

```

#!/bin/sh

# MRTK-Variablen setzen
str_Version_Nummer="1.1"
str_Version_ID="66"
str_Version_Datum="2012-09-09 - 2013-07-14"
str_FID_gbasemap="21011"
str_FID_gsperrung="21012"

cd -- "$(dirname -- "$0")"
CD="CocoaDialog.app/Contents/MacOS/CocoaDialog"

if [ $filename == "" ];
then
  if [ ! "$@" = ' ' ];
  then
    filename=$(basename "$@")
    pathname=$(dirname "$@")
  else
    echo
    echo "MotoRoute-Tourer-Karte"
    echo "Version " $str_Version_Nummer "." $str_Version_ID " / " $str_Version_Datum
    echo "Geo-Daten von OpenStreetMap"
    echo "Erstellt für MotoRoute.de von Gernot Skottke"
    echo "mailto:motoroute@motorkuh.de"
    echo
    echo
    echo "Lade die Daten Deiner Wunsch-Region im Format '.osm.pbf' unter"
    echo "http://download.geofabrik.de/"
  echo
  echo "Lade aktuelle Boundaries (Grenzen & Meer) unter"
  echo "http://www.navmaps.eu/ in den gleichen Ordner."
  echo
  echo "Dann zieh die '.osm.pbf'-Datei der zu bauenden Region hier rein!"
  echo
  echo
  echo "Ich warte ..."
  exit
  fi
fi

while [ ! $filename = ' ' ]
do

  # MRTK-Variablen setzen
  str_Startdatum=$(date +%Y-%m-%d)
  str_Jahr=$(date +%Y)

  str_Region=$(basename "$@" .osm.pbf)

  # Name (ohne "-latest") und Kurzname der Region festlegen
  str_RegionLaenge=${#str_Region}
  if [ "$str_RegionLaenge" > "7" ];
  then
    if [ "${str_Region: (-7)}" == "-latest" ];
    then
      str_Latest=${str_Region: (-7)}
      str_Region=${str_Region%$str_Latest}
    else
      str_Region=$str_Region
    fi
  fi

  if [ "${str_RegionLaenge}" > "2" ];
  then
    str_Region_kurz=${str_Region:0:3}
  else
    str_Region_kurz=$str_Region
  fi

  echo "Ich baue jetzt Deine 'MotoRoute-Tourer-Karte' für:"
  echo
  echo "    "$str_Region "("$str_Region_kurz")"
  echo

# -----
  echo
  echo "Step 1 Inaktives RAM ermitteln"

  #bash top -l 1 | awk '/PhysMem:/ {print $6}'

```

```

free="$(( $(vm_stat | grep free | awk '{ print $3 }' | sed 's/\./')*4096/1048576 ))"
inactive="$(( $(vm_stat | grep inactive | awk '{ print $3 }' | sed
's/\./')*4096/1048576 ))"

```

```

let val_RAM=$free+$inactive
echo $val_RAM"m wird für Java reserviert."

```

```

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 2 Prüfung Ordner /osm_mrtk/"

```

```

if [ -d ~/osm_mrtk ];
then
echo "Ordner /osm_mrtk/ ist vorhanden."
else
mkdir ~/osm_mrtk
echo "Ordner /osm_mrtk/ wurde angelegt."
fi

```

```

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 3 Prüfung Ordner /tmp/"

```

```

if [ -d ~/osm_mrtk/tmp ];
then
echo "Ordner /tmp/ ist vorhanden."
else
mkdir ~/osm_mrtk/tmp/
echo "Ordner /tmp/ wurde angelegt."
fi

```

```

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 4 Java-Version"

```

```

java -version 2> ~/osm_mrtk/tmp/java.log
echo $(head -n 1 ~/osm_mrtk/tmp/java.log)

```

```

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 5 Prüfung Ordner /maps/"

```

```

if [ -d ~/osm_mrtk/maps ];
then
echo "Ordner /maps/ ist vorhanden."
else
mkdir ~/osm_mrtk/maps/
echo "Ordner /maps/ wurde angelegt."
fi

if [ -d ~/osm_mrtk/maps/$str_Region ];
then
rm -rf ~/osm_mrtk/maps/$str_Region
else
mkdir ~/osm_mrtk/maps/$str_Region
fi

```

```

mkdir ~/osm_mrtk/maps/$str_Region
mkdir ~/osm_mrtk/maps/$str_Region/SD
mkdir ~/osm_mrtk/maps/$str_Region/SD/GARMIN
mkdir ~/osm_mrtk/maps/$str_Region/SD/GARMIN/GPX
mkdir ~/osm_mrtk/maps/$str_Region/SD/GARMIN/POI

```

```

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 6 Nützliche Infos 'pdf_"$str_Jahr".zip' laden und entpacken"

```

```

if [ ! -d ~/osm_mrtk/pdf_$str_Jahr ];
then
curl -o ~/osm_mrtk/pdf_$str_Jahr.zip -S -C \
--url "http://motorkuh.de/blog/download/pdf_"$str_Jahr".zip"
echo "Nützliche Infos 'pdf_"$str_Jahr".zip' wurden geladen und entpackt."
unzip ~/osm_mrtk/pdf_$str_Jahr.zip -d ~/osm_mrtk
if [ -f ~/osm_mrtk/pdf_$str_Jahr.zip ];
then
rm -rf ~/osm_mrtk/pdf_$str_Jahr.zip
fi
else
echo "Ordner /pdf_"$str_Jahr"/ vorhanden."

```



```

echo "Step 10 Version ermitteln der geladenen Boundaries"
echo "      'precompiled bounds' und 'precompiled sea'"

ls -l $pathname > ~/osm_mrtk/tmp/precompiled.txt

grep -o 'bounds_[0-9]*.zip' ~/osm_mrtk/tmp/precompiled.txt > ~/osm_mrtk/tmp/bounds.txt
echo "lese letzten Stand 'precompiled bounds'"

if [ -f $pathname/$(tail -n 1 ~/osm_mrtk/tmp/bounds.txt) ];
then
    echo "Precompiled Bounds sind vorhanden:"
    echo $(tail -n 1 ~/osm_mrtk/tmp/bounds.txt)
else
    rv=`$CD ok-msgbox --no-cancel --title "Vermissе Precompiled Bounds" --icon x \
        --text "bounds_*.zip ist nicht im Ordner:" \
        --informative-text "$pathname ----> Bitte lade zuerst die aktuellste Datei
'Precompiled Bounds' (bounds_*.zip) unter http://www.navmaps.org/boundaries/" \
        --no-newline --float`

    if [ "$rv" == "1" ]; then
        echo "Vermissе Precompiled Bounds"
        echo "'bounds_*.zip' ist nicht im Ordner: "$pathname
        echo "Bitte lade zuerst die aktuellste Datei 'Precompiled Bounds'
(bounds_*.zip) unter http://www.navmaps.org/boundaries/"
        exit
    fi
fi

grep -o 'sea_[0-9]*.zip' ~/osm_mrtk/tmp/precompiled.txt > ~/osm_mrtk/tmp/sea.txt
echo "lese letzten Stand 'precompiled sea'"

if [ -f $pathname/$(tail -n 1 ~/osm_mrtk/tmp/sea.txt) ];
then
    echo "Precompiled Sea ist vorhanden:"
    echo $(tail -n 1 ~/osm_mrtk/tmp/sea.txt)
else
    rv=`$CD ok-msgbox --no-cancel --title "Vermissе Precompiled Sea" --icon x \
        --text "sea_*.zip ist nicht im Ordner:" \
        --informative-text "$pathname ----> Bitte lade zuerst die aktuellste Datei
'Precompiled Sea' (sea_*.zip) unter http://www.navmaps.org/boundaries/" \
        --no-newline --float`

    if [ "$rv" == "1" ]; then
        echo "Vermissе Precompiled Sea"
        echo "'sea*.zip' ist nicht im Ordner: "$pathname
        echo "Bitte lade zuerst die aktuellste Datei 'Precompiled Sea' (sea*.zip)
unter http://www.navmaps.org/boundaries/"
        exit
    fi
fi

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 11 Neuesten 'splitter' ermitteln und laden"

curl -L http://www.mkgmap.org.uk/download/splitter.html > ~/osm_mrtk/tmp/splitter.html

grep -o 'splitter-r[0-9]*.zip' ~/osm_mrtk/tmp/splitter.html > ~/osm_mrtk/tmp/splitter.txt
echo "lese letzten Stand 'splitter'"
echo $(head -n 1 ~/osm_mrtk/tmp/splitter.txt)

splitterfolder=$(basename $(head -n 1 ~/osm_mrtk/tmp/splitter.txt) .zip)

if [ -d ~/osm_mrtk/$(basename $(head -n 1 ~/osm_mrtk/tmp/splitter.txt) .zip) ];
then
    echo "Splitter '$(basename $(head -n 1 ~/osm_mrtk/tmp/splitter.txt) .zip)'" ist
aktuell."
else
    rm -rf ~/osm_mrtk/splitter-r*
    echo "Alter Splitter wurde gelöscht."
    curl -o ~/osm_mrtk/$(head -n 1 ~/osm_mrtk/tmp/splitter.txt) -S -C \
        --url http://www.mkgmap.org.uk/download/$(head -n 1
~/osm_mrtk/tmp/splitter.txt)
    unzip ~/osm_mrtk/$(head -n 1 ~/osm_mrtk/tmp/splitter.txt) -d ~/osm_mrtk
    if [ -f ~/osm_mrtk/$(head -n 1 ~/osm_mrtk/tmp/splitter.txt) ];
    then
        rm -rf ~/osm_mrtk/$(head -n 1 ~/osm_mrtk/tmp/splitter.txt)
    fi
    echo "Neuester Splitter '$(head -n 1 ~/osm_mrtk/tmp/splitter.txt)'" wurde geladen
und entpackt."

```

```

fi
# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 12 Neuestes 'mkgmap' ermitteln und laden"

curl -L http://www.mkgmap.org.uk/download/mkgmap.html > ~/osm_mrtk/tmp/mkgmap.html

grep -o 'mkgmap-r[0-9]*.zip' ~/osm_mrtk/tmp/mkgmap.html > ~/osm_mrtk/tmp/mkgmap.txt
echo "lese letzten Stand 'mkgmap'"
echo $(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt)

mkgmapfolder=$(basename $(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt) .zip)

if [ -d ~/osm_mrtk/$(basename $(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt) .zip) ];
then
echo "MkGMap "$(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt)" ist aktuell."
else
rm -rf ~/osm_mrtk/mkgmap-r*
echo "Altes MkGMap wurde gelöscht."
curl -o ~/osm_mrtk/$(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt) -S -C \
--url http://www.mkgmap.org.uk/download/$(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt)
unzip ~/osm_mrtk/$(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt) -d ~/osm_mrtk
if [ -f ~/osm_mrtk/$(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt) ];
then
rm -rf ~/osm_mrtk/$(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt)
fi
echo "Neuestes MkGMap "$(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt)" wurde geladen und
entpackt."
fi

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 13 Alle Dateien geladen"

echo "- Region : "$pathname/$filename
echo "- bounds : "$(tail -n 1 ~/osm_mrtk/tmp/bounds.txt)
echo "- seas : "$(tail -n 1 ~/osm_mrtk/tmp/sea.txt)
echo "- style : "$(head -n 1 ~/osm_mrtk/tmp/style.txt)
echo "- sperrung: "$(head -n 1 ~/osm_mrtk/tmp/sperrung.txt)
echo "- splitter: "$(head -n 1 ~/osm_mrtk/tmp/splitter.txt)
echo "- mkgmap : "$(head -n 1 ~/osm_mrtk/tmp/mkgmap.txt)

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 14 splitter"

rm -rf ~/osm_mrtk/tiles

mkdir ~/osm_mrtk/tiles

echo "Ich splitte jetzt die Datei "$filename"."
echo "Die Laufzeit ist abhängig von der Dateigröße."
echo "Start splitter: "`date`" RAM: "$val_RAM"m"

cd ~/osm_mrtk/

java -Xmx$val_RAM"m" -jar ~/osm_mrtk/$splitterfolder/splitter.jar \
--max-nodes=1600000 \
--output-dir=./tiles \
$pathname/$filename \
> ~/osm_mrtk/tmp/$splitterfolder.log

cd -- "$dirname -- "$@"
# echo "Ich bin wieder in : "$(pwd)

echo "Ende : "`date`

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 15 mkgmap: [gbasemap] als 'gmapsupp.img'"

mkdir ~/osm_mrtk/tiles/gbasemap

cp ~/osm_mrtk/$stylefolder/motorout.TYP ~/osm_mrtk/tiles/gbasemap/motorout.TYP
echo "TYP-File kopiert nach:"
echo "/osm_mrtk/tiles/gbasemap"

echo "Ich baue jetzt die Karten-Kacheln für "$str_Region"."
echo "Die Laufzeit ist abhängig von der Dateigröße."

```

```

rm -rf ~/osm_mrtk/tiles/gbasemap/gmapsupp

cd ~/osm_mrtk

echo "Start: "`date`" RAM: "$val_RAM"m"

# Alternative: --generate-sea:multipolygon,extend-sea-sectors,close-gaps=6000,floodblocker \

java -Xmx$val_RAM"m" -jar ~/osm_mrtk/$mkgmapfolder/mkgmap.jar \
  --bounds=$pathname/${tail -n 1 ~/osm_mrtk/tmp/bounds.txt} \
  --precomp-sea=$pathname/${tail -n 1 ~/osm_mrtk/tmp/sea.txt} \
  --output-dir=./tiles/gbasemap/ \
  --max-jobs \
  --style-file=./$stylefolder/motorout_style/ \
  --description="osm mrtk "$str_Region" "$str_Startdatum \
  --country-name=$str_Region \
  --country-abbr=$str_Region_kurz \
  --family-id=$str_FID_gbasemap \
  --product-id=1 \
  --series-name="osm mrtk "$str_Region" "$str_Startdatum \
  --family-name="osm mrtk "$str_Region" "$str_Startdatum \
  --area-name=$str_Region_kurz \
  --code-page=1250 \
  --latin1 \
  --mapname=63240001 \
  --draw-priority=10 \
  --tdbfile \
  --add-pois-to-areas \
  --link-pois-to-ways \
  --remove-short-arcs \
  --levels=0:24,1:22,2:21,3:19,4:18,5:16 \
  --location-autofill=is_in,nearest \
  --merge-lines \
  --index \
  --net \
  --route \
  --gmapsupp \
  --nsis \
  ./tiles/*.osm.pbf \
  ./tiles/gbasemap/motorout.TYP \
  1> ./tmp/$mkgmapfolder_"$str_Region.log 2> ./tmp/$mkgmapfolder_"$str_Region.err.log

echo "Ende : "`date`"

cd -- "$(dirname -- "$0")"
#echo "Ich bin wieder in : "$(pwd)

mv ~/osm_mrtk/tiles/gbasemap/gmapsupp.img ~/osm_mrtk/maps/$str_Region/gmapsupp.img

echo "'gmapsupp.img' [gbasemap] wurde erzeugt und kopiert nach:"
echo "/osm_mrtk/maps/$str_Region/gmapsupp.img"

# -----+
echo
echo "Step 16 gpsbabel:"
echo "'sperrung_TRACKS.GPX' zu 'sperrung.osm' konvertieren"

if [ -f ~/osm_mrtk/$sperrungfolder/sperrung.osm ];
then
  rm -rf ~/osm_mrtk/$sperrungfolder/sperrung.osm
fi

./gpsbabel/gpsbabel -i gpx -o osm \
  ~/osm_mrtk/$sperrungfolder/sperrung_TRACKS.gpx \
  ~/osm_mrtk/$sperrungfolder/sperrung_osm.tmp

# Einfügen <tag k='highway' v='motorway' />
sed -e 's/'\<'tag k='\''name'\''/'\<'tag k='\''highway'\'' v='\''motorway'\''/'\>'$'\n
'\<'tag k='\''name'\''/'g' \
  ~/osm_mrtk/$sperrungfolder/sperrung.osm.tmp > ~/osm_mrtk/$sperrungfolder/sperrung.osm

### Nächste Zeile - Einfügen <tag k='highway' v='motorway' />
# sed -e 's/'\<'tag k='\''name'\''/'\<'tag k='\''highway'\'' v='\''motorway'\''/'\>'|
'\<'tag k='\''name'\''/'g' \
#
~/osm_mrtk/$sperrungfolder/sperrung.osm > ~/osm_mrtk/$sperrungfolder/sperrung.osm.tmp
### Nächste Zeile - Ersetzen der temporären Pipe | in Zeilenumbrüche
#sed -e 's/|/'$'\n/'g' ~/osm_mrtk/$sperrungfolder/sperrung.osm.tmp > ~/osm_mrtk/
$sperrungfolder/sperrung.osm

```

```

rm -rf ~/osm_mrtk/$sperrungfolder/sperrung.osm.tmp

echo "'sperrung.osm' wurde erzeugt und angepasst."

# -----+
echo
echo "Step 17 mkgmap: [gsperrung]"

mkdir ~/osm_mrtk/tiles/gsperrung

cp ~/osm_mrtk/$stylefolder/sperrung.TYP ~/osm_mrtk/tiles/gsperrung/sperrung.TYP

echo "TYP-File kopiert nach:"
echo "/osm_mrtk/tiles/gsperrung"

echo "Jetzt baue ich Deine Karten-Kacheln für die Strecken-Sperrungen."

cd ~/osm_mrtk

echo "Start: "`date`" RAM: "$val_RAM"m"

java -Xmx$val_RAM"m" -jar ~/osm_mrtk/$mkgmapfolder/mkgmap.jar \
--output-dir=./tiles/gsperrung/ \
--max-jobs \
--style-file=./$stylefolder/sperrung_style/ \
--description="osm mrtk sperrungen "$str_Startdatum \
--country-name=Sperrungen \
--country-abbr=XX \
--family-id=$str_FID_gsperrung \
--product-id=1 \
--series-name="osm mrtk sperrungen "$str_Startdatum \
--family-name="osm mrtk sperrungen "$str_Startdatum \
--area-name=XX \
--code-page=1250 \
--latin1 \
--mapname=64240001 \
--draw-priority=20 \
--add-pois-to-areas \
--gmapsupp \
--location-autofill=0 \
--transparent \
./$sperrungfolder/sperrung.osm \
./tiles/gsperrung/sperrung.TYP \
1> ./tmp/$mkgmapfolder_"sperrung.log 2> ./tmp/$mkgmapfolder_"sperrung.err.log

echo "Ende : "`date`"

cd -- "$($dirname -- "$0")"
#echo "Ich bin wieder in : "$($pwd)

mv ~/osm_mrtk/tiles/gsperrung/gmapsupp.img \
~/osm_mrtk/$sperrungfolder/gmapsupp.img

echo "'gmapsupp.img' [gsperrung] wurde erzeugt und kopiert nach:"
echo "/osm_mrtk/sperrung/gmapsupp.img"

# -----+
echo
echo "Step 18 gmt: 2x 'gmapsupp.img' zusammenkleben"
echo "[gbasemap + gsperrung]"

./gmt/gmt -jo ~/osm_mrtk/maps/$str_Region/SD/GARMIN/gmapsupp.img \
~/osm_mrtk/maps/$str_Region/gmapsupp.img \
~/osm_mrtk/$sperrungfolder/gmapsupp.img \
1> ~/osm_mrtk/tmp/gmt.log 2> ~/osm_mrtk/tmp/gmt.err.log

echo "'gmapsupp.img' [gbasemap + gsperrung] kopiert nach:"
echo "'/osm_mrtk/maps/"$str_Region"/SD/GARMIN/gmapsupp.img'"

# -----+
echo
echo "Step 19 sperrung_POI.GPI kopieren"

cp ~/osm_mrtk/sperrung/sperrung_POI.GPI \
~/osm_mrtk/maps/$str_Region/SD/GARMIN/POI/sperrung_POI.GPI

echo "'sperrung_POI.GPI' kopiert nach:"
echo "'/osm_mrtk/maps/"$str_Region"/SD/GARMIN/POI'"

```

```

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 20 Splash-Screens kopieren"

cp ~/osm_mrtk/splash/zumo550_splash_MR_OSM_255.jpg \
~/osm_mrtk/maps/$str_Region/SD/GARMIN/zumo550_splash_MR_OSM_255.jpg
cp ~/osm_mrtk/splash/zumo660_splash_MR_OSM_255.jpg \
~/osm_mrtk/maps/$str_Region/SD/GARMIN/zumo660_splash_MR_OSM_255.jpg

echo "Splash-Screens kopiert nach:"
echo "/osm_mrtk/maps/"$str_Region"/SD/GARMIN'"

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 21 gmapi-builder"
echo "Karten fuer Garmin BaseCamp® aufbereiten"

rm -rf ~/osm_mrtk/maps/$str_Region/'osm mrtk '$str_Region' '$str_Startdatum'.gmapi'

# Info: gmapi-builder erstellt .gmapi-Datei mit dem Family- und Series-Name
./gmapi-builder/gmapi-builder.py \
-t ~/osm_mrtk/tiles/gbasemap/osmmap.tdb \
-b ~/osm_mrtk/tiles/gbasemap/osmmap.img \
-i ~/osm_mrtk/tiles/gbasemap/osmmap.mdx \
-m ~/osm_mrtk/tiles/gbasemap/osmmap_mdr.img \
-s ~/osm_mrtk/tiles/gbasemap/motorout.TYP \
~/osm_mrtk/tiles/gbasemap/*.img \
-o ~/osm_mrtk/maps/$str_Region \
1> ~/osm_mrtk/tmp/gmapi-builder.log 2> ~/osm_mrtk/tmp/gmapi-builder.err.log

echo "Karte 'osm mrtk "$str_Region" "$str_Startdatum".gmapi'"
echo "für Garmin BaseCamp® erzeugt"

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 22 Installation mit Garmin MapManager®"

echo "Ich starte jetzt den Garmin MapInstaller®"
echo "Folge bitte dessen Installations-Aufforderungen!"
echo "Beachte: Es wird immer nur eine 'osm mrtk'-Karte angezeigt."
echo "Falls nötig, bitte andere Karten vorher löschen!"

#rv=`$CD ok-msgbox --no-cancel --title "Start Garmin MapInstaller®" --icon globe \
# --text "Ich starte jetzt den Garmin MapInstaller®" \
# --informative-text "Folge bitte dessen Installations-Aufforderungen! \
# Beachte: Garmin BaseCamp® wird immer nur eine 'osm mrtk'-Karte anzeigen! Falls nötig,
bitte andere 'osm mrtk'-Karten vorher löschen!"`

#if [ "$rv" == "1" ]; then
# # Info: .gmapi-Datei hat den Family-Name
# open ~/osm_mrtk/maps/$str_Region/"osm mrtk "$str_Region" "$str_Startdatum".gmapi"
#fi

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 23 Installation auf SD-Karte"

echo "Ich öffne den Ordner der Region $str_Region."
echo "Bitte kopiere den Ordner 'GARMIN' samt Inhalt auf eine ausreichend große SD-Karte!"

#rv=`$CD ok-msgbox --no-cancel --title "Öffne Ordner "$str_Region --icon folder \
# --text "Ich öffne den Ordner der Region $str_Region." \
# --informative-text "Bitte kopiere den Ordner 'GARMIN' samt Inhalt auf eine ausreichend große
SD-Karte!" \
# --no-newline --float`

#if [ "$rv" == "1" ]; then
# open ~/osm_mrtk/maps/$str_Region/SD
#fi

# -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
echo
echo "Step 24 Aufräumen"

if [ -d ~/osm_mrtk/tmp ];
then
rm -rf ~/osm_mrtk/tmp/*.txt
rm -rf ~/osm_mrtk/tmp/*.html

#find . -type f -size 0 | xargs rm

```



```

find ~/osm_mrtk/tmp/ -type f -size 0 | xargs rm

if [ -d ~/osm_mrtk/tiles ];
then
rm -rf ~/osm_mrtk/tiles
fi

if [ -d ~/osm_mrtk/logs ];
then
rm -rf ~/osm_mrtk/logs
fi
mv ~/osm_mrtk/tmp ~/osm_mrtk/logs

echo
echo "Temporäre Dateien wurden gelöscht."
fi

# -----+
echo
echo "Step 25 Copyright-Hinweise"

echo
echo "Daten von OpenStreetMap"
echo "http://www.openstreetmap.org/"
echo "Veröffentlicht unter ODbL"
echo "http://opendatacommons.org/licenses/odbl/"
echo "Map created with mkgmap"

$CD bubble --debug --titles "MotoRoute-Tourer-Karte" \
--texts "Deine "$str_Region"-Karte ist fertig." \
--border-colors "a25f0a" \
--text-colors "000000" \
--background-tops "dfa723" \
--background-bottoms "fdde88" \
--icon-files "appIcon.icns" \
--timeout 5

rv=`$CD textbox --title "Fertig :-)" \
--informative-text "Nur noch folgende 3 Schritte ... " \
--text-from-file ./textbox_content.txt \
--button1 " OK " --no-newline --float`

if [ "$rv" == "1" ];
then
echo
echo "Deine "$str_Region"-Karte ist fertig!"
echo
echo "Präsentiert von MotoRoute - http://motoroute.de"
fi

# -----+
fileexist=0

filename=""

done

exit

```