```
#!/bin/bash
export TERM=${TERM:-dumb}
export PATH=.:$PATH
MRTKPATH=$HOME/osm_mrtk

# Mac
# Dieses Script läuft nur richtig im Paket der App!

# (Ubuntu-)Linux
# 1. Dieses Script ausführbar machen: chmod +x "MRTK_1.6.84.sh"
# 2. Java "OpenJDK-Java-7-Laufzeitumgebung" installieren:
#    sudo apt-get install openjdk-7-jre
# 3. "zlib1g-dev" installieren:
#    sudo apt-get install zlib1g-dev
# 4. "gpsbabel" installieren:
#    sudo apt-get install gpsbabel
# 5. "gmt" laden unter http://www.gmaptool.eu/en/content/linux-version
#    und in den Ordner dieses Scripts entpacken
#    dann "gmt" ausführbar machen:
#    chmod +x ./lgmt08186/gmt
# 6. optional "nsis" (Windows-Installer) installieren:
#    sudo apt-get install nsis

# Inhalt "parameter.txt"
# 1 par_Version       : 1.6.84
# 2 par_MRTK_Server   : http://motoroute.de/sites/default/files/
# 3 par_Seazip_Server : http://osm2.pleiades.uni-wuppertal.de/sea/latest/s
# 4 par_mkgmap_Server : http://www.mkgmap.org.uk/download/
# 5 par_Preprocessor  : uk.me.parabola.mkgmap.reader.osm.boundary.Boundary
# 6 par_Style         : style-r008.zip
# 7 par_Sperrung      : sperrung_2016.zip
# 8 par_convert_Server: http://m.m.i24.cc/
# 9 par_MR_Website    : http://motoroute.de/de/content/motoroute-tourer-ka
#10 par_bin           : bin-r002_0.zip (Windows)
#11 par_Version       : 1.6.84 (Windows)

# MRTK-Variablen setzen
str_Version="1.6.84"
str_Version_Datum="2012-09-09 - 2015-12-16"
# Die Family-ID wird ab Version 1.5.82 aus den rechten 3 Stellen
# des modifizierten Julianischen Datums (MJD) der .osm.pbf und einer Zufal
str_FID_gbasemap="21011"
str_FID_gsperrung="21012"
str_Parameter="http://motoroute.de/sites/default/files/parameter.txt"

# Maske aufbauen
str_Maske_01="MotoRoute-OpenStreetMap (Version "$str_Version" / "$str_Vers
str_Maske_02="Mac OS X 10.11.1 (El Capitan) - Ubuntu-Linux 14.04 (32-bit)"
```

```
str_Maske_03=""
str_Maske_04="Geo-Daten von OpenStreetMap"
str_Maske_05="Erstellt für MotoRoute.de von Gernot Skottke (motorkuh@web.c
str_Maske_06=""
str_Maske_07=""
str_Maske_08="Lade die Daten Deiner Wunsch-Region im Format '.osm.pbf' unt

# Mac
if [ "$(uname)" == "Darwin" ];
then
    str_Maske_09="http://download.geofabrik.de/"

# Linux
elif [ "$(uname)" == "Linux" ];
then
    str_Maske_09="http://download.geofabrik.de/ in den Ordner dieses Scrip
fi
str_Maske_10=""
str_Maske_11="• Land-Boundaries (bounds) werden während der Laufzeit kompi
str_Maske_12="• Option: bounds.zip manuell in Ordner 'osm_mrkt' speichern.
str_Maske_13="• Vorhandene bounds.zip werden alle 30 Tage neu geladen."
str_Maske_14="• Sea-Boundaries (sea.zip) werden automatisch alle 30 Tage r
str_Maske_15=""

# Mac
if [ "$(uname)" == "Darwin" ];
then
    str_Maske_16="Zieh die '.osm.pbf'-Datei der zu bauenden Region hier re
    str_Maske_17="Ich warte ..."
# Linux
elif [ "$(uname)" == "Linux" ];
then
    str_Maske_16=""
    str_Maske_17="Bist Du bereit?  (Abbruch mit [Ctrl]-C)"
fi
str_Maske_18=""
str_Maske_19=""


Mapbuilder()
{
# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
# Zu bauende Region ermitteln

# Mac
if [ "$(uname)" == "Darwin" ];
then
    str_FileExt="$filename"
```

```

# Linux
elif [ "$(uname)" == "Linux" ];
then
    echo "MRTK: Verwende zuerst gefundenen Dateinamen mit Endung '.osm.pbf
    ls | grep ".osm.pbf$" > $pathname/region.txt
    str_FileExt=$(head -n 1 $pathname/region.txt)
    rm -rf $pathname/region.txt
    if [ ! -f ./$str_FileExt ];
    then
        MissingOsmPbf
    fi
fi
#echo "MRTK: Dateiname mit  Extension: "$str_FileExt

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

# Name (ohne "-latest") und Kurzname der Region festlegen
str_File="${str_FileExt%%.*}"
# echo "MRTK: Dateiname ohne Extension: "$str_File

str_Region=${str_File%-*}
# echo "MRTK: Region                  : "$str_Region

# Region in Teil-Worte zerlegen (rheinland-pfalz in $1:rheinland, $2:pfalz
IFS="-"
set - $str_Region
if [ "${#2}" -eq 0 ];
then
    str_RegionLaenge=${#str_Region}
    if [ "${str_RegionLaenge}" -gt 2 ];
    then
        str_Region_kurz=${str_Region:0:3} # bayern = bay
    else
        str_Region_kurz=$str_Region
    fi
else
    str_Region_kurz=${1:0:1}${2:0:1} # rheinland-pfalz = rp
fi

IFS=" "
# klein in GROSS Buchstaben
str_Region=`echo "$str_Region" | tr '[:lower:]' '[:upper:]'`
str_Region_kurz=`echo "$str_Region_kurz" | tr '[:lower:]' '[:upper:]'`

```

```
echo " "
echo "MRTK: Zu bauende Region: "$str_Region "("$str_Region_kurz")"

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
# Zeitstempel '.osm.pbf'-Datei

str_Region_Date=$("$osmconvert" "$pathname/$str_FileExt" --out-timestamp)
str_Region_Date=${str_Region_Date:0:10}

echo " "
echo "MRTK: '"$str_FileExt"': "$str_Region_Date

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
# Family-ID
if [ "$conf_FIDb" == "" ];
then
    # Gregorian to Julian Day Number
    gtojdn ${str_Region_Date:8:2} ${str_Region_Date:5:2} ${str_Region_Date
    typeset -i num
    num=$(($RANDOM % 9+1))
    str_FID_gbasemap=$mjd$num
    # echo str_FID_gbasemap : $str_FID_gbasemap
    num=$num-1
    str_FID_gsperrung=$mjd$num
    # echo str_FID_gsperrung: $str_FID_gsperrung
else
    str_FID_gbasemap=$conf_FIDb
    # echo str_FID_gbasemap : $str_FID_gbasemap
    str_FID_gsperrung=$conf_FIDx
    # echo str_FID_gsperrung: $str_FID_gsperrung
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
# Prüfung Ordner /maps/

echo " "
if [ -d $MRTKPATH/maps ];
then
    echo "MRTK: Ordner /maps/ vorhanden"
else
    mkdir $MRTKPATH/maps/
    echo "MRTK: Ordner /maps/ angelegt"
fi

if [ -d $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasemap
then
    rm -rf $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasem
fi
```

```

mkdir $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasemap
mkdir $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasemap/SD
mkdir $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasemap/SD
mkdir $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasemap/SD
mkdir $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasemap/SD

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
# 'style' ermitteln und laden

style=$par_Style
stylefolder=$(basename $style .zip)

echo " "
if [ -d $MRTKPATH/$stylefolder ];
then
    echo "MRTK: Style '"$style"' aktuell"
else
    rm -rf $MRTKPATH/style*
    ./wget -c $par_MRTK_Server$style -O $MRTKPATH/$style

    unzip -q $MRTKPATH/$style -d $MRTKPATH
    if [ -f $MRTKPATH/$style ];
    then
        rm -rf $MRTKPATH/$style
    fi
    echo "MRTK: Style '"$style"' geladen und entpackt"
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
# 'sperrung' ermitteln und laden

sperrung=$par_Sperrung
sperrungfolder=$(basename $sperrung .zip)

echo " "
if [ -d $MRTKPATH/$sperrungfolder ];
then
    echo "MRTK: Sperrung '"$sperrung"' aktuell"
else
    rm -rf $MRTKPATH/sperrung*
    ./wget -c $par_MRTK_Server$sperrung -O $MRTKPATH/$sperrung
    unzip -q $MRTKPATH/$sperrung -d $MRTKPATH
    if [ -f $MRTKPATH/$sperrung ];
    then
        rm -rf $MRTKPATH/$sperrung
    fi
    echo "MRKT: Sperrung '"$sperrung"' geladen und entpackt"
```

```
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
# 'splitter' ermitteln und laden

./wget -L $par_mkgmap_Server"splitter.html" -q -O $MRTKPATH/tmp/splitter.h

grep -o 'splitter-r[0-9]*.zip' $MRTKPATH/tmp/splitter.html > $MRTKPATH/tmp
splitter=$(sed -n '1p' $MRTKPATH/tmp/splitter.txt)
splitterfolder=$(basename $splitter .zip)

echo " "
if [ -d $MRTKPATH/$(basename $splitter .zip) ];
then
    echo "MRTK: '"$(basename $splitter .zip)"' aktuell"
else
    rm -rf $MRTKPATH/splitter-r*
    # echo "MRTK: 'splitter' gelöscht."
    ./wget -c $par_mkgmap_Server$splitter -O $MRTKPATH/$splitter
    unzip -q $MRTKPATH/$splitter -d $MRTKPATH
    if [ -f $MRTKPATH/$splitter ];
    then
        rm -rf $MRTKPATH/$splitter
    fi
    echo "MRTK: '" $splitter"' geladen und entpackt"
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
# 'mkgmap' ermitteln und laden"

./wget -L $par_mkgmap_Server"mkgmap.html" -q -O $MRTKPATH/tmp/mkgmap.html

grep -o 'mkgmap-r[0-9]*.zip' $MRTKPATH/tmp/mkgmap.html > $MRTKPATH/tmp/mkg
mkgmap=$(sed -n '1p' $MRTKPATH/tmp/mkgmap.txt)
mkgmapfolder=$(basename $mkgmap .zip)

echo " "
if [ -d $MRTKPATH/$(basename $mkgmap .zip) ];
then
    echo "MRTK: '"$(basename $mkgmap .zip)"' aktuell"
else
    rm -rf $MRTKPATH/mkgmap-r*
    # echo "MRTK: 'mkgmap' gelöscht."
    ./wget -c $par_mkgmap_Server$mkgmap -O $MRTKPATH/$mkgmap
    unzip -q $MRTKPATH/$mkgmap -d $MRTKPATH
    if [ -f $MRTKPATH/$mkgmap ];
    then
        rm -rf $MRTKPATH/$mkgmap
```

```
    fi
    echo "MRTK '"$mkgmap"' geladen und entpackt"
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# gpsbabel

echo " "
echo "MRTK: gpsbabel: 'sperrung_TRACKS.gpx' konvertieren"

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

"$gpsbabel" \
    -i gpx -o osm \
    $MRTKPATH/$sperrungfolder/sperrung_TRACKS.GPX \
    $MRTKPATH/$sperrungfolder/sperrung.osm.tmp

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

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

rm -rf $MRTKPATH/$sperrungfolder/sperrung.osm.tmp

echo "MRTK: 'sperrung.osm' erzeugt und angepasst"

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Sea-Boundaries 'sea.zip' laden

ZipAge "sea"

if [ "`echo "$zip_age < 100" | bc -l`" = "1" ] ;
then
    echo "MRTK: Geladene 'sea.zip' wird verwendet"
else
    if [ "`echo "$zip_age == 999" | bc -l`" = "1" ] ;
    then
        echo "MRTK: 'sea.zip' wird geladen"
    else
        echo "MRTK: Geladene 'sea.zip' zu alt - wird neu geladen"
```

```
        rm -rf $MRTKPATH/sea.zip
    fi
    ./wget -c $par_Seazip_Server -O $MRTKPATH/sea.zip
    echo "MRTK: 'sea.zip' geladen"
    ZipAge
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Boundaries kompilieren

if ! [ -f $MRTKPATH/bounds.zip ];
then

    echo " "
    echo "MRTK: Boundaries "$str_Region" kompilieren"

    cd $MRTKPATH/tmp/

    echo "MRTK: 1. Start osmconvert: "`date +%d.%m.%Y\ %H:%M:%S`
    "$osmconvert" \
        "$pathname/$str_FileExt" \
        -o=$str_Region.o5m \
        1> $MRTKPATH/tmp/osmconvert_$str_Region.log 2> $MRTKPATH/tmp/osmc
    echo "MRTK:    Ende  osmconvert: "`date +%d.%m.%Y\ %H:%M:%S`
    echo "MRTK: "$str_Region".o5m erzeugt"

    echo " "
    echo "MRTK: 2. Start osmfilter: "`date +%d.%m.%Y\ %H:%M:%S`
    "$osmfilter" \
            $MRTKPATH/tmp/$str_Region.o5m \
            --keep-nodes= \
            --keep-ways-relations="boundary=administrative =postal_code p
            -o=$str_Region-boundaries.o5m \
            1> $MRTKPATH/tmp/osmfilter_$str_Region.log 2> $MRTKPATH/tmp/c
    echo "MRTK:    Ende  osmfilter: "`date +%d.%m.%Y\ %H:%M:%S`
    echo "MRTK:" $str_Region"-boundaries.o5m erzeugt"

    # Freies RAM neu zuteilen
    RAMfree

    echo " "
    echo "MRTK: 3. Start Boundary-Preprocessor: "`date +%d.%m.%Y\ %H:%M:%
    java -Xmx$val_RAM"m" -cp $MRTKPATH/$mkgmapfolder/mkgmap.jar \
        $par_Preprocessor \
        $MRTKPATH/tmp/$str_Region-boundaries.o5m \
        $MRTKPATH/bounds \
        1> $MRTKPATH/tmp/$mkgmapfolder"_BoundaryPreprocessor_"$str_Region
    echo "MRTK:    Ende  Boundary-Preprocessor: "`date +%d.%m.%Y\ %H:%M:%
```

```
    echo "MRTK: Boundaries "$str_Region" erzeugt"

    rm -rf $MRTKPATH/tmp/*.o5m
    echo "MRTK: Temporäre '.o5m'-Dateien gelöscht"

    bounds_zip=""

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# ODER bei vorhandener 'bounds.zip' wie 'sea.zip' ueberpruefen und alle 3
else
    ZipAge "bounds"

    if [ "`echo "$zip_age < 100" | bc -l`" = "1" ] ;
    then
        echo "MRTK: Geladene 'bounds.zip' wird verwendet"
    else
        if [ "`echo "$zip_age == 999" | bc -l`" = "1" ] ;
        then
            echo "MRTK: 'bounds.zip' wird geladen"
        else
            echo "MRTK: Geladene 'bounds.zip' zu alt - wird neu geladen"
            rm -rf $MRTKPATH/bounds.zip
        fi
        ${par_Seazip_Server//sea/bounds}
        ./wget -c ${par_Seazip_Server//sea/bounds} -O $MRTKPATH/bounds.zi
        echo "MRTK: 'bounds.zip' geladen"
        ZipAge
    fi
    bounds_zip=".zip"

fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# splitter

rm -rf $MRTKPATH/tiles
mkdir $MRTKPATH/tiles

# Freies RAM neu zuteilen
RAMfree

echo " "
echo "MRTK: splitte '"$str_FileExt"'"
echo "MRTK: Start splitter: "`date +%d.%m.%Y\ %H:%M:%S`" (RAM: "$val_RAM"

cd $MRTKPATH/

if [ -f $MRTKPATH/areas.list ];
```

```
433 then
434     cp $MRTKPATH/areas.list $MRTKPATH/tiles/areas.list
435     str_splitfile="--split-file=areas.list"
436     #echo $str_splitfile
437     echo "MRTK: 'areas.list' wird verwendet"
438 else
439     str_splitfile=""
440 fi
441
442 java -Xmx$val_RAM"m" -jar $MRTKPATH/$splitterfolder/splitter.jar \
443     $str_splitfile \
444     --max-nodes=1600000 \
445     --output-dir=./tiles \
446     "$pathname/$str_FileExt" \
447     > $MRTKPATH/tmp/$splitterfolder.log
448
449 # cd -- "$(dirname -- "$0")"
450 # echo "Ich bin wieder in : "$(pwd)
451
452 echo "MRTK: Ende  splitter: "`date +%d.%m.%Y\ %H:%M:%S`
453
454 # ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
455 # mkgmap: gbasemap 'gmapsupp.img'"
456
457 echo " "
458 echo "MRTK: baue Karten-Kacheln "$str_Region
459
460 mkdir $MRTKPATH/tiles/gbasemap
461
462 #cp $MRTKPATH/$stylefolder/motorout.TYP $MRTKPATH/tiles/gbasemap/motorout
463 cp $MRTKPATH/$stylefolder/motorout.txt $MRTKPATH/tiles/gbasemap/motoroute
464 echo "MRTK: TYP-File kopiert nach '/osm_mrtk/tiles/gbasemap'"
465
466 sed 's/FID=21011/FID='$str_FID_gbasemap'/' \
467     $MRTKPATH/tiles/gbasemap/motoroute.txt > $MRTKPATH/tiles/gbasemap/mot
468 rm -rf $MRTKPATH/tiles/gbasemap/motoroute.txt
469 echo "MRTK: Family-ID (FID) im TYP-File geändert in $str_FID_gbasemap"
470
471 rm -rf $MRTKPATH/tiles/gbasemap/gmapsupp
472
473 cd $MRTKPATH
474
475 if [ $str_Region = "EUROPE" ];
476 then
477     str_gmapsupp=""
478 else
479     str_gmapsupp="--gmapsupp"
480 fi
```

```

# Freies RAM neu zuteilen
RAMfree

echo "MRTK: Start mkgmap: "`date +%d.%m.%Y\ %H:%M:%S`" (RAM: "$val_RAM"m)

# Alternative:  --generate-sea:multipolygon,extend-sea-sectors,close-gaps

java -Xmx$val_RAM"m" -jar $MRTKPATH/$mkgmapfolder/mkgmap.jar \
    --bounds=./bounds$bounds_zip \
    --precomp-sea=./sea.zip \
    --output-dir=./tiles/gbasemap/ \
    --max-jobs \
    --style-file=./$stylefolder/motorout_style/ \
    --description="OSM MRTK "$str_Region" "$str_Region_Date" "$str_FID_gb
    --country-name=$str_Region \
    --country-abbr=$str_Region_kurz \
    --family-id=$str_FID_gbasemap \
    --product-id=1 \
    --series-name="OSM MRTK "$str_Region" "$str_Region_Date" "$str_FID_gb
    --family-name="OSM MRTK "$str_Region" "$str_Region_Date" "$str_FID_gb
    --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 \
    $conf_nsis \
    $str_gmapsupp \
    ./tiles/*.osm.pbf \
    ./tiles/gbasemap/motorout.txt \
    1> ./tmp/$mkgmapfolder"_"$str_Region.log 2> ./tmp/$mkgmapfolder"_"$st

echo "MRTK: Ende  mkgmap: "`date +%d.%m.%Y\ %H:%M:%S`

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

if [ $str_Region = "EUROPE" ];
```

```
529 then
530     echo "MRTK: Vollständige Abdeckung für EUROPE größer als 4 GB!"
531     echo "MRTK: Bitte verwende Garmin MapInstall®"
532 else
533     mv $MRTKPATH/tiles/gbasemap/gmapsupp.img $MRTKPATH/maps/$str_Region"_
534     echo "MRTK: gbasemap 'gmappsupp.img' erzeugt und kopiert nach:"
535     echo "MRTK: '/osm_mrtk/maps/"$str_Region"_"$str_Region_Date"_"$str_FI
536 fi
537
538 # ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
539 # mkgmap: gsperrung
540
541 echo " "
542 echo "MRTK: baue Karten-Kacheln für Strecken-Sperrungen"
543
544 mkdir $MRTKPATH/tiles/gsperrung
545
546 #cp $MRTKPATH/$stylefolder/sperrung.TYP $MRTKPATH/tiles/gsperrung/sperrur
547 cp $MRTKPATH/$stylefolder/sperrung.txt $MRTKPATH/tiles/gsperrung/sperrung
548 echo "MRTK: TYP-File kopiert nach '/osm_mrtk/tiles/gsperrung'"
549
550 sed 's/FID=21012/FID='$str_FID_gsperrung'/' \
551     $MRTKPATH/tiles/gsperrung/sperrungen.txt > $MRTKPATH/tiles/gsperrung/
552 rm -rf $MRTKPATH/tiles/gsperrung/sperrungen.txt
553 echo "MRTK: Family-ID (FID) im TYP-File geändert in $str_FID_gsperrung"
554
555 # Freies RAM neu zuteilen
556 RAMfree
557
558 cd $MRTKPATH
559
560 echo "MRTK: Start mkgmap: "`date +%d.%m.%Y\ %H:%M:%S`" (RAM: "$val_RAM"m)
561
562 java -Xmx$val_RAM"m" -jar $MRTKPATH/$mkgmapfolder/mkgmap.jar \
563     --output-dir=./tiles/gsperrung/ \
564     --max-jobs \
565     --style-file=./$stylefolder/sperrung_style/ \
566     --description="osm mrtk sperrungen "$str_Startdatum" "$str_FID_gsperr
567     --country-name=Sperrungen \
568     --country-abbr=XX \
569     --family-id=$str_FID_gsperrung \
570     --product-id=1 \
571     --series-name="osm mrtk sperrungen "${sperrungfolder: -8}" "$str_FID_
572     --family-name="osm mrtk sperrungen "${sperrungfolder: -8}" "$str_FID_
573     --area-name=XX \
574     --code-page=1250 \
575     --latin1 \
576     --mapname=64240001 \
```

```
    --draw-priority=20 \
    --add-pois-to-areas \
    --gmapsupp \
    --location-autofill=0 \
    --transparent \
    ./$sperrungfolder/sperrung.osm \
    ./tiles/gsperrung/sperrung.txt \
    1> ./tmp/$mkgmapfolder"_"sperrung.log 2> ./tmp/$mkgmapfolder"_"sperru

echo "MRTK: Ende  mkgmap: "`date +%d.%m.%Y\ %H:%M:%S`

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

mv $MRTKPATH/tiles/gsperrung/gmapsupp.img \
    $MRTKPATH/$sperrungfolder/gmapsupp.img

echo "MRTK: gsperrung 'gmapsupp.img' erzeugt und kopiert nach"
echo "MRTK: '/osm_mrtk/sperrung/gmapsupp.img'"

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# gmt: 2x 'gmapsupp.img' zusammenkleben

echo " "
echo "MRTK: gmt: gbasemap + gsperrung"

if [ "$str_Region" = "EUROPE" ];
then
    echo "MRTK: Vollständige Abdeckung für EUROPE größer als 4 GB!"
    echo "MRTK: Bitte verwende Garmin MapInstall®"
else
    "$gmt" \
        -jo $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbas
        $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasemap
        $MRTKPATH/$sperrungfolder/gmapsupp.img \
        1> $MRTKPATH/tmp/gmt.log  2> $MRTKPATH/tmp/gmt.err.log

    echo "MRTK: gbasemap + gsperrung 'gmapsupp.img' kopiert nach"
    echo "MRTK: '/osm_mrtk/maps/"$str_Region"_"$str_Region_Date"_"$str_FI
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# sperrung_POI.gpi kopieren

cp $MRTKPATH/$sperrungfolder/sperrung_POI.gpi \
    $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasemap/SD/

echo " "
```

```
echo "MRTK: 'sperrung_POI.GPI' kopiert nach"
echo "MRTK: '/osm_mrtk/maps/"$str_Region"_"$str_Region_Date"_"$str_FID_gb

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# gmapi-builder (wenn in Konfigurations-Datei Zeile 4 eingetragen ODER Ma

if ! [ -z "$gmapibuilder" ] || [ "$(uname)" == "Darwin" ];
then
    echo " "
    echo "MRTK: Karten für Garmin BaseCamp® (Mac) aufbereiten"

    rm -rf $MRTKPATH/maps/$str_Region/'osm mrtk '$str_Region' '$str_Start

    # Wichtiger Hinweis:
    # Versionen des gmapi-builder unter
    # https://bitbucket.org/berteun/gmapibuilder/downloads/gmapi-builder.
    # oder
    # http://svn.mkgmap.org.uk/mkgmap/trunk/scripts/gmapi-builder.py
    # sind ohne Option -i und können keinen Adress-Index aufbauen!!!
    #
    # Zum modifizierten gmapi-builder gibt es hier Informationen:
    # http://comments.gmane.org/gmane.comp.gis.openstreetmap.mkgmap.devel
    # Ehemals Download unter:
    # http://cache.gmane.org//gmane/comp/gis/openstreetmap/mkgmap/devel/3
    # Modifizierte Version Download unter:
    # http://motoroute.de/sites/default/files/gmapi-builder-index-py.txt
    # umbenennen in gmapi-builder.py
    #
    # Standard-Versionen des gmapi-builder wie folgt modifizieren:
    # suchen nach:
    # help="the name of the base img")
    #
    # folgenden Code unter die gefundene Zeile einfügen:
    #
    #    oparser.add_option("-i", "--idxfile", dest="mdx",
    #            help="the name of this mapset's index (mdx) file")
    #    oparser.add_option("-m", "--mdrfile", dest="mdr",
    #            help="the name of this mapset's MDR file")
    #
    # Info: gmapi-builder erstellt .gmapi-Datei mit dem Family- und Serie

    "$gmapibuilder" \
        -t $MRTKPATH/tiles/gbasemap/osmmap.tdb \
        -b $MRTKPATH/tiles/gbasemap/osmmap.img \
        -i $MRTKPATH/tiles/gbasemap/osmmap.mdx \
        -m $MRTKPATH/tiles/gbasemap/osmmap_mdr.img \
        -s $MRTKPATH/tiles/gbasemap/motorout.typ \
        $MRTKPATH/tiles/gbasemap/*.img \
```

```
        -o $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbase
        1> $MRTKPATH/tmp/gmapi-builder.log  2> $MRTKPATH/tmp/gmapi-builde

    echo "MRTK: 'osm mrtk "$str_Region" "$str_Startdatum" "$str_FID_gbase

fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# pimp_nsis

if [ "$conf_nsis" = "--nsis" ];
then
    echo
    #echo "MRTK: NSIS-Konfiguration 'osmmap.nsi' anpassen"
    # Installation der Kartenkacheln in Nsis-Installer entfernen
    grep -i -v "File .6324[0-9]*.img" $MRTKPATH/tiles/gbasemap/osmmap.nsi
    # Section in .nsi-Script ergaenzen, um Kartenkacheln zu kopieren
    # Section -MapInstall
    # CopyFiles /SILENT /FILESONLY "$EXEDIR\..\..\tiles\gbasemap\6324*.im
    # SectionEnd
    echo >> $MRTKPATH/tiles/gbasemap/osmmap_2.nsi
    echo Section "MapInstall" >> $MRTKPATH/tiles/gbasemap/osmmap_2.nsi
    echo "  "CopyFiles /SILENT /FILESONLY "\$EXEDIR\gbasemap\6324*.img" "
    echo SectionEnd >> $MRTKPATH/tiles/gbasemap/osmmap_2.nsi

    # Mac
    if [ "$(uname)" == "Darwin" ];
    then
        # Bitmap, Icon, Kopf in nsis-Konfiguration einspielen
        bmp="$(dirname -- "$0")/nsis/MRTKNsis.bmp"
        cp "$bmp" $MRTKPATH/tiles/gbasemap/MRTKNsis.bmp

        ico="$(dirname -- "$0")/nsis/MRTKIcon.ico"
        cp "$ico" $MRTKPATH/tiles/gbasemap/MRTKIcon.ico

        nsis1="$(dirname -- "$0")/nsis/osmmap_1.nsi"
        cp "$nsis1" $MRTKPATH/tiles/gbasemap/osmmap_1.nsi
    # Linux
    elif [ "$(uname)" == "Linux" ];
    then
        echo '!define MUI_WELCOMEPAGE_TITLE "MotoRoute-OpenStreetMap"' >>
        echo '!define MUI_FINISHPAGE_TITLE "MotoRoute-OpenStreetMap"' >>
    fi
    rm -rf $MRTKPATH/tiles/gbasemap/osmmap.nsi
    cat $MRTKPATH/tiles/gbasemap/osmmap_1.nsi $MRTKPATH/tiles/gbasemap/os
    rm -rf $MRTKPATH/tiles/gbasemap/osmmap_*.nsi
    echo "MRTK: NSIS-Konfiguration 'osmmap.nsi' angepasst"

```

```
# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# makensis

    # Mac
    if [ "$(uname)" == "Darwin" ];
    then
        # Wine.app installiert
        if [ -d /Applications/Wine.app ];
        then
            WINEPATH=/Applications/Wine.app/Contents/Resources/bin/wine
        fi
        MAKENSISPATH="$(dirname -- "$0")/nsis-2.46/makensis.exe"
        echo "MRTK: erzeuge Windows-Installer für Garmin BaseCamp©"
        echo "MRTK: Start makensis: "`date +%d.%m.%Y\ %H:%M:%S`
        "$WINEPATH" "$MAKENSISPATH" "Z:/$MRTKPATH/tiles/gbasemap/osmmap.r
            1> $MRTKPATH/tmp/makensis.log 2> $MRTKPATH/tmp/makensis.err.l
    # Linux
    elif [ "$(uname)" == "Linux" ];
    then
        # MAKENSIS installiert=?
        if ! [ -z "echo $(which makensis)" ];
        then
            MAKENSISPATH=$(which makensis)
            echo "MRTK: erzeuge Windows-Installer für Garmin BaseCamp©"
            echo "MRTK: Start makensis: "`date +%d.%m.%Y\ %H:%M:%S`
            "$MAKENSISPATH" $MRTKPATH/tiles/gbasemap/osmmap.nsi \
                1> $MRTKPATH/tmp/makensis.log  2> $MRTKPATH/tmp/makensis.
        fi
    fi
    if ! [ "$MAKENSISPATH" == "" ];
    then
        echo "MRTK: Ende  makensis: "`date +%d.%m.%Y\ %H:%M:%S`
        mv "$MRTKPATH/tiles/gbasemap/OSM MRTK $str_Region $str_Region_Dat
            $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbas
        echo "MRTK: Windows-Installer für Garmin BaseCamp© erzeugt"
    fi
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# gbasemap

# Linux (generell)
# ODER
# QLandkarte GT.app vorhanden? (Mac)
# ODER
# Nsis-Installer soll erzeugt werden (optional mrtkconf.txt)

if [ "$(uname)" == "Linux" ] || [ -d /Applications/QLandkarte\ GT.app ] |
```

```
then
    echo " "
    echo "MRTK: gbasemap - QLandkarte GT® - http://www.qlandkarte.org"
    mv $MRTKPATH/tiles/gbasemap $MRTKPATH/maps/$str_Region"_"$str_Region_
    # Nsis-Installer soll erzeugt werden (optional mrtkconf.txt; Linux ge
    if [ "$conf_nsis" = "--nsis" ];
    then
        echo "MRTK: gbasemap - Karten-Kacheln für Windows-Installer"
    fi
    echo "MRTK: verschoben nach "
    echo "MRTK: '$MRTKPATH/maps/"$str_Region"_"$str_Region_Date"_"$str_FI
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Installation auf SD-Karte

echo " "
if [ "$str_Region" = "EUROPE" ];
then
    echo "MRTK: Vollständige Abdeckung für Region 'europe' größer als 4 G
    echo "MRTK: Bitte verwende Garmin MapInstall®."
else
    echo "MRTK: Bitte kopiere den Ordner "
    echo "MRTK: '$MRTKPATH/maps/"$str_Region"_"$str_Region_Date"_"$str_FI
    echo "MRTK: auf eine ausreichend große SD-Karte!"

#   # Mac
#   if [ "$(uname)" == "Darwin" ];
#   then
#       # say "Bitte kopiere den Ordner GARMIN auf eine ausreichend große
#       # osascript -e 'tell app "Terminal" to display dialog "Bitte kopi
#
#       rv=`$CD ok-msgbox --no-cancel --title "MotoRoute-OpenStreetMap" -
#           --text "$str_Region $str_Region_Date (Family-ID $str_FID_gbas
#           --informative-text "Bitte kopiere den Ordner '$MRTKPATH/maps/
#           --float --timeout 30`
#
#       # open $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_g
#
#   # Linux
#   elif [ "$(uname)" == "Linux" ];
#   then
#       xmessage "Bitte kopiere den Ordner '$MRTKPATH/maps/"$str_Region"_
#   fi
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Cleanup
```

```

if [ -d $MRTKPATH/tmp ];
then
    # Linux
    if [ "$(uname)" == "Linux" ];
    then
        rm -rf ./wget
    fi

    # Diese zu löschende Meldung erscheint im 'makensis.err.log':
    # Wine cannot find the FreeType font library.  To enable Wine to
    # use TrueType fonts please install a version of FreeType greater tha
    # or equal to 2.0.5.
    # http://www.freetype.org
    rm -rf $MRTKPATH/tmp/makensis.err.log

    rm -rf $MRTKPATH/tmp/*.txt
    rm -rf $MRTKPATH/tmp/*.html
    rm -rf $MRTKPATH/tmp/*.o5m

    find $MRTKPATH/tmp/ -type f -size 0 | xargs rm

    if [ -d $MRTKPATH/tiles ];
    then
        rm -rf $MRTKPATH/tiles
    fi

    if [ -d $MRTKPATH/bounds ];
    then
        rm -rf $MRTKPATH/bounds
    fi

    if [ -d $MRTKPATH/logs ];
    then
        rm -rf $MRTKPATH/logs
    fi

    mv $MRTKPATH/tmp $MRTKPATH/logs

    echo " "
    echo "MRTK: Temporäre Dateien gelöscht"
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Copyright/Hinweise

echo " "
echo "MRTK: Daten von OpenStreetMap"
```

```
echo "MRTK: http://www.openstreetmap.org/"
echo "MRTK: Veröffentlicht unter ODbL"
echo "MRTK: http://opendatacommons.org/licenses/odbl/"
echo "MRTK: Map created with mkgmap"

echo " "
echo "MRTK: Deine "$str_Region"-Karte ist fertig!"
echo "MRTK: Präsentiert von MotoRoute - http://motoroute.de"

# Mac
if [ "$(uname)" == "Darwin" ];
then
    # Maps-Ordner öffnen
    open $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbasema
    # gmapi-Datei öffnen
    # open $MRTKPATH/maps/$str_Region"_"$str_Region_Date"_"$str_FID_gbase

    # say "Deine $str_Region-Karte ist fertig! - Präsentiert von MotoRout
    # osascript -e 'tell app "Terminal" to display dialog "Deine '$str_Re

    $CD bubble --debug --titles "MotoRoute-OpenStreetMap" \
        --texts "Deine "$str_Region"-Karte ist fertig." \
        --border-colors "a25f0a" \
        --text-colors "000000" \
        --background-tops "dfa723" \
        --background-bottoms "fdde88" \
        --icon-files "appIcon.icns" \
        --timeout 30



# Linux
elif [ "$(uname)" == "Linux" ];
then
    xmessage "Deine "$str_Region"-Karte ist fertig! - Praesentiert von Mo
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
fileexist=0

filename=""

exit
}


ZipAge()
{
```

```
913 if [ -f $MRTKPATH/$1.zip ];
914 then
915     unzip -q -o $MRTKPATH/$1.zip version.txt -d $MRTKPATH/tmp
916     mv $MRTKPATH/tmp/version.txt $MRTKPATH/tmp/$1.txt
917     zip=$(sed -n '1p' $MRTKPATH/tmp/$1.txt)
918     echo
919     echo "MRTK: '"$1".zip': "$zip
920     # '$1.zip' älter als 30 Tage?
921     if [ "`echo "$zip < $str_JMT" | bc -l`" = "1" ] ;
922     then
923         zip_age=$[$str_JMT-$zip]
924     fi
925 else
926     # '$1.zip' nicht vorhanden!
927     zip_age=999
928 fi
929 }
930
931
932
933 RAMfree()
934 {
935 # Freies RAM neu zuteilen
936 # Mac
937 if [ "$(uname)" == "Darwin" ];
938 then
939     free="$(( $(vm_stat | grep free | awk '{ print $3 }' | sed 's/\.//')*
940     inactive="$(( $(vm_stat | grep inactive | awk '{ print $3 }' | sed 's
941     let val_RAM=$free+$inactive
942 # Linux
943 elif [ "$(expr substr $(uname -s) 1 5)" == "Linux" ];
944 then
945     # Erzeugt der Befehl "free -mt" die Ausgabe "Speicher:" oder "Mem:"?
946     if [ $(free -mt | awk '/^Speicher:/{print $1}') ];
947     then
948         val_RAM=$(free -mt | awk '/^Speicher:/{print $4}')
949         # Tests in Version 1.5.83 liefen nur mit 70% des freien RAM
950         val_RAM=$(( $val_RAM * 7/10 ))
951     elif [ $(free -mt | awk '/^Mem:/{print $1}') ];
952     then
953         val_RAM=$(free -mt | awk '/^Mem:/{print $4}')
954         # Tests in Version 1.5.83 liefen nur mit 70% des freien RAM
955         val_RAM=$(( $val_RAM * 7/10 ))
956     fi
957 fi
958 return $val_RAM
959 }
960
```

```


MissingOsmPbf()
{
echo " "
echo "  ################################################################
echo "  ### Vermisse '.osm.pbf'-Datei im Verzeichnis des Scripts.      ##
echo "  ### Ausführung wird beendet.                                   ##
echo "  ###                                                           ##
echo "  ### Bitte lade zuerst Deine Wunsch-Region im Format '.osm.pbf' ##
echo "  ### unter http://download.geofabrik.de/                       ##
echo "  ################################################################
echo " "

# Mac
if [ "$(uname)" == "Darwin" ];
then
    # say "Vermisse osm.pbf-Datei. Ausfuehrung wird beendet."
    osascript -e 'tell app "Terminal" to display dialog "Vermisse '.osm.p

# Linux
elif [ "$(uname)" == "Linux" ];
then
    xmessage "Vermisse '.osm.pbf'-Datei. Ausfuehrung wird beendet" -nearm
fi
}



gtojdn()
{
# https://blog.sleeplessbeastie.eu/2013/05/17/how-to-convert-date-to-juli
# gtojdn
# convert Gregorian calendar date to Julian Day Number
#
# parameters:
# day
# month
# year
#
# example:
# gtojdn 15 5 2013
#
# notes:
# algorithm is simplified
# as it will return 2456428 instead of 2456427.5
if [ $2 -le 2 ]; then
    y=$(($3 - 1))
```

```
    m=$(($2 + 12))
else
    y=$3
    m=$2
fi
d=$1

x=$(echo "2 - $y / 100 + $y / 400" | bc)
x=$(echo "($x + 365.25 * ($y + 4716))/1" | bc)
x=$(echo "($x + 30.6001 * ($m + 1))/1" | bc)
jd=$(echo "($x + $d - 1524.5 + 0.5)/1" | bc) # added 0.5
mjd=${jd:4:3}
return $mjd
}


clear

# Mac
# if [ "$(uname)" == "Darwin" ];
# then
#   say "Hallo $USER"
# fi

cd -- "$(dirname -- "$0")"

# Mac
if [ "$(uname)" == "Darwin" ];
then
    CD="CocoaDialog.app/Contents/MacOS/CocoaDialog"

    if ! [ $(basename -- "$0") = 'script' ];
    then
        clear
        echo "Dieses Script kann nur in der {MotoRoute-OpenStreetMap.app}
        exit
    fi

    if [ $filename == ""];
    then
        if [ ! "$@" = '' ];
        then
            filename=$(basename "$@")
            pathname=$(dirname "$@")
        else
            echo "$str_Maske_01 "
            echo "$str_Maske_02 "
            echo "$str_Maske_03 "
```

```
            echo "$str_Maske_04 "
            echo "$str_Maske_05 "
            echo "$str_Maske_06 "
            echo "$str_Maske_07 "
            echo "$str_Maske_08 "
            echo "$str_Maske_09 "
            echo "$str_Maske_10 "
            echo "$str_Maske_11 "
            echo "$str_Maske_12 "
            echo "$str_Maske_13 "
            echo "$str_Maske_14 "
            echo "$str_Maske_15 "
            echo "$str_Maske_16 "
            echo "$str_Maske_17 "
            echo "$str_Maske_18 "
            echo "$str_Maske_19 "

            # Mac
            # if [ "$(uname)" == "Darwin" ];
            # then
            #   say "Bist Du bereit?"
            # fi
        fi
    fi

# Linux
elif [ "$(uname)" == "Linux" ];
then
    pathname=$(dirname "$0")
    echo "$str_Maske_01 "
    echo "$str_Maske_02 "
    echo "$str_Maske_03 "
    echo "$str_Maske_04 "
    echo "$str_Maske_05 "
    echo "$str_Maske_06 "
    echo "$str_Maske_07 "
    echo "$str_Maske_08 "
    echo "$str_Maske_09 "
    echo "$str_Maske_10 "
    echo "$str_Maske_11 "
    echo "$str_Maske_12 "
    echo "$str_Maske_13 "
    echo "$str_Maske_14 "
    echo "$str_Maske_15 "
    echo "$str_Maske_16 "
    echo "$str_Maske_17 "
    echo "$str_Maske_18 "
    echo "$str_Maske_19 "
```

```
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Prüfung Ordner /osm_mrtk/

if [ -d $MRTKPATH ];
then
    echo "MRTK: Ordner /osm_mrtk/ vorhanden"
else
    mkdir $MRTKPATH
    echo "MRTK: Ordner /osm_mrtk/ angelegt"
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Prüfung Ordner /tmp/

if [ -d $MRTKPATH/tmp ];
then
    echo "MRTK: Ordner /tmp/ vorhanden"
else
    mkdir $MRTKPATH/tmp/
    echo "MRTK: Ordner /tmp/ angelegt"
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Prüfung Internet-Verbindung

host openstreetmap.org>/dev/null
if ! [ $? -eq 0 ];
then
    echo "MRTK: KEINE Internet Verbindung! Bitte verbinden und erneut sta
    $CD bubble --debug --titles "MotoRoute-OpenStreetMap" \
        --texts "KEINE Internet Verbindung!           Bitte verbinden und
        --border-colors "a40a0F" \
        --text-colors "ffff00" \
        --background-tops "df6824" \
        --background-bottoms "fd889a" \
        --icon-files "appIcon.icns" \
        --no-timeout
    exit
fi

# Linux
if [ "$(uname)" == "Linux" ];
then
# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
    # Symbolische Verknüpfung zu WGET anlegen, um mit Mac identisch aufzu
    # WGET wird in '/usr/bin/wget' erwartet. Ggf. prüfen mit 'WHEREIS WGE
```

```
    ln -s /usr/bin/wget ./wget
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# echo
# echo "Parameter ermitteln"

./wget -L $str_Parameter -q -O $MRTKPATH/tmp/parameter.txt

par_Version=$(sed -n '1p' $MRTKPATH/tmp/parameter.txt)
# echo $par_Version

par_MRTK_Server=$(sed -n '2p' $MRTKPATH/tmp/parameter.txt)
# echo $par_MRTK_Server

par_Seazip_Server=$(sed -n '3p' $MRTKPATH/tmp/parameter.txt)
# echo $par_Seazip_Server

par_mkgmap_Server=$(sed -n '4p' $MRTKPATH/tmp/parameter.txt)
# echo $par_mkgmap_Server

par_Preprocessor=$(sed -n '5p' $MRTKPATH/tmp/parameter.txt)
# echo $par_Preprocessor

par_Style=$(sed -n '6p' $MRTKPATH/tmp/parameter.txt)
# echo $par_Style

par_Sperrung=$(sed -n '7p' $MRTKPATH/tmp/parameter.txt)
# echo $par_Sperrung

par_convert_Server=$(sed -n '8p' $MRTKPATH/tmp/parameter.txt)
# echo $par_convert_Server

par_MR_Website=$(sed -n '9p' $MRTKPATH/tmp/parameter.txt)
# echo $par_MR_Website

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Prüfung auf neue Version und Download
if ! [ "$par_Version" == "$str_Version" ];
then
    echo "MRTK: Neue Version "$par_Version" verfügbar!"

    # Mac
    if [ "$(uname)" == "Darwin" ];
    then
        # say "Neue Version verfügbar!"
        # osascript -e 'tell app "Terminal" to display dialog "Neue Versi
```

```
        $CD bubble --debug --titles "MotoRoute-Tourer-Karte" \
            --texts "Neue Version "$par_Version" verfügbar!" \
            --border-colors "a25f0a" \
            --text-colors "000000" \
            --background-tops "dfa723" \
            --background-bottoms "fdde88" \
            --icon-files "appIcon.icns"

    # Linux
    elif [ "$(uname)" == "Linux" ];
    then
        xmessage "MRTK: Neue Version "$par_Version" verfuegbar!" -nearmou
    fi
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Java-Version

java -version 2> $MRTKPATH/tmp/java.log
echo "MRTK: "$(head -n 1 $MRTKPATH/tmp/java.log)

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Konfigurations-Datei einlesen, falls vorhanden

# Inhalt "mrtkconf.txt"
# 1 conf_nsis       : --nsis

if [ -f $MRTKPATH/mrtkconf.txt ];
then
    # Family-ID gbasemap
    conf_FIDb=$(sed -n '1p' $MRTKPATH/mrtkconf.txt)
    # echo $conf_FIDb

    # Family-ID gsperrung
    conf_FIDx=$(sed -n '2p' $MRTKPATH/mrtkconf.txt)
    # echo $conf_FIDx

    # mkgmap-Option '--nsis' aktivieren (Windows-Installations-Datei)
    conf_nsis=$(sed -n '3p' $MRTKPATH/mrtkconf.txt)
    # echo $conf_nsis

    # gmapi-Option 'gmapi-builder.py'
    conf_gmapi=$(sed -n '4p' $MRTKPATH/mrtkconf.txt)
    # echo $conf_gmapi

    echo "MRTK: 'mrtkconf.txt' wird verwendet"
else
    conf_FIDb=""
```

```
    conf_FIDx=""
    conf_nsis=""
    conf_gmapi=""
fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Variablen setzen

# Mac
if [ "$(uname)" == "Darwin" ];
then
    osmconvert="$(dirname -- "$0")/osmconvert"
    osmfilter="$(dirname -- "$0")/osmfilter"

    # "gpsbabel" laden unter http://www.gpsbabel.org/download.html
    # /Applications/GPSBabelFE.app/Contents/MacOS/gpsbabel
    gpsbabel="$(dirname -- "$0")/gpsbabel/gpsbabel"
    #echo $gpsbabel

    # gmt laden unter http://www.gmaptool.eu/en/content/os-x-version
    gmt="$(dirname -- "$0")/mgmt042a/gmt"
    #echo $gmt

    gmapibuilder="$(dirname -- "$0")/gmapi-builder/gmapi-builder.py"
    #echo $gmapibuilder

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Aufruf Karten-Erstellung (Mac)
    while [ ! $filename = '' ]
    do
        Mapbuilder
    done

# Linux
elif [ "$(uname)" == "Linux" ];
then

while read -n1 -s ; do
# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Prüfung Ordner /bin/

    if [ -d $MRTKPATH/bin ];
    then
        echo "MRKT: Ordner /bin/ vorhanden"
    else
        mkdir $MRTKPATH/bin/
        echo "MRTK: Ordner /bin/ angelegt"
    fi
```

```

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Prüfung 'gmapi-builder.py' (wenn in Konfigurations-Datei Zeile 4 einget

    if [ "$conf_gmapi" = "gmapi-builder.py" ];
    then
        if [ ! -f $MRTKPATH/bin/gmapi-builder.py ];
        then
            ./wget -c $par_MRTK_Server"gmapi-builder-index-py.txt" -q \
                -O $MRTKPATH/bin/gmapi-builder.py
            chmod u+x $MRTKPATH/bin/gmapi-builder.py
            echo "MRTK: Für Adress-Index modifizierter 'gmapi-builder' ge
        else
            echo "MRTK: Für Adress-Index modifizierter 'gmapi-builder' vo
        fi
        gmapibuilder=$MRTKPATH/bin/gmapi-builder.py
    fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# 'osmconvert' und 'osmfilter' kompilieren"

    if [ ! -f $MRTKPATH/bin/osmconvert ];
    then
        ./wget -c $par_convert_Server"osmconvert.c" -q \
            -O $MRTKPATH/bin/osmconvert.c
        gcc -w -O $MRTKPATH/bin/osmconvert.c -lz -O3 \
            -o $MRTKPATH/bin/osmconvert \
            1> $MRTKPATH/tmp/osmconvert.log
        chmod u+x $MRTKPATH/bin/osmconvert
        echo "MRTK: 'osmconvert' geladen und kompiliert"
    else
        echo "MRKT: 'osmconvert' vorhanden"
    fi

    if [ ! -f $MRTKPATH/bin/osmfilter ];
    then
        ./wget -c $par_convert_Server"osmfilter.c" -q \
            -O $MRTKPATH/bin/osmfilter.c
        gcc -w -O $MRTKPATH/bin/osmfilter.c -lz -O3 \
            -o $MRTKPATH/bin/osmfilter \
            1> $MRTKPATH/tmp/osmfilter.log
        chmod u+x $MRTKPATH/bin/osmfilter
        echo "MRTK: 'osmfilter' geladen und kompiliert"
    else
        echo "MRTK: 'osmfilter' vorhanden"
    fi

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
```

```
# Variablen setzen (Linux)

    osmconvert=$MRTKPATH/bin/osmconvert
    osmfilter=$MRTKPATH/bin/osmfilter

    # "gpsbabel" muss als Paket installiert sein
    # sudo apt-get install gpsbabel
    gpsbabel=gpsbabel

    # gmt laden unter http://www.gmaptool.eu/en/content/linux-version
    # und in den Ordner dieses Scripts entpacken
    # dann ausfuehrbar machen: chmod u+x ./lgmt08186/gmt
    chmod u+x "$pathname/lgmt08186/gmt"
    gmt="$pathname/lgmt08186/gmt"

# ----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
# Aufruf Karten-Erstellung (Linux)
    Mapbuilder
done

fi

exit
```