
                     else                                       -1.0
                     )
          )

)

assign MTB_factor_for_road
   if  ( equal MTB_factor 0.0 ) then 0.0
                                else multiply MTB_factor  univ_factor_coef

assign abs_univ_factor =
    if ( greater 0.0 univ_factor_coef ) then ( multiply -1.0 univ_factor_coef )
                                        else univ_factor_coef

assign smallpaved_factor_for_road

if  ( equal smallpaved_factor 0.0 ) then 0.0
else multiply abs_smallpaved_factor
              if  positive_smallpaved_factor then abs_univ_factor
                                             else   ( add 1.0 multiply -1.0 abs_univ_factor )

assign trackclass_penalty  (
if not istrack              then 0.0
else if highway=track       then 0.0
else if highway=path        then 0.5
else if highway=road        then 0.0
else if highway=footway     then ( if and footway=sidewalk not isbike then 4.0 else 0.5 )
                            else 0.0
)

assign tracktype_penalty (
if not istrack              then 0.0    else if tracktype=       then 0.2
else if tracktype=grade1    then 0.0    else if tracktype=grade2 then 0.3
else if tracktype=grade3    then 0.6    else if tracktype=grade4 then 1.0
else if tracktype=grade5    then 1.5    else 0.1
)

assign surface_penalty (
if not istrack                                  then 0.0
else if surface=asphalt|paved                   then 0.0
else if surface=concrete|paving_stones          then 0.1
else if surface=cobblestone|sett                then 0.2
else if surface=compacted|fine_gravel           then 0.2
else if surface=                                then 0.1
else if surface=ground|earth|unpaved|dirt       then 0.3
else if surface=grass|sand                      then 1.0
else if surface=gravel|pebblestone              then 1.2
                                                else 0.3
)

assign not_isbike_track_penalty
if not istrack                     then 0.0  # this is track context / isbike for mainroads is addressed elsewhere.// 2.2.1 ALFA
```

```
else if or ispaved isbike          then 0.0
else if and tracktype= surface=    then 4.0
else if tracktype=                 then 3.0
else if tracktype=grade1           then 0.2
else if tracktype=grade2           then 0.6
else if tracktype=grade3           then 1.2
else if tracktype=grade4           then 2.0
else if tracktype=grade5           then 3.0
                                   else 3.0

assign wet_penalty
if   not iswet                      then 0.0
else if not istrack                 then 0.0
else if ispaved                     then 0.0
else if surface=compacted           then 0.6
else if surface=fine_gravel         then 0.8
else if surface=cobblestone         then 0.5
else if surface=                    then if highway=path then 2.0 else 1.0
else if surface=grass               then if highway=path then 2.5 else 1.5
else if surface=ground|earth|unpaved|sand|dirt|mud
                                    then if highway=path then 3.5 else 2.0
                                    else 2.0

assign hascycleway = not
  and ( or cycleway= cycleway=no|none ) and ( or cycleway:left= cycleway:left=no )  ( or cycleway:right= cycleway:right=no )

assign trafficpenalty0 = (
    if highway=primary|primary_link then
    (
      if      estimated_traffic_class=4 then 0.2
      else if estimated_traffic_class=5 then 0.4
      else if estimated_traffic_class=6|7 then 0.6
      else 0
    )
    else if highway=secondary|secondary_link then
    (
      if      estimated_traffic_class=3 then 0.2
      else if estimated_traffic_class=4 then 0.4
      else if estimated_traffic_class=5 then 0.6
      else if estimated_traffic_class=6|7 then 1
      else 0
    )
    else if highway=tertiary|tertiary_link then
    (
      if      estimated_traffic_class=2 then 0.1
      else if estimated_traffic_class=3 then 0.3
      else if estimated_traffic_class=4 then 0.5
      else if estimated_traffic_class=5|6|7 then 1
      else 0
    )
    else 0
    )

assign trafficpenalty =
  if hascycleway then min 0.3 trafficpenalty0
  else trafficpenalty0

assign nonpath_penalty = (
    if      ( equal path_preference 0.0 )   then 0.0                # nonpath_penalty inactive
    else if not istrack                     then path_preference    #istrack = highway=track/path/road/footway
    else if ispaved                         then ( multiply path_preference 0.5 )
    else if or  ( and not isunpaved
                      not highway=path )
                ( tracktype=grade1|grade2 ) then ( multiply path_preference 0.25 )
    else if not ( and isunpaved
                  and highway=path
                  and tracktype=grade1|grade2
                  not surface=gravel|cobblestone|pebblestone )
                                            then                    ( multiply path_preference 0.125 )
                                            else                    0.0

    )

assign rawcostfactor ( # can be <1, adjusted to >=1 in final step of calculation of costfactor, uphillcostfactor,downhillcostfactor


  add ( max onewaypenalty max accesspenalty  switch allow_traffic_penalty trafficpenalty 0 )

  add nonpath_penalty

  switch and highway= not route=ferry  100000
  switch    highway=steps  switch allow_steps   40 100000
  switch    route=ferry    switch allow_ferries 5.67 100000


  switch    highway=pedestrian                        switch ismuddy 5 3
  switch    highway=bridleway                         switch ismuddy 8 5
  switch    highway=cycleway                          ( switch ismuddy 1.8 switch isunpaved 1.5 1.0 )
  switch    highway=residential|living_street         ( switch ismuddy 2.0 switch isunpaved 1.6 1.2 )
  switch    highway=service                           ( switch ismuddy 1.8 switch isunpaved 1.5 1.1 )

  if istrack then ( add 1.0  max trackclass_penalty
                             max tracktype_penalty
                             max surface_penalty
                             max not_isbike_track_penalty
                             max roughness_penalty
                                 wet_penalty
                  )

 else

# Penalty for unsafe roads. As wet fallback, it does not so strictly avoid unsafe

  add if ( and avoid_unsafe not isbike ) then ( multiply unsafe_penalty     ( if ( iswet ) then    0.5 else 1.0 ) )
                                         else 0

  switch highway=motorway|motorway_link|proposed|abandoned|construction  100000

  switch highway=trunk|trunk_link          ( switch isbike switch iswet 1.5 1.8 switch iswet 5 10 )
  switch highway=primary|primary_link      ( switch isbike switch iswet 1.2 1.5 switch iswet 2 3 )
  switch highway=secondary|secondary_link  ( switch isbike switch iswet 1.1 1.2 switch iswet 1.4 1.6 )
  switch highway=tertiary|tertiary_link    ( switch isbike switch iswet 1.0 1.1 switch iswet 1.2 1.4 )
  switch highway=unclassified              ( switch isbike switch iswet 1.0 1.0 switch iswet 1.15 1.3 )

  cost_of_unknown # above highway tags do not apply
  )


  assign rawcostfactor2 (                                        # MTB_factor_for_road + (non)/cycleroute CF adjustment
    add (
    if ( equal cycleroutes_pref 0.0 )  then rawcostfactor                        # ignore cycleroutes, no distinguishing
    else if ( not nodeaccessgranted  ) then add (    multiply rawcostfactor no_network_coef )
                            (    multiply -1.0 ( multiply cycleroutes_pref network_coef ) )
                        # is not a cycleroute
                        # rawcostfactor2 = rawcostfactor * no_network_coef - cycleroutes_pref * network_coef
                        # RCF2 = RCF * ( 1 + crp ) - crp * ( 1- crp )
        else if is_ldcr                       then add (    multiply rawcostfactor network_coef )
                            (    multiply cycleroutes_pref network_coef )
                        # is a considered cycleroute
                        # rawcostfactor2 = rawcostfactor * network_coef + cycleroutes_pref * network_coef
                        # RCF2 = RCF * ( 1 - crp ) + crp * ( 1- crp )

    else                   rawcostfactor                        # is a cycleroute, but not a considered one
    )

    add   MTB_factor_for_road           # penalizes paved and promotes unpaved roads See *****) for more
          smallpaved_factor_for_road    # penalizes mainroads and unpaved roads, promotes the middle
    )

  assign costfactor  # calculations may provide illegal costfactor < 1
    max 1.0 rawcostfactor2



  assign uphillcostfactor = (
    max 1.0
    if ( equal hills 4 ) then ( multiply rawcostfactor2 valley_nonflat_multiplier )
    else
    add uphillCFshift
    add rawcostfactor2
    if   ismainroad                                 then 0.1
    else if highway=residential|living_street       then -0.1
    else if ispaved                                 then -0.1
    else if surface=gravel|pebblestone              then 0.2
    else if ismuddy                                 then 0.2
    else if isunpaved                               then 0.1
                                                    else 0.0
    )

  assign downhillcostfactor =  (
    max 1.0
    if ( equal hills 4 ) then ( multiply rawcostfactor2 valley_nonflat_multiplier )
    else
    add downhillCFshift
    add rawcostfactor2
    if   ismainroad                                 then -0.1
    else if  highway=residential|living_street      then 0.0
    else if  ispaved                                then -0.1
    else if surface=gravel|pebblestone              then 0.5
    else if surface=fine_gravel|sand                then 0.4
    else if ismuddy                                 then 0.5
    else if tracktype=grade5                        then 0.4
    else if tracktype=grade4                        then 0.3
    else if highway=path|footway                    then 0.2
    else if tracktype=grade3                        then 0.2
    else if surface=cobblestone                     then if iswet then 0.3 else 0.1
    else if tracktype=grade2                        then 0.1
    else if isunpaved                               then 0.2
                                                    else 0.0
    )

# way priorities used for voice hint generation

assign priorityclassifier = (

  if      ( highway=motorway                  ) then  30
  else if ( highway=motorway_link             ) then  29
  else if ( highway=trunk                     ) then  28
  else if ( highway=trunk_link                ) then  27
  else if ( highway=primary                   ) then  26
  else if ( highway=primary_link              ) then  25
  else if ( highway=secondary                 ) then  24
  else if ( highway=secondary_link            ) then  23
  else if ( highway=tertiary                  ) then  22
  else if ( highway=tertiary_link             ) then  21
  else if ( highway=unclassified              ) then  20
  else if ( highway=residential|living_street ) then  16
  else if ( highway=service                   ) then  14
  else if ( tracktype=grade1                  ) then  6
  else if ( highway=cycleway                  ) then  6
  else if ( bicycle=designated                ) then  6
  else if ( highway=bridleway                 ) then  4
  else if ( highway=track|road|footway        ) then  4
  else if ( highway=steps|path|pedestrian     ) then  2
  else 0
  )

# some more classifying bits used for voice hint generation...

assign isbadoneway = not equal onewaypenalty 0
assign isgoodoneway = if reversedirection=yes then oneway=-1
                      else if oneway= then junction=roundabout else oneway=yes|true|1
assign isroundabout = junction=roundabout
assign islinktype = highway=motorway_link|trunk_link|primary_link|secondary_link|tertiary_link
assign isgoodforcars = if greater priorityclassifier 6 then true
                  else if highway=residential|living_street|service then true
                  else if ( and highway=track tracktype=grade1 ) then true
                  else false

# ... encoded into a bitmask

assign classifiermask add          isbadoneway
                      add multiply isgoodoneway   2
                      add multiply isroundabout   4
                      add multiply islinktype     8
                          multiply isgoodforcars 16
# bend /way
# bstart /node
---context:node  # following code refers to node tags

assign   initial_cost_node    0     # 0 as default

assign defaultaccess =
       if ( access= ) then true # add default barrier restrictions here!
       else if ( access=private|no ) then false
       else true

assign bikeaccess =
       if nodeaccessgranted=yes then true
       else if bicycle= then
       (
         if vehicle= then defaultaccess
         else not vehicle=private|no
       )
       else not bicycle=private|no|dismount

assign footaccess =
       if bicycle=dismount then true
       else if foot= then defaultaccess
       else not foot=private|no


assign initialcost =
        add ( if highway=traffic_signals then 120
              else if highway=stop  then 60
              else if and highway=crossing bicycle=no then 60
              else if and highway=crossing crossing=no then 300 # workaround affecting road routing
              else initial_cost_node )
            if bikeaccess then 0  else ( if footaccess then 100 else 1000000 )

# bend /node
# bstart /comments

############################################################################################
#                                                                                          #
#                      V E R B O S E   L E G E N D S                                       #
#                                                                                          #
############################################################################################
# *) ISWET             iswet=0 is default with meaning of dry surfaces
############################################################################################
# iswet=1 with meaning surfaces are wet. It increases costfactors of potentially bad surfaces,
# especially those supposed to get muddy/slicky in wet weather,
# OTOH it decreases costfactors for better quality tracks or highway network.
# It shifts the profile little toward fastbike standard profile,
# but still focusses on trekking.
#
#
############################################################################################
# **)  CYCLEROUTE PERFECTNESS - removed
# Cycleroutes have calculates their costfactors and turncosts by the same way as regular ways,
# Regular ways have additional penalty cycleroutes_pref for not being cycleroute.
# 0.0 = ignore cycleroutes, 0.1-0.2 Prefering cycloroutes, >0.6 similar as stick_to_cycleroutes

############################################################################################
#
############################################################################################
# ***) ISMUDDY
############################################################################################
# ismuddy indicates potentially bad surface conditions during/after wet weather period( mud, slickness )
# Is used for increasing costfactor for such roads, and decreasing costfastor for minor highways/roads
# It requires iswet=1
#
#
#############################################################################################
# ****) MTB_factor
#############################################################################################
#
# MTB_factor tweaks/trims MTB approach of the profile  by preferring/penalizing in progressive order
# nonpaved  - preferred
# not paved - little preferred
# paved     - little penalized
# mainroads - penalized
#
# MTB_factor can be used for one-time tweaking of routing profile for particular trip,
# or trimming of the profile according to biker preferencing without need of profile deep insight
#
# Positive values progessively promote/penalize roads in favour of MTB riding.

# Negative value has the opposite effect, preferring mainroads and penalizing unpaved roads.
# This effect is somewhat similar to iswet=1 ( *) wet weather mode ),
# but does not distinguish particular road classes / surfaces / smoothness,
# aside of mentioned schema below.

# The calculated values below is added to the costfactor.

# + MTB_factor                                    for main roads (tertiaries and better),
# + 0.33 * MTB_factor                             for paved roads,
# - 0.33 * MTB_factor                             for not paved/not unpaved roads,
# - MTB_factor * ( 1 + 0.33 * smoothnesspenalty ) for unpaved roads.  - at MTB_factor 3.0 smootheness is ignored
#
# Default   is 0.0 = no effect.
# Recommended -0.5 - +1.0
# Reasonable  -2.0 .. +3.0,
#
# Final costfactor is kept >= 1 for final costfacto values.
#
#############################################################################################
# *****) routelevel
#############################################################################################
#
# global parameter routelevel determines what cyclenetwork levels are preferred by cycleroutes_pref
#            routelevel = 1  icn only
#            routelevel = 2  icn + ncn only
#            routelevel = 3  icn + ncn + rcn only
#            routelevel = 4  all icn + ncn + rcn + lcn, including sproposed ones, if use_proposed_cn=1
# Regardless of routelevel value,
# all current + proposed  networks are used in nodeaccessgranted and isbike
#
#######################################################################################
# History log
#######################################################################################
# feature flags: + = new / ! = fixed / - = removed / * = changed or improved
# Version 2.0.0 BETA - * organizing script without code changes
# Version 2.0.1 BETA - ! node initial cost logic (Mar 26 )
# Version 2.0.2 BETA - * partial syntax conversion (May 8 ), simplified uphill/downhill costs
# Version 2.1   BETA - * abandoned default compatibility with reference Trekking profile
# Version 2.1.1 BETA - + MTB_factor + up/downhillcostfactors
# Version 2.1.2 BETA - + 3state muddy/wet/dry track costfactors
# Version 2.1.3 BETA - + class-selective cycloroute usage
# Version 2.1.4 BETA - + smoothness penalty
# Version 2.1.5 RELEASE  - + hill profiles
# Version 2.1.6 BETA - + mtb:scale integrated to smoothness penalty
# Version 2.1.7 BETA - + smoothnesspenalty integrated to MTB factor,
#                      * default routelevel=2 and cycleroutes_pref=0.1
# Version 2.1.8 BETA - * splitting smoothness penalty
# Version 2.1.9 ALFA - * Modified Cycleroute preference calculation - non cycleroutes, cycleroutes not considered, cycleroutes considered,
# Version 2.1.10 ALFA - * increased smoothness penalties
# Version 2.1.11 RELEASE- * modified MTB and smoothness penalties
# Version 2.2.1 ALFA - * Track costfactors penalty driven now
# Version 2.2.2 ALFA - * highway=path penalty for downhill, * up/downhill costafactor , modified cycloroute preferencing
# Version 2.2.3 ALFA - + initialclassifier + tweakeked track penalties
# Version 2.2.4 ALFA - + MTB factor affects hillcosts/cutoffs
# Version 2.2.5 ALFA - * synced with 2.1.14 RELEASE - but Track penalty systemor affects hillcosts/cutoffs
# Version 2.2.6 ALFA - * modified node initial cost, addressing some highway=crossing issues
# Version 2.2.7 ALFA - * Tweaked mainroad + residental costfactors
# Version 2.2.8 ALFA - * backporting to 2.1 Release
# Version 2.2.9 ALFA - * Rearranging global context
# Branch  2.3   ALFA - + Maximizing Track penalty system, instead of additive system of brach 2.2,
#                      * partial penalty values changed, to follow different approach
# Version 2.3.1 ALFA - * Tweaked Track penalties, * tweaked Rougness penalties
# Version 2.3.2 ALFA - * Tweaked Track penalties, * rearranging general context code
# Version 2.3.3 ALFA - * Tweaked Cycleroute preferencing to nonlinear progressive penalizing
# Version 2.3.4 ALFA - ! fixed mean cutoff variable
#                      * Tweaked Cycleroute preferencing to nonlinear progressive penalizing
#                      * Switched MTB_factor vs cycleroute evaluation order
# Version 2.3.5 ALFA - * changed  elevationbufferreduce  = 0.333 * max (uphillcutoffvalue,downhillcutoffvalue)
# Version 2.3.6 ALFA - ! fix  elevationbufferreduce  placement and logic,  = 0.333 * max (uphillcutoff,downhillcutoff)
# Version 2.3.7 ALFA - -  Removed MTB_hillcutfactor, i.e. MTF factor affects hillcosts only.
# Version 2.4.1 ALFA - +  traffic penalty from fastbike low traffic profile
# Version 2.4.2 ALFA - +  smallpaved_factor
# Version 2.4.3 ALFA - *! smallpaved_factor rearranging and fixing error for negative values
# Version 2.4.4 ALFA - +  experimental up/downhillcostfactor shifts
# Version 2.4.5 ALFA - +  experimental small path only prioritization
# Version 2.4.6 ALFA - +  added priorityclassifier
# Version 2.4.7 ALFA - *  tweaked priorityclassifier value for footway and path
# Version 2.4.8 ALFA - +  assign   turnInstructionMode  = 2  # 0=none, 1=gpsies-style, 2=locus-style
# Version 2.4.9 BETA - *  updated  turnInstructionMode, code cleanup, release preparation
# Version 2.4.10 BETA - *  merged with recent trekking.brf changes
# Version 2.4.11 RELEASE - *  merged with release trekking.brf changes
# Version 2.4.12 BETA - + implemented valley mode ( assign hills 4 )
# Version 2.4.13 BETA - Fixed default hills value back to 1
# Version 2.4.14 ALFA -  * Switching MTB Factor granularity to smallpaved factor one - with keeping its effect, of course. Small Smoothness tweak
#                        * tweaked costfactors for cycleway/residential|living_street/service
# Version 2.4.15 ALFA -  * Tweaked universal MTB/smallpaved factor coeficients
# Version 2.4.16 RELEASE * Updated trafficpenalty calculation from fastbike-lowtraffic
# Template version 2.5.1  * tweaked track costs
# bend /comments
```