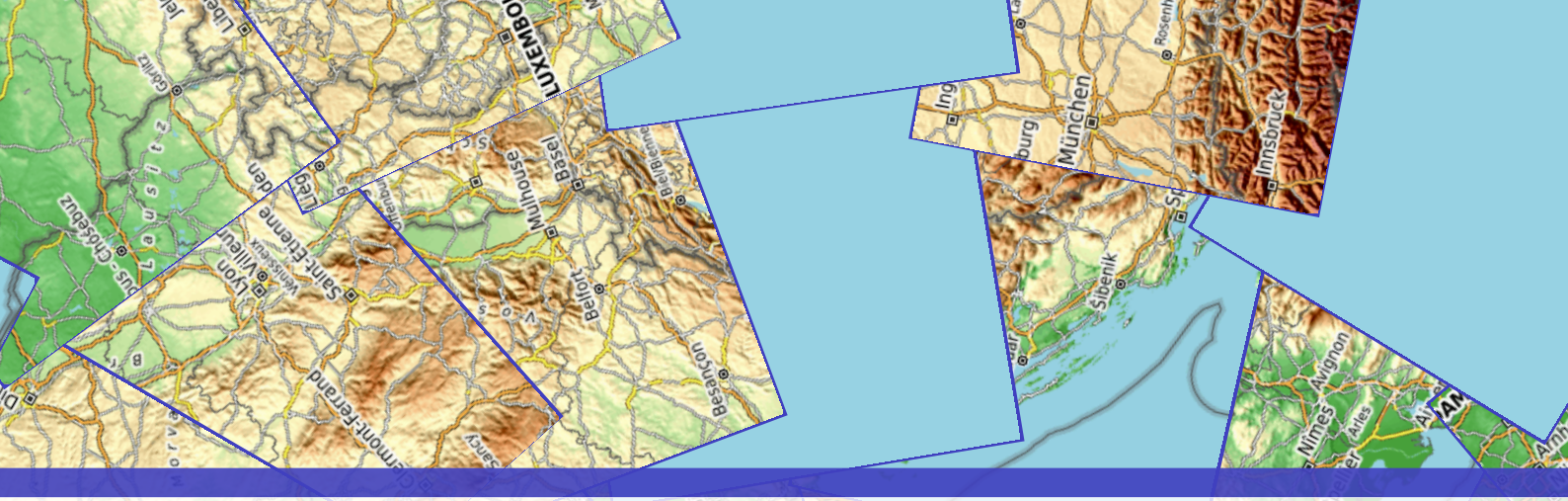
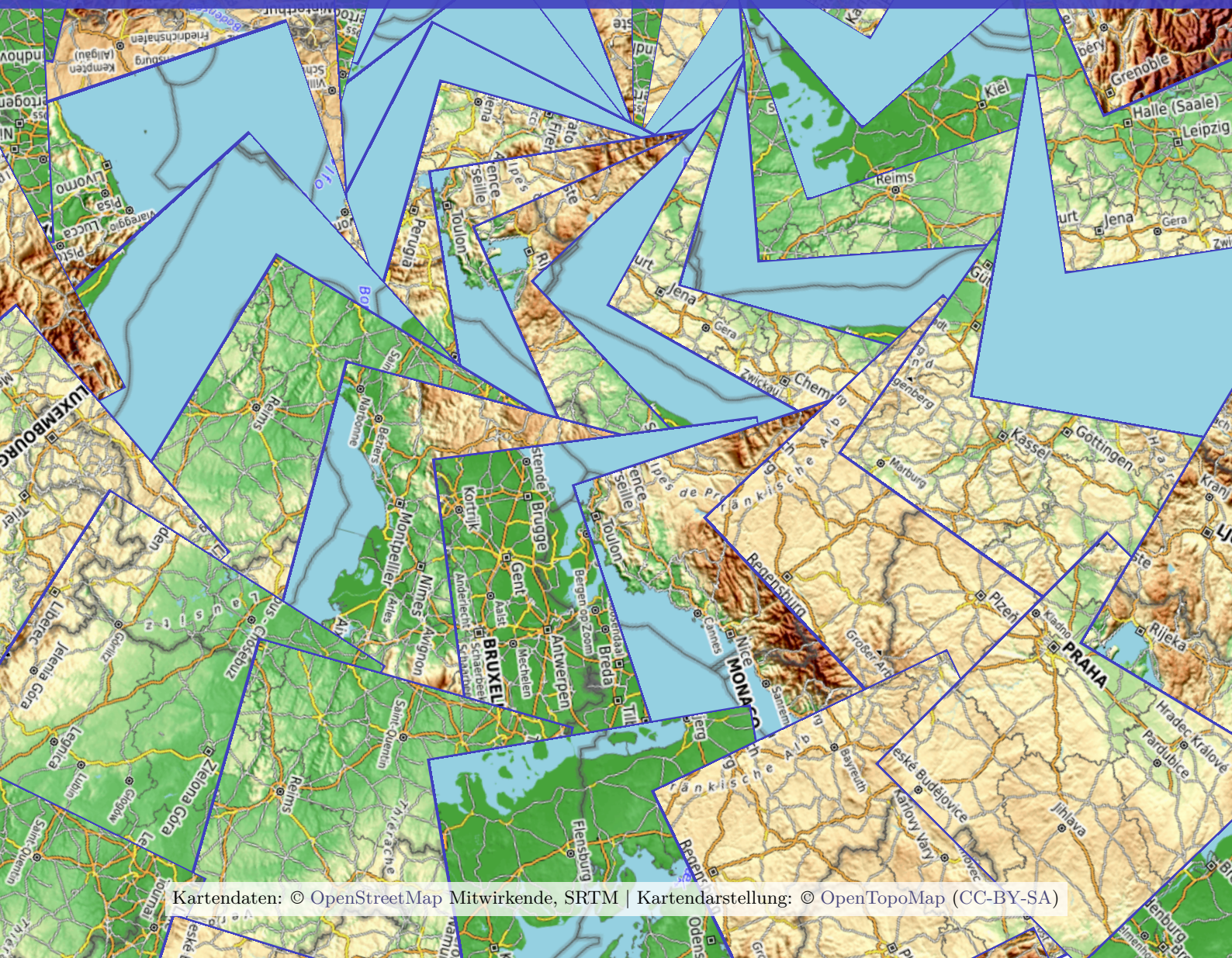


The mercatormap package

Manual for version 1.1.0 (2024/08/01)

Thomas F. Sturm



Cover code

```
\begin{tcolorbox}[spread,blankest]
\mermapset{flex tile size=40mm}
\begin{tikzpicture}
\mrcmap[type=areafit,west=5,east=15,south=46,north=54,
  source = opentopomap,
  flex area scale=4 000 000,
  tex width=\tcbtextwidth,
  tex height=\tcbtextheight,
  ]{title}
\mrcclipmap
\ExplSyntaxOn
\sys_gset_rand_seed:n {15}
\int_set:Nn \l_tmpa_int {1}
\int_until_do:nNnn \l_tmpa_int > {77}
  {
    \fp_set:Nn \l_tmpa_fp {\tcbtextwidth/2
      + \l_tmpa_int*2.5mm*cosd(18*\l_tmpa_int)}
    \fp_set:Nn \l_tmpb_fp {\tcbtextheight/2
      + \l_tmpa_int*2.5mm*sind(18*\l_tmpa_int)}
    \node[inner~sep=0pt,draw=blue!50!gray,line~width=1mm,
      at={(\fp_to_dim:N\l_tmpa_fp,\fp_to_dim:N\l_tmpb_fp)},
      rotate=\fp_eval:n{18*\l_tmpa_int+10*rand()} ]
        {\includegraphics[width=55mm,height=55mm]
          {tiles/opentopomap_
            \int_use:N \l__mermap_tile_zoom_int _
            \fp_eval:n {
            ↪  randint(\l__mermap_tile_xmin_int,\l__mermap_tile_xmax_int) }_
            \fp_eval:n {
            ↪  randint(\l__mermap_tile_ymin_int,\l__mermap_tile_ymax_int) }.png
          }
        };
    \int_incr:N\l_tmpa_int
  }
\ExplSyntaxOff
\node[font=\small\footnotesize,fill=white,opacity=0.75,text opacity=1]
  at (\tcbtextwidth/2,1cm) {\mrcmapattribution};
\node at (\tcbtextwidth/2,\tcbtextheight*0.667)
  {\begin{tcolorbox}[
    center upper, fontupper=\bfseries,boxsep=15mm, boxrule=4mm,
    sharp corners, oversize=5mm,
    colback=white, colframe=blue!50!gray,
    enhanced jigsaw, opacityback=0.8, opacityframe=0.9 ]
      {\Huge The mercatormap package\par}
      \vspace{1cm}
      Manual for version \version\ (\datum)\par
      \vspace{5mm}
      Thomas F.~Sturm
  \end{tcolorbox}};
\end{tikzpicture}
\end{tcolorbox}
```

The mercatormap package

Manual for version 1.1.0 (2024/08/01)

Thomas F. Sturm[1]

https://www.ctan.org/pkg/mercatormap

https://github.com/T-F-S/mercatormap

Abstract

The `mercatormap` package extends Ti*k*Z with tools to create map graphics. The provided coordinate system relies on the Web Mercator projection used on the Web by OpenStreetMap and others. The package supports the seamless integration of graphics from public map tile servers by a Python script. Also, common map elements like markers, geodetic networks, bar scales, routes, orthodrome pieces, and more are part of the package.

Contents

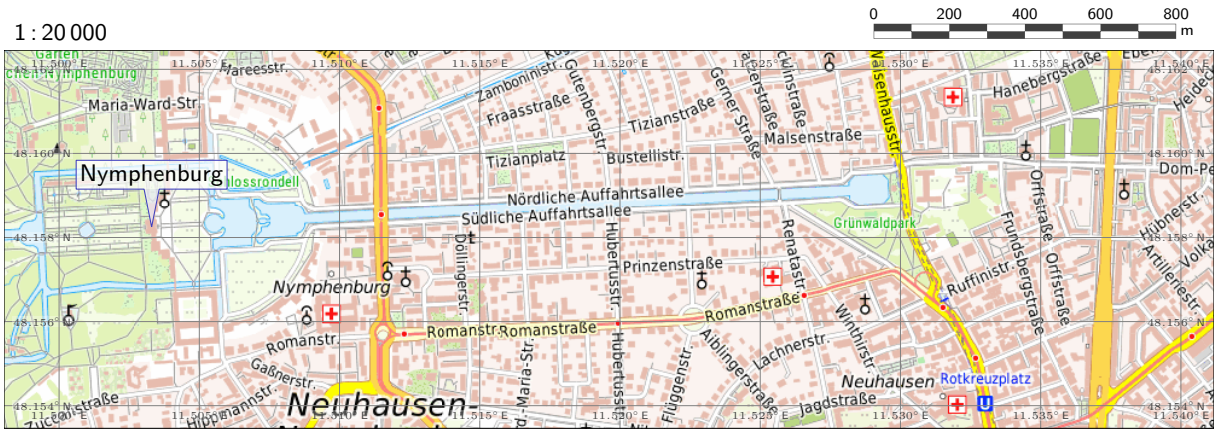
[1]Prof. Dr. Dr. Thomas F. Sturm, Institut für Mathematik und Informatik, Universität der Bundeswehr München, D-85577 Neubiberg, Germany; email: thomas.sturm@unibw.de

1 Introduction

The `mercatormap` package enables map drawing with the Web Mercator projection. This is done as an extension to Ti*k*Z [2] with is complemented by a map coordinate system and many additional commands and options to add elements like markers, geodetic networks, bar scales, routes, orthodrome pieces, distance calculations, etc. Also, the seamless integration of graphics from public map tile servers is provided through a Python script.

If you are interested in the mathematical background of the Web Mercator projection and the algorithms of this packages, you are invited to read "Thomas F. Sturm. *Some Principles of Web Mercator Maps and their Computation*. Monograph. Neubiberg: Universität der Bundeswehr München, Mar. 5, 2020. 44 pages. DOI: 10.18726/2020_3".

With very few exceptions, the package is programmed with the `expl3` programming interface for LaTeX3 [3].



Kartendarstellung: © BKG (2024), Datenquellen

1.1 Quick Start

The package is accompanied with a Python script. You should read Section 1.2 on page 6 for the Python preparations. The package can be used in three ways:

- Completely without the Python script. This is not recommended, because the usage will be quite restricted.
- With Python script, but without map tile download. There is no usage restriction, but you have to create all content yourself. To prevent map tile download, set

```
\mermapset{supply/target=none}
```

- With Python script and map tile download. You need permission and access to a map tile server. Section 4.5 on page 36 lists a selection of servers with free access (some require registration of an API key).

After Python is prepared, you may try to compile `mercatormap-example.tex` (found in the documentation directory) which contains a map of Bavaria with map tile download. Section 2 on page 7 exhibits further examples which may serve as tutorials what can be done. After the examples you find the reference manual for the package.

1.2 Installation of Python and Packages

A Python 3 script is part of the `mercatormap` package. The main purpose of this script is to download selected map tiles for the maps of the document. Also, some coordinate system computation is done by this script.

1.2.1 Python 3

Python 3 is a required prerequisite and can be downloaded from
https://www.python.org/downloads/
On systems like Linux Python is typically already installed.

To test your installation, type into a command or terminal window:

```
python --version
```

This should give a version number starting with 3. Otherwise, try

```
python3 --version
```

If this is successful, `mermap/python` → P. 27 has to be adapted to `python3`

1.2.2 Python Packages

The Python packages `Pillow` (https://pypi.org/project/Pillow/) and `requests` (https://pypi.org/project/requests/) have to be present. With some luck, they are already installed. With

```
pip3 list
```

and/or

```
pip3 list --user
```

the installed packages are listed. If `Pillow` and `requests` are not among these package, they have to be added by

```
pip3 install --user Pillow
pip3 install --user requests
```

or

```
pip3 install Pillow
pip3 install requests
```

The second choice needs administrative rights and may give conflicts with package managers. Pythonians know furthers installation methods.

1.2.3 Document Setup

For your map document you need the following:

- Add `\mrcactivatescript` → P. 27 to the document preamble. Without this command, the script is not active.
- Compile the map document with the `--shell-escape` compiler option. This allows to execute external programs like the Python script.
  Be aware that `--shell-escape` should only be used with trusted documents. Note that external programs can do anything!

2 Examples

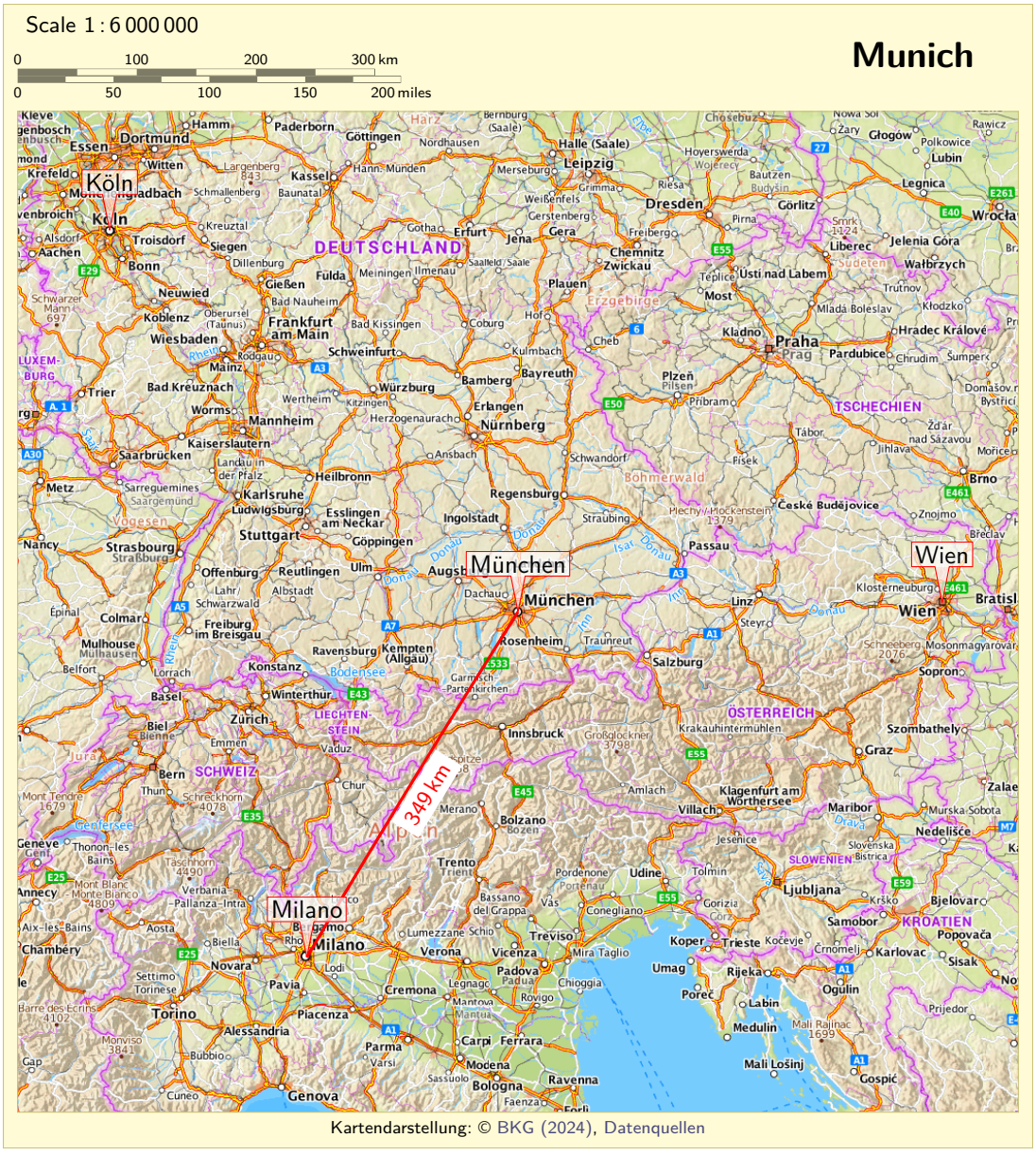
The following map examples may be used as tutorials and starting point for own applications. Also see `mercatormap-example.tex` for a compilable full example. Note to do all preparations documented in Section 1.2 on page 6.

2.1 Reference Position

With `mermap/supply/type`→P. 29`=reference` a map with a *reference position* is constructed. Here, Munich is taken as reference position and center of the map. Since the position is used more than once, it is stored with `\mrcNPdef`→P. 22 for further reference. With `mermap/supply/flex reference scale`→P. 33 the scale is set to 1:6 000 000. For the background map tiles, a `mermap/supply/source`→P. 36 is selected for download. This setup is done by `\mrcmap`→P. 35 while `\mrcdrawmap`→P. 47 draws the downloaded map tiles.

```
\begin{tikzpicture}
  \sffamily
  \mrcNPdef{munich}{48.137222}{11.575556}
  \mrcNPdef{vienna}{48.208333}{16.373056}
  \mrcNPdef{cologne}{50.938056}{6.956944}
  \mrcNPdef{milano}{45.4625}{9.186389}
  \mrcmap[type=reference,
    named position=munich,
    flex reference scale=6 000 000,
    source=topplusopen web,
    tex width=14cm,
    tex height=14cm]{examples_reference}
  \path[draw=yellow!50!gray,fill=yellow!20]
    ([xshift=-2mm,yshift=-5mm]mrcmap.south west) rectangle
    ([xshift=2mm,yshift=15mm]mrcmap.north east);
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
    {\mrcmapattribution};
  \path[draw=yellow!50!gray] (mrcmap.south west) rectangle (mrcmap.north east);

  \mrcNPdraworthodrome[red,very thick]{munich}{milano}
  \path[blue,very thick] (\mrcNPcs{munich}) --
    node[red,fill=white,sloped,below] {\mrcNPprettyloxodistance{munich}{milano}}
    (\mrcNPcs{milano});

  \mermapsetmarker{type=pin, draw=red, fill=red!10, font=\sffamily\small}
  \mrcmarker{named position=munich,  contents={M\"unchen}}
  \mrcmarker{named position=vienna,  contents={Wien}}
  \mrcmarker{named position=cologne, contents={K\"oln}}
  \mrcmarker{named position=milano,  contents={Milano}}

  \node[above left=5mm,font=\Large\bfseries] at (mrcmap.north east) {Munich};

  \node[above right] at ([yshift=10mm]mrcmap.north west)
    {Scale \mrcprettymapscale};

  \mrcdrawscalebar[width-in-km=300,partitions=6,north-west-outside=0mm;5mm,
    single, height=1mm, major style={yellow!50!gray!50!black}]
  \path[every node/.style={above,inner sep=0.5mm,font=\sffamily\tiny}]
    (mrcscalebar.north west) -- (mrcscalebar.north east)
    node[pos=0]{0} node[pos=0.3333]{100} node[pos=0.6667]{200}
    node[pos=1]{300\,km};
```

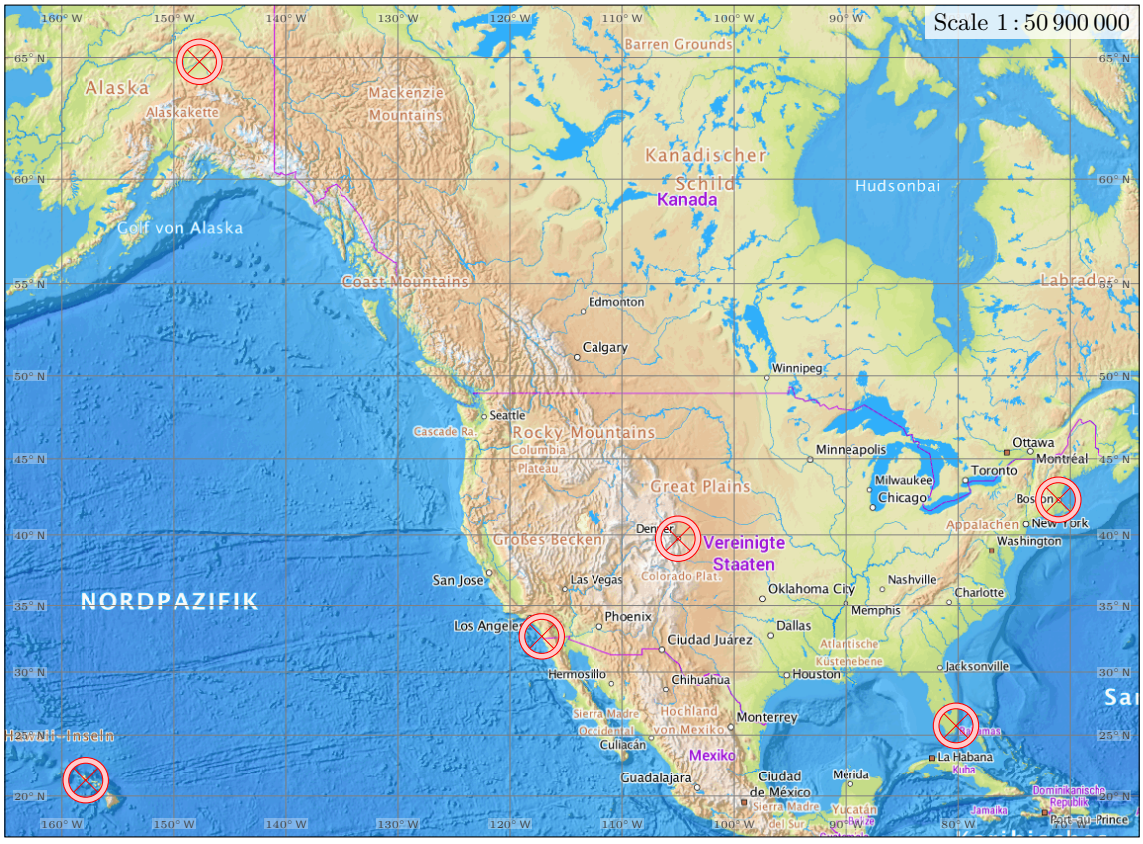
```
  \mrcdrawscalebar[width-in-mile=200,partitions=8,
    at={(mrcscalebar.south west)},placement=below right,
    single, height=1mm, major style={yellow!50!gray!50!black}]
  \path[every node/.style={below,inner sep=0.5mm,font=\sffamily\tiny}]
    (mrcscalebar.south west) -- (mrcscalebar.south east)
    node[pos=0]{0} node[pos=0.25]{50} node[pos=0.5]{100} node[pos=0.75]{150}
    node[pos=1]{200\,miles};
\end{tikzpicture}
```



2.2 Fitting Area

With mermap/supply/type →P. 29=areafit a map is constructed where a given area is fitted in. The following example lists some US-American cities and constructs an mermap/supply/area →P. 30 which contains all of them. With mermap/supply/flex area fit →P. 34=15mm a border region is added.

```
\begin{tikzpicture}
  \mrcNPdef{honolulu}{21.305225}{-157.867}
  \mrcNPdef{fairbanks}{64.8379435}{-147.7192214}
  \mrcNPdef{sandiego}{32.7146781}{-117.1640995}
  \mrcNPdef{miami}{25.7599333}{-80.1951257}
  \mrcNPdef{boston}{42.359744}{-71.061322}
  \mrcNPdef{denver}{39.7372435}{-104.997378}
  \mrcmap[type=areafit,
    area={honolulu,fairbanks,sandiego,miami,boston,denver},
    source=topplusopen web,
    tex width=15cm, tex height=11cm,
    flex area fit=15mm,
    ]{examples_fitting_area}
  \mrcdrawmap \mrcdrawnetwork
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
    {\mrcmapattribution};
  \node[below left,fill=white,opacity=0.8,text opacity=1] at (mrcmap.north east)
    {Scale \mrcprettymapscale};
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
  \foreach \city in {honolulu,fairbanks,sandiego,miami,boston,denver}
    {\mrcmarker{type=ringx, draw=red, fill=red!20, named position=\city}}
\end{tikzpicture}
```

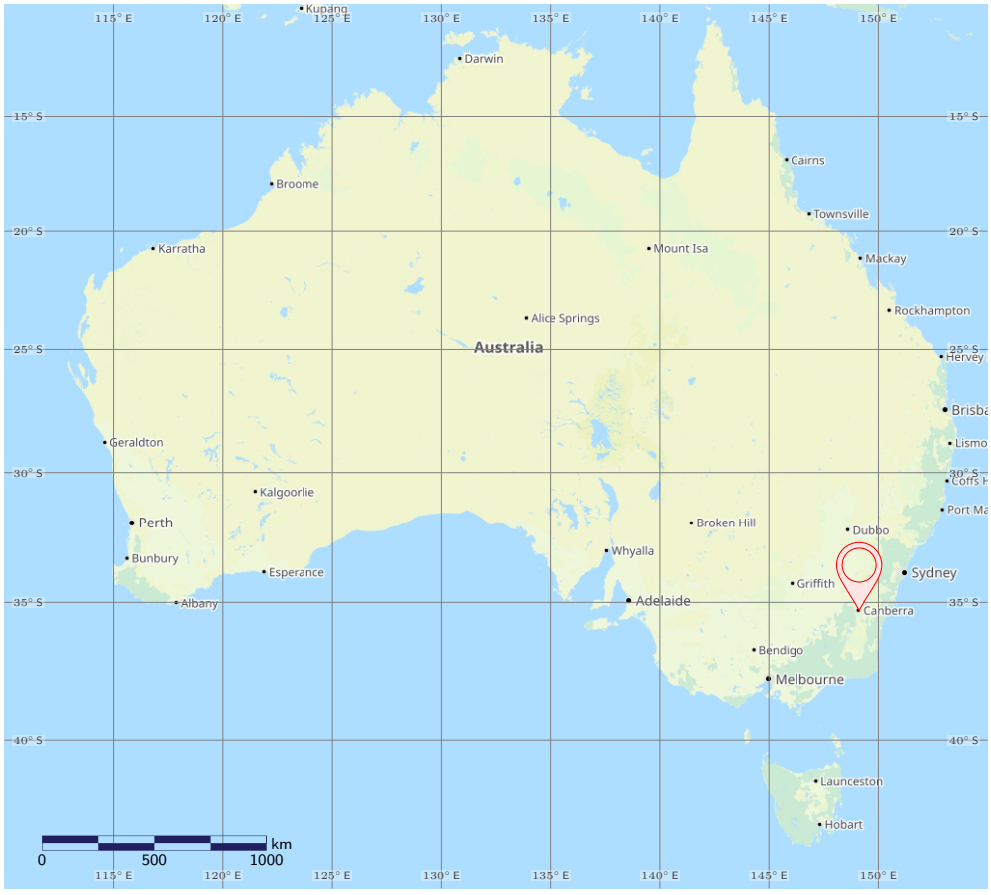


Kartendarstellung: © BKG (2024), Datenquellen

2.3 Fixed Boundaries

With mermap/supply/type →P. 29=**boundaries** a map is constructed with fixed boundaries. In contrast to the other map types, the document map size cannot be given directly but derives from the map setup. This bears the risk of too large maps. The following example is a map with exact boundaries 45.0000° S to 10.0000° S and 110.0000° E to 155.0000° E. A decent mermap/supply/zoom →P. 30 is 5 (every zoom step doubles the map size in each direction).

```
% \mrcsetapikey{thunderforest}{YOUR-API-KEY}  % registered key
\begin{tikzpicture}
  \mrcmap[type=boundaries,
    west=110,east=155,south=-45,north=-10,
    zoom=5,
    source=thunderforest outdoors,
    ]{examples_boundaries}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
    {\mrcmapattribution};
  \mrcdrawnetwork
  \mrcdrawscalebar[width-in-km=1000,partitions=4,south-west-inside=5mm,
    major style={blue!50!gray!50!black}]
  \path[every node/.style={below,inner sep=0.5mm,font=\sffamily\tiny}]
    (mrcscalebar.south west) -- (mrcscalebar.south east)
    node[pos=0]{0} node[pos=0.5]{500}
    node[pos=1]{1000} node[pos=1,right,yshift=1mm]{km};
  \mrcmarker{lat=-35.3,lon=149.116667,type=pictodropring,
    draw=red,fill=red!10}
\end{tikzpicture}
```



Maps © Thunderforest, Data © OpenStreetMap contributors

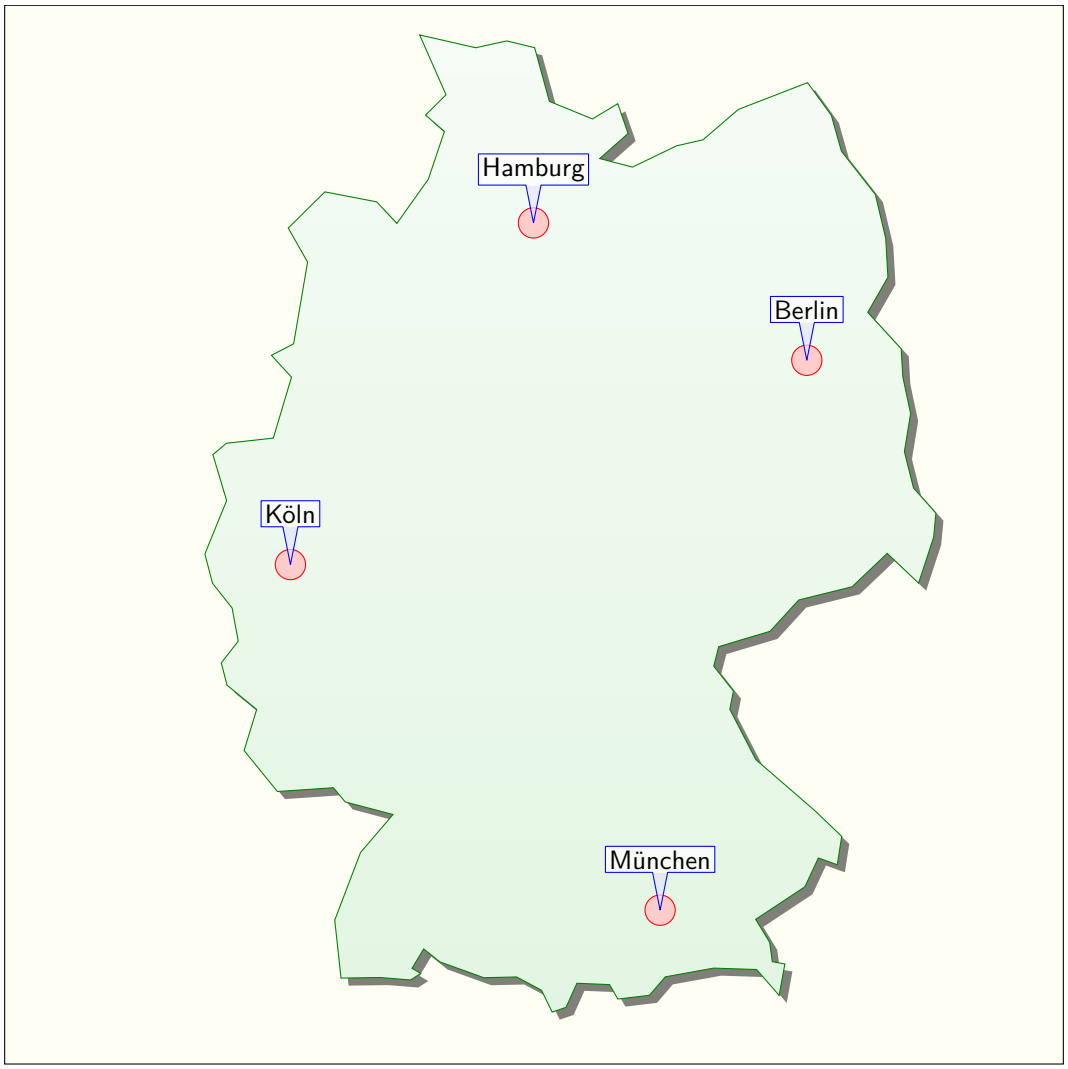
2.4 Map Without Map Tiles

There is no coercion to use downloaded map tiles, if they are not needed or wanted. With mermap/supply/target→ P. 31=none no map tiles are downloaded. The following example draws a rough polygon shape of Germany using mrcroute*→ P. 71.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.137222}{11.575556}
  \mrcNPdef{cologne}{50.938056}{6.956944}
  \mrcNPdef{hamburg}{53.550556}{9.993333}
  \mrcNPdef{berlin}{52.518611}{13.408333}
  \mrcmap[type=areafit,
    west=5,east=15,south=47,north=55,
    target=none,
    tex width=14cm, tex height=14cm,
    flex area fit=5mm
    ]{examples_routemap}
  \mrcclipmap
  \path[draw,fill=yellow!5] (mrcmap.south west) rectangle (mrcmap.north east);
  \begin{mrcroute*}[
      preaction={fill=black,opacity=.5,
          transform canvas={xshift=1mm,yshift=-1mm}},
      draw         = green!50!black,
      top color    = green!50!gray!5,
      bottom color = green!50!gray!15]
    \mrcpoint{47.57268069220318}{8.07968771809688}
    \mrcpoint{47.55206513030159}{8.458302263852103}
    \mrcpoint{47.60652271644701}{8.58564576447632}
    \mrcpoint{47.65002478441761}{8.475977737743394}
    \mrcpoint{47.8129149100372}{8.621611270810012}
    \mrcpoint{47.70405949548734}{8.824753417980977}
    \mrcpoint{47.57118357939037}{9.373387060586875}
    \mrcpoint{47.57763968687206}{9.783024034250792}
    \mrcpoint{47.46598581489455}{10.09335967295321}
    \mrcpoint{47.28014625489598}{10.22678291669482}
    \mrcpoint{47.32098723021947}{10.40092026792467}
    \mrcpoint{47.52413897927237}{10.53589647271817}
    \mrcpoint{47.5130331145614}{10.94403341747166}
    \mrcpoint{47.39097338968944}{11.04734763681045}
    \mrcpoint{47.42252867275465}{11.44170163808758}
    \mrcpoint{47.57832219946236}{11.64025530020549}
    \mrcpoint{47.65203740659043}{12.2451260343467}
    \mrcpoint{47.64005851006706}{12.78135708694006}
    \mrcpoint{47.42189293170733}{13.06242166895717}
    \mrcpoint{47.68727761120528}{13.13421507093485}
    \mrcpoint{47.70570852317097}{12.97737528172392}
    \mrcpoint{47.86836520373472}{12.94442906710615}
    \mrcpoint{48.06063312315187}{12.76937524145701}
    \mrcpoint{48.33220372303314}{13.38497370385176}
    \mrcpoint{48.57060635511297}{13.55281830406328}
    \mrcpoint{48.51465473490364}{13.78916086481733}
    \mrcpoint{48.74889509826911}{13.84448153318651}
    \mrcpoint{48.95974762545411}{13.50952525317658}
    \mrcpoint{49.37838143356481}{12.77000417129028}
    \mrcpoint{49.78495654813383}{12.44617209199484}
    \mrcpoint{49.93155217044134}{12.49159565814237}
    \mrcpoint{50.13144236663469}{12.24546848540114}
    \mrcpoint{50.28771270517358}{12.30810150664619}
```

```
\mrcpoint{50.40989390494432}{12.94903619964524}
\mrcpoint{50.65841629251407}{13.3075062965852}
\mrcpoint{50.76329878585482}{13.97452403458363}
\mrcpoint{51.02604511746936}{14.41278607347103}
\mrcpoint{50.79185062876172}{14.80477069337384}
\mrcpoint{51.15202957355457}{14.99270251686064}
\mrcpoint{51.34128169635556}{15.02098384386006}
\mrcpoint{51.53750266315752}{14.74120138252564}
\mrcpoint{51.81924062442433}{14.62724686001225}
\mrcpoint{52.11180711219343}{14.70514397703483}
\mrcpoint{52.39266284958671}{14.60756306378661}
\mrcpoint{52.60597277515545}{14.58725108586684}
\mrcpoint{52.88134471470577}{14.16977248675916}
\mrcpoint{53.14560306739344}{14.4215985570986}
\mrcpoint{53.43447627332744}{14.39440456086107}
\mrcpoint{53.76080228279986}{14.26477636730856}
\mrcpoint{54.07894830697219}{13.84113247149844}
\mrcpoint{54.34144910466647}{13.71477685386829}
\mrcpoint{54.57879841452108}{13.41784866398943}
\mrcpoint{54.38430816973541}{12.55218424463898}
\mrcpoint{54.16285895583508}{12.11170524536321}
\mrcpoint{54.11909611435144}{11.78253136604545}
\mrcpoint{53.9629318241985}{11.23001765303107}
\mrcpoint{54.02548042559235}{10.82100535035693}
\mrcpoint{54.20942342601608}{11.17572026938692}
\mrcpoint{54.42622022041155}{11.04601054827628}
\mrcpoint{54.31534430886417}{10.73027571352005}
\mrcpoint{54.4430955768925}{10.19233706975044}
\mrcpoint{54.8317448735389}{10.00757468096804}
\mrcpoint{54.87986478845961}{9.659528688958062}
\mrcpoint{54.8308224503691}{9.271730840012292}
\mrcpoint{54.92302453288877}{8.571394535898566}
\mrcpoint{54.49119535498197}{8.900730234661252}
\mrcpoint{54.34335254823343}{8.645116397508792}
\mrcpoint{54.22420563732749}{8.880431477246834}
\mrcpoint{53.87404422621243}{8.679945700393079}
\mrcpoint{53.54658560722722}{8.286556227539295}
\mrcpoint{53.7066272365128}{8.034813408953866}
\mrcpoint{53.78031118673399}{7.387421106389351}
\mrcpoint{53.51218238635882}{6.930006752473437}
\mrcpoint{53.25894778199789}{7.171145313101468}
\mrcpoint{52.64374200661974}{6.997019504055286}
\mrcpoint{52.55662475372522}{6.719778263650637}
\mrcpoint{52.38988292317155}{6.970320972580792}
\mrcpoint{51.92299983362206}{6.742905287863996}
\mrcpoint{51.88316670940459}{6.156741607019097}
\mrcpoint{51.79567780329431}{5.98801954642294}
\mrcpoint{51.43928428452571}{6.159613225461791}
\mrcpoint{51.0215877919207}{5.887936542477828}
\mrcpoint{50.79077378475993}{5.983374968822379}
\mrcpoint{50.59470107381249}{6.228393825507839}
\mrcpoint{50.33147032961352}{6.304617231833176}
\mrcpoint{50.15747573005454}{6.092865298740861}
\mrcpoint{49.97845765993797}{6.16527138749146}
\mrcpoint{49.78369815200002}{6.535162836379246}
\mrcpoint{49.45067080157749}{6.378063336429978}
\mrcpoint{49.11701097547841}{6.794068795882435}
\mrcpoint{49.14800972775451}{7.494253217695865}
```

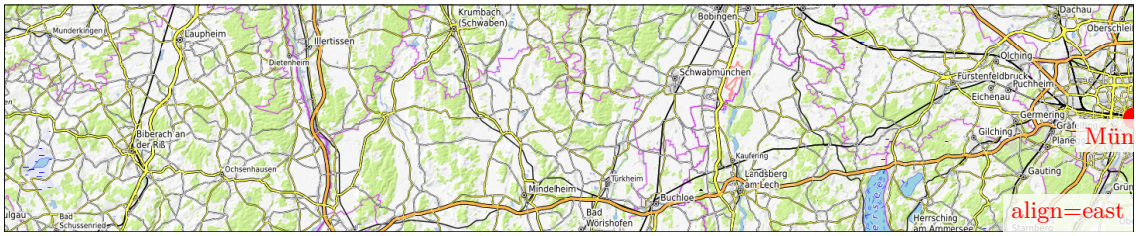
```
    \mrcpoint{49.03198382192998}{7.640486837553122}
    \mrcpoint{48.92859396692077}{8.238456398546767}
    \mrcpoint{48.61606185482966}{7.832853167339047}
    \mrcpoint{48.05389084062409}{7.510556381211842}
    \mrcpoint{47.56795313962416}{7.588782160556193}
    \mrcpoint{47.57268069220318}{8.07968771809688}
  \end{mrcroute*}
  \foreach \city / \name in {munich/M\"unchen, cologne/K\"oln,
    hamburg/Hamburg, berlin/Berlin}
    {
      \mrcmarker{named position=\city,type=knob,fill=red!20,draw=red,
        radius=2mm}
      \mrcmarker{named position=\city,type=pin,fill=blue!10,draw=blue,
        contents=\name}
    }
\end{tikzpicture}
```



2.5 Alignment of the Reference Position

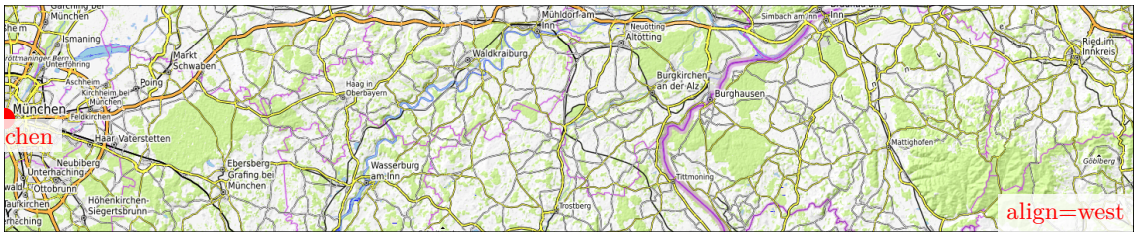
With mermap/supply/align → P. 31 the reference position can be aligned at different map positions.

```
\mrcNPdef{munich}{48.137222}{11.575556}
\foreach \a in {east,center,west,north} {
\begin{tikzpicture}
  \mrcmap[type=reference,
    named position=munich,
    flex reference scale=1 000 000,
    source=opentopomap,
    tex width=\linewidth,tex height=3cm,
    align=\a]{examples_alignment_\a}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
    {\mrcmapattribution};   \mrcclipmap
  \tikzset{every node/.style={fill=white,fill opacity=0.8,text opacity=1}}
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
  \ifmrcNPinvicinity{munich}{
    \fill[red] (mrcpos) circle (4pt) node[below] {M\"unchen};}{}
  \node[text=red,above left] at (mrcmap.south east) {align=\a};
\end{tikzpicture}
\par}
```



Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)

Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)



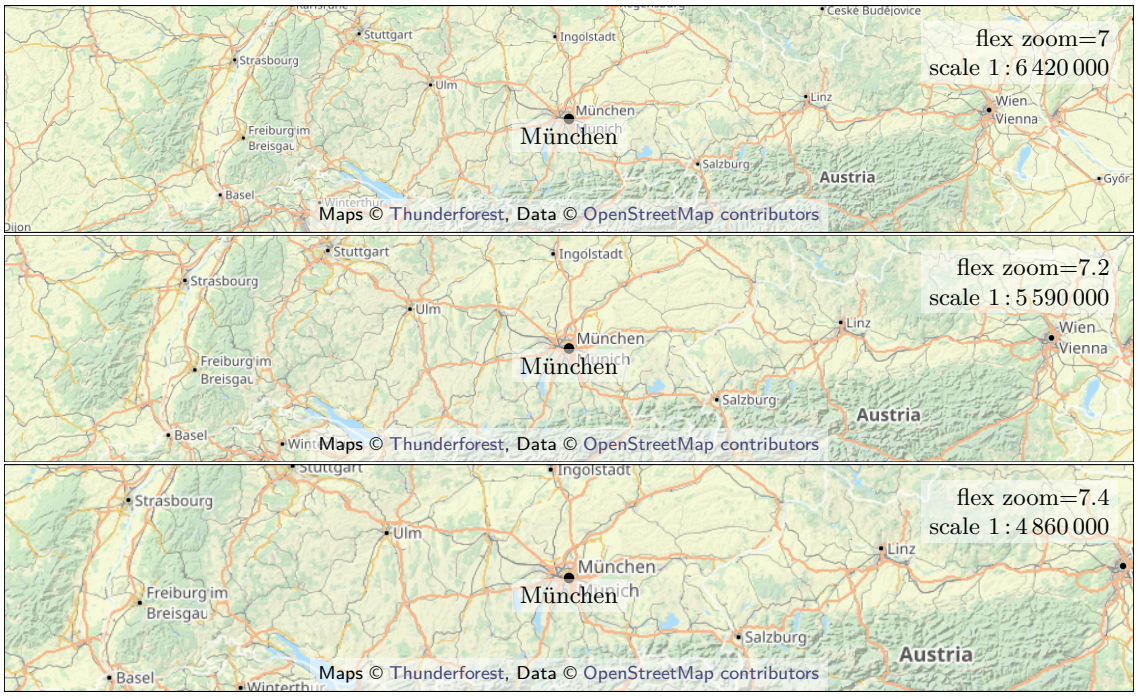
Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)

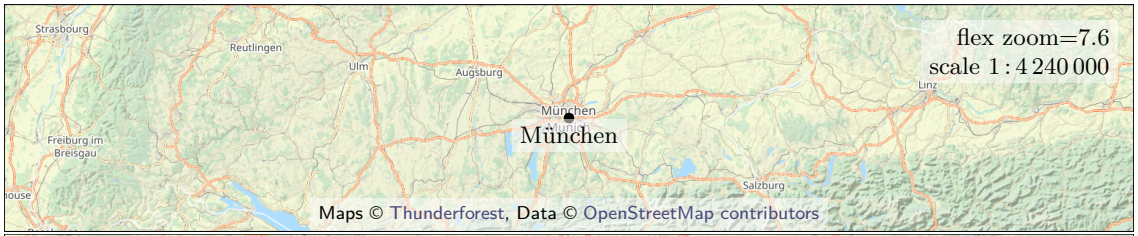
Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)

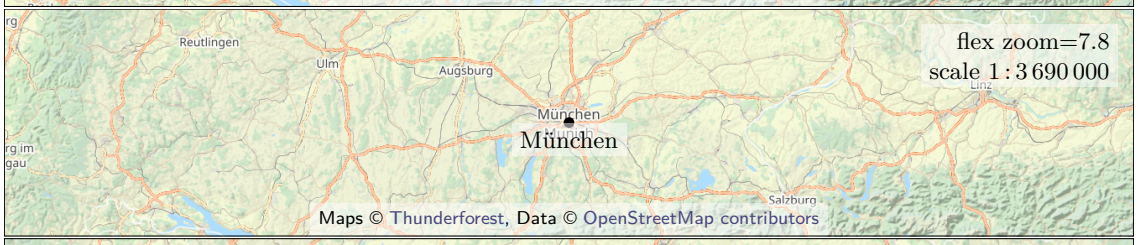
2.6 Flexible Zoom

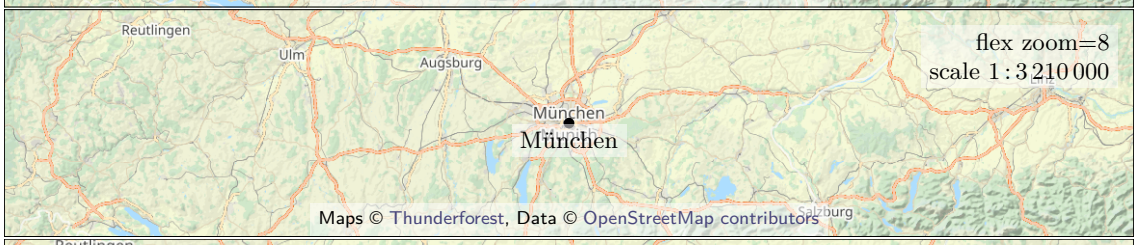
Map tiles are only provided at fixed zoom levels with natural numbers, but the package allows a `mermap/flex zoom` → P. 48 with rational numbers. The flexible zoom is realized by combining a suitable fixed zoom with an adapted document tile scaling, see Section 5.2 on page 48. The following example shows a more or less smooth zoom increase. The same technique is used by all options starting with `flex`, e.g. `mermap/supply/flex reference scale` → P. 33 or `mermap/supply/flex area fit` → P. 34 as seen in the examples before.

```
% \mrcsetapikey{thunderforest}{YOUR-API-KEY}  % registered key
\mrcNPdef{munich}{48.137222}{11.575556}
\foreach \zz in {7.0,7.2,...,9.0} {
\edef\z{\fpeval{round(\zz,1)}}
\begin{tikzpicture}
  \mermapset{flex zoom=\z}
  \mrcmap[type=reference,
      named position=munich,
      source=thunderforest outdoors,
      tex width=\linewidth,
      tex height=3cm
    ]{examples_flex_zoom_\z}
  \mrcdrawmap
  \mrcclipmap
  \tikzset{every node/.style={fill=white,fill opacity=0.5,text opacity=1}}
  \node[above,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
      {\mrcmapattribution};
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
  \node[below left=2mm,align=right] at (mrcmap.north east)
    {flex zoom=\fpeval{round(\z,1)}\\ scale \mrcprettymapscale};
  \ifmrcNPinvicinity{munich}{
    \fill (mrcpos) circle (2pt) node[below] {M\"unchen};}{}
\end{tikzpicture}\par}
```




flex zoom=7.6
scale 1 : 4 240 000
München
Maps © Thunderforest, Data © OpenStreetMap contributors


flex zoom=7.8
scale 1 : 3 690 000
München
Maps © Thunderforest, Data © OpenStreetMap contributors


flex zoom=8
scale 1 : 3 210 000
München
Maps © Thunderforest, Data © OpenStreetMap contributors


flex zoom=8.2
scale 1 : 2 790 000
München
Maps © Thunderforest, Data © OpenStreetMap contributors


flex zoom=8.4
scale 1 : 2 430 000
München
Maps © Thunderforest, Data © OpenStreetMap contributors

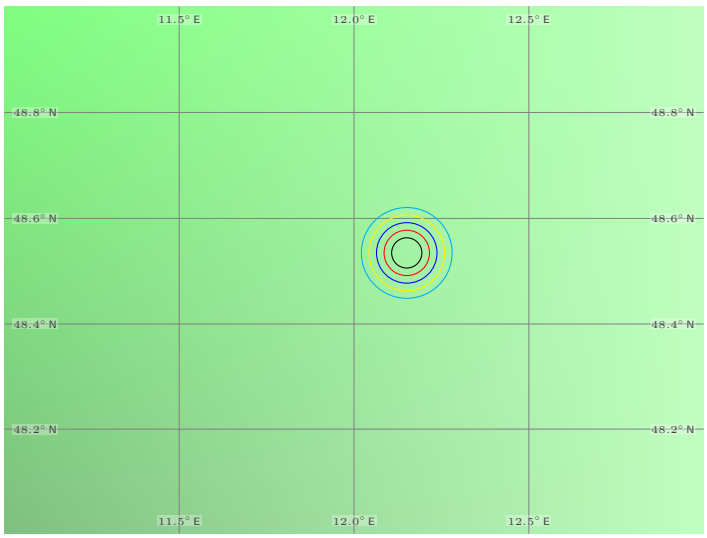

flex zoom=8.6
scale 1 : 2 120 000
München
Maps © Thunderforest, Data © OpenStreetMap contributors


flex zoom=8.8
scale 1 : 1 840 000
München
Maps © Thunderforest, Data © OpenStreetMap contributors


flex zoom=9
scale 1 : 1 600 000
München
Maps © Thunderforest, Data © OpenStreetMap contributors

3 Map Definition and Map Coordinates

```
\begin{tikzpicture}
\mrcdefinemap{west=11,east=13,north=49,south=48}
\mrcdrawmap[draw=path]
\mrcdrawnetwork
\draw       (mrc cs:latitude=48.53475,longitude=12.15087)   circle (2mm);
\draw[red]  (mrc cs:lat=48.53475,lon=12.15087)              circle (3mm);
\draw[blue] (mrcq cs:48.53475:12.15087)                     circle (4mm);
\ifmrcinmap{48.53475}{12.15087}{     \draw[yellow] (mrcpos) circle (5mm);}{}
\ifmrcinvicinity{48.53475}{12.15087}{\draw[cyan]   (mrcpos) circle (6mm);}{}
\end{tikzpicture}
```



3.1 Option Setting

`\mermapset{`⟨*options*⟩`}`

Sets ⟨*options*⟩ for all following maps inside the current TEX group. All options share the common prefix `mermap/`, e.g. for setting `mermap/vicinity` → P. 23 use

```
\mermapset{vicinity=3cm}
```

Also see `\mrcdefinemap` → P. 18, `\mermapsetsupply` → P. 29, and `\mermapsetmarker` → P. 59. Note that the options by `\mermapset` are `expl3` [3] keys while Ti*k*Z [2] uses its own key management.

3.2 Manual Map Definition

The following map definition is only relevant, if no script setup is used and maps are generated completely manually. See Section 4 on page 27 for script aided map definitions.

`\mrcdefinemap{`⟨*options*⟩`}`

Establishes a map inside a `tikzpicture` environment following and applying the given ⟨*options*⟩. All options share the common prefix `mermap/mapdef/`. After `\mrcdefinemap` is applied, map drawing and map coordinates can be used.

- `\mrcdefinemap` can be used directly, if no tile download and no script setup is intended.
- `\mrcdefinemap` is implicitly used with `\mrcapplymap` → P. 35 and `\mrcmap` → P. 35. In this case, all options are also set implicitly.

`mermap/mapdef/north=`⟨*map north latitude*⟩ (no default, initially 50)

Northern latitude degree of the visible map, possibly negative for the southern hemisphere, lower than 90 but always larger than `mermap/mapdef/south`. It is accessible as `\mrcmapnorth` (use read-only).

`mermap/mapdef/south=`⟨*map south latitude*⟩ (no default, initially 48)

Southern latitude degree of the visible map, possibly negative for the southern hemisphere, larger than −90 but always lower than `mermap/mapdef/north`. It is accessible as `\mrcmapsouth` (use read-only).

`mermap/mapdef/west=`⟨*map west longitude*⟩ (no default, initially 11)

Western longitude degree of the visible map, possibly negative for the western hemisphere, possibly shifted periodically, but always lower than `mermap/mapdef/east`. It is accessible as `\mrcmapwest` (use read-only).

`mermap/mapdef/east=`⟨*map east longitude*⟩ (no default, initially 13)

Eastern longitude degree of the visible map, possibly negative for the western hemisphere, possibly shifted periodically, but always larger than `mermap/mapdef/west`. It is accessible as `\mrcmapeast` (use read-only).

3.3 Further Map Definition Options

The following options are typically implicitly set by `\mrcapplymap` → P. 35 and not manually by `\mrcdefinemap`. However, some values are computationally used in all cases. They can be ignored as pure technical information.

`mermap/mapdef/xmin=`⟨*map tile x minimum*⟩ (no default, initially 271)

Minimal x coordinate of the map tiles.

`mermap/mapdef/xmax=`⟨*map tile x maximum*⟩ (no default, initially 275)

Maximal x coordinate of the map tiles.

`mermap/mapdef/ymin=`⟨*map tile y minimum*⟩ (no default, initially 173)

Minimal y coordinate of the map tiles.

`mermap/mapdef/ymax=`⟨*map tile y maximum*⟩ (no default, initially 177)

Maximal y coordinate of the map tiles.

`mermap/mapdef/zoom=`⟨*map zoom*⟩ (no default, initially 9)

Map tile zoom factor alias z coordinate of the map tiles.

`mermap/mapdef/pixelwidth=`⟨*map width in pixels*⟩ (no default, initially 100)

Width of the visible map expressed in pixels of the source file(s). It is accessible as `\mrcpixelwidth` (use read-only).

mermap/mapdef/pixelheight=⟨*map height in tiles*⟩ (no default, initially 100)

Height of the visible map expressed in pixels of the source file(s). It is accessible as **\mrcpixelheight** (use read-only).

mermap/mapdef/westoffset=⟨*map tile offset (west)*⟩ (no default, initially 0)

Distance of the visible map from the western edge of the most western tile expressed in tiles (range from 0 to 1).

mermap/mapdef/northoffset=⟨*map tile offset (north)*⟩ (no default, initially 0)

Distance of the visible map from the northern edge of the most northern tile expressed in tiles (range from 0 to 1).

mermap/mapdef/southoffset=⟨*map tile offset (south)*⟩ (no default, initially 0)

Distance of the visible map from the southern edge of the most southern tile expressed in tiles (range from 0 to 1).

mermap/mapdef/basename=⟨*map tile base name*⟩ (no default, initially `tiles/tile`)

File base name for the tiles.

mermap/mapdef/attribution=⟨*attribution text*⟩ (no default, initially empty)

Attribution text for the map source. Typically, it acknowledges the copyright of the map data provider. It may contain hyperlinks. It is accessible as **\mrcmapattribution** (use read-only).

mermap/mapdef/attribution print=⟨*attribution text*⟩ (no default, initially empty)

Attribution text for the map source. In contrast to `mermap/mapdef/attribution` it is intended for media that does not support hyperlinks like printed posters, books, etc. It is accessible as **\mrcmapattributionprint** (use read-only).

mermap/mapdef/resource=⟨*map resource*⟩ (no default, initially `none`)

Available map resource with following feasible values:

- `none`: No tiles and no merged map.
- `tiles`: Map tiles locally available.
- `mergedmap`: Single map picture file merged from tiles locally available.
- `wmsmap`: Single map picture file locally available.

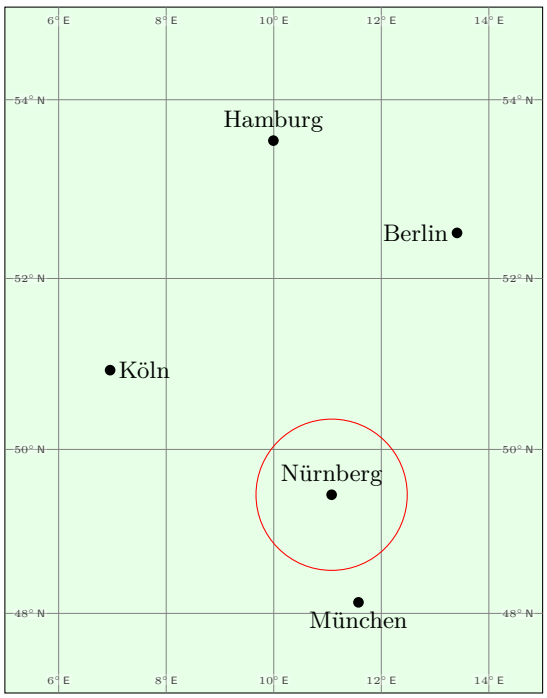
mermap/mapdef/tile size=⟨*length*⟩ (no default, initially `32.512mm`)

Typically set computationally. It is identical to `mermap/tile size` → P. 48 which is the recommended user option for manual setup.

3.4 Map Coordinate System

After a map is defined inside a `tikzpicture` environment by `\mrcdefinemap` → P. 18, `\mrcapplymap` → P. 35, or `\mrcmap` → P. 35, a Mercator map coordinate system can be used. The border of the visible map is denoted by a Ti*k*Z node `mrcmap`.

```
\begin{tikzpicture}
  \mrcNPdef{nuremberg}{49.45522}{11.07631}
  \mermapset{tile size=2cm}
  \mrcdefinemap{west=5,east=15,south=47,north=55,zoom=7}
  \path[draw,fill=green!10] (mrcmap.south west) rectangle (mrcmap.north east);
  \mrcdrawnetwork
  \fill (mrc cs:latitude=48.137222,longitude=11.575556) circle (2pt)
    node[below] {M\"unchen};
  \fill (mrc cs:lat=53.550556,lon=9.993333) circle (2pt)
    node[above] {Hamburg};
  \fill (mrcq cs:52.518611:13.408333) circle (2pt)
    node[left] {Berlin};
  \fill (\mrcNPcs{nuremberg}) circle (2pt) node[above] {N\"urnberg};
  \ifmrcinmap{50.938056}{6.956944}{
    \fill (mrcpos) circle (2pt) node[right] {K\"oln};}{}
  \ifmrcNPinmap{nuremberg}{\draw[red] (mrcpos) circle (1cm);}{}
\end{tikzpicture}
```



The `mrc cs` coordinate system defines a map point by `mermap/cs/latitude` and `mermap/cs/longitude`

`mermap/cs/latitude`=⟨*latitude*⟩ (no default)

Sets the ⟨*latitude*⟩ of a map point.

`mermap/cs/longitude`=⟨*longitude*⟩ (no default)

Sets the ⟨*longitude*⟩ of a map point.

```
\fill (mrc cs:latitude=48.137222,longitude=11.575556) circle (2pt);
```

A map point can also be defined by shorter variants `mermap/cs/lat` → P. 21 and `mermap/cs/lon` → P. 21

`mermap/cs/lat=`⟨*latitude*⟩ (no default)

Sets the ⟨*latitude*⟩ of a map point.

`mermap/cs/lon=`⟨*longitude*⟩ (no default)

Sets the ⟨*longitude*⟩ of a map point.

```
\fill (mrc cs:lat=48.137222,lon=11.575556) circle (2pt);
```

A map point can be defined even quicker by `(mrcq cs:`⟨*latitude*⟩`:`⟨*longitude*⟩`)`.

```
\fill (mrcq cs:48.137222:11.575556) circle (2pt);
```

`\mrcpgfpoint{`⟨*latitude*⟩`}{`⟨*longitude*⟩`}`

Yields a low level `pgf` point location given by ⟨*latitude*⟩ and ⟨*longitude*⟩. This can be used like `\pgfpoint`.

```
\pgfpathcircle{\mrcpgfpoint{49.45522}{11.07631}}{2pt}
\pgfusepath{fill}
```

3.5 Named Positions

`\mrcNPdef{`⟨*name*⟩`}{`⟨*latitude*⟩`}{`⟨*longitude*⟩`}`

A coordinate pair of ⟨*latitude*⟩ and ⟨*longitude*⟩ can be saved as *named position* (NP) to a ⟨*name*⟩ for later use. The *named position* just stores the given values as evaluated floating points but without coordinate system processing. Therefore, a named position can be used outside a map definition or `tikzpicture` environment, even as a preset for the whole document. Note that this saving is not global but only effective inside the current TEX group.

```
\mrcNPdef{nuremberg}{49.45522}{11.07631}
```

`\mrcNPfrompoint{`⟨*name*⟩`}{`⟨Ti*k*Z *point*⟩`}`

Latitude and *longitude* of a given ⟨Ti*k*Z *point*⟩ are calculated and saved as *named position* (NP) with given ⟨*name*⟩. `\mrcNPfrompoint` can only be used after a valid a map definition inside a `tikzpicture` environment.

```
\mrcNPfrompoint{mapcenter}{mrcmap.center}
\mrcNPfrompoint{mytest}{[xshift=1cm,yshift=1cm]mrcmap.south west}
```

`\mrcNPcs{`⟨*name*⟩`}`

A map point definition from the ⟨*name*⟩ of a previously saved *named position* (NP).

```
\fill (\mrcNPcs{nuremberg}) circle (2pt) node[above] {N\"urnberg};
```

`\mrcNPlat{`⟨*name*⟩`}`

Inserts the *latitude* of a *named position* with given ⟨*name*⟩. `\mrcNPlat` is expandable and may be used in floating point expressions.

```
\mrcNPdef{nuremberg}{49.45522}{11.07631}
Latitude: \mrcNPlat{nuremberg}\\
Longitude: \mrcNPlon{nuremberg}
```

Latitude: 49.45522
Longitude: 11.07631

`\mrcNPlon{`⟨*name*⟩`}`

Inserts the *longitude* of a *named position* with given ⟨*name*⟩. `\mrcNPlon` is expandable and may be used in floating point expressions.

3.6 Tests for Points to be inside or outside a Map

When a map is drawn, \mrcclipmap → P. 47 can be used to set up a Ti*k*Z clip environment which automatically removes all content which is not inside the defined map. However, the Ti*k*Z position of a geographic point has to be computed first to decide, if this point is to be drawn. Since TEX length registers do not allow large dimensions, compiler errors are possible to happen.

The following tests check given geographic coordinates before they are transformed to TEX dimensions and avoid such compiler errors.

\ifmrcinmap{⟨*latitude*⟩}{⟨*longitude*⟩}{⟨*true*⟩}{⟨*false*⟩}

If the given ⟨*latitude*⟩ and ⟨*longitude*⟩ describes a point inside the visible map, the ⟨*true*⟩ code is executed, otherwise the ⟨*false*⟩ code.
Inside the ⟨*true*⟩ code a Ti*k*Z coordinate mrcpos describes the given point. Also, mrclastpos denotes the *last* position before.

```
\ifmrcinmap{48.137222}{11.575556}{\fill (mrcpos) circle (2pt);}{}
```

\ifmrcNPinmap{⟨*name*⟩}{⟨*true*⟩}{⟨*false*⟩}

If the given *named position* (NP) ⟨*name*⟩ describes a point inside the visible map, the ⟨*true*⟩ code is executed, otherwise the ⟨*false*⟩ code.

```
\mrcNPdef{munich}{48.137222}{11.575556}
\ifmrcNPinmap{munich}{\fill (mrcpos) circle (2pt);}{}
```

Very similar to \ifmrcinmap is \ifmrcinvicinity.

\ifmrcinvicinity{⟨*latitude*⟩}{⟨*longitude*⟩}{⟨*true*⟩}{⟨*false*⟩}

If the given ⟨*latitude*⟩ and ⟨*longitude*⟩ describes a point inside a vicinity of the visible map, i.e. the map *plus* a margin of mermap/vicinity, the ⟨*true*⟩ code is executed, otherwise the ⟨*false*⟩ code.
Inside the ⟨*true*⟩ code a Ti*k*Z coordinate mrcpos describes the given point. Also, mrclastpos denotes the *last* position before.
\ifmrcinvicinity may be used for objects of a certain size like markers which could be partly visible even when their reference point is outside the visible map (but nearby).

```
\ifmrcinvicinity{48.137222}{11.575556}{\fill (mrcpos) circle (2pt);}{}
```

\ifmrcNPinvicinity{⟨*name*⟩}{⟨*true*⟩}{⟨*false*⟩}

If the given *named position* (NP) ⟨*name*⟩ describes a point inside a vicinity of the visible map, the ⟨*true*⟩ code is executed, otherwise the ⟨*false*⟩ code, see \ifmrcinvicinity.

```
\mrcNPdef{munich}{48.137222}{11.575556}
\ifmrcNPinvicinity{munich}{\fill (mrcpos) circle (2pt);}{}
```

mermap/vicinity=⟨*width*⟩ (no default, initially 2cm)

The vicinity of the map is the given map plus a border in all directions with the given ⟨*width*⟩.

3.7 Formatted Coordinate Output

`\mrcformlat[`⟨*options*⟩`]{`⟨*latitude*⟩`}`

Formatted output for a given ⟨*latitude*⟩ following given ⟨*options*⟩. Formatting ⟨*options*⟩ are described in the following.

```
Latitude from \mrcformlat{-24.29} to \mrcformlat{12.3456789}.
```

Latitude from 24.2900° S to 12.3457° N.

`\mrcformlon[`⟨*options*⟩`]{`⟨*longitude*⟩`}`

Formatted output for a given ⟨*longitude*⟩ following given ⟨*options*⟩. Formatting ⟨*options*⟩ are described in the following.

```
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24.2900° W to 12.3457° E.

`mermap/format angle=`⟨*type*⟩ (no default, initially `decimal-4`)

The ⟨*type*⟩ defines some formatting settings for `\mrcformlat` and `\mrcformlon`. Internally, the `\ang` macro from package `siunitx` [4] is used which can be controlled by further settings of `siunitx` like digit grouping or changing the decimal marker.
Feasible values for ⟨*type*⟩ are

- `decimal`: decimal output without rounding.

```
\mermapset{format angle=decimal}
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24.29° W to 12.345 678 9° E.

- `decimal-0`: decimal output with rounding to full degrees.

```
\mermapset{format angle=decimal-0}
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24° W to 12° E.

- `decimal-1`: decimal output with rounding to one place.

```
\mermapset{format angle=decimal-1}
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24.3° W to 12.3° E.

- `decimal-2`: decimal output with rounding to two places.

```
\mermapset{format angle=decimal-2}
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24.29° W to 12.35° E.

- `decimal-3`: decimal output with rounding to three places.

```
\mermapset{format angle=decimal-3}
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24.290° W to 12.346° E.

- `decimal-4`: decimal output with rounding to four places.

```
\mermapset{format angle=decimal-4}
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24.2900° W to 12.3457° E.

- `degree`: output with rounding to full degrees. This is an alias for `decimal-0`.

```
\mermapset{format angle=degree}
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24° W to 12° E.

- `minute`: output with rounding to degrees and full minutes.

```
\mermapset{format angle=minute}
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24°17′ W to 12°21′ E.

- `second`: output with rounding to degrees, minutes, and full seconds.

```
\mermapset{format angle=second}
Longitude from \mrcformlon{-24.29} to \mrcformlon{12.3456789}.
```

Longitude from 24°17′24″ W to 12°20′44″ E.

`mermap/format south=`⟨*code*⟩ (no default, initially `#1\,S`)

Defines the format ⟨*code*⟩ for a negative latitude. Use `#1` to place the number (without sign).

```
\mermapset{format south={$-#1$}}
Latitude \mrcformlat{-24.29}.
```

Latitude −24.2900°.

`mermap/format north=`⟨*code*⟩ (no default, initially `#1\,N`)

Defines the format ⟨*code*⟩ for a non-negative latitude. Use `#1` to place the number.

```
\mermapset{format north={#1 North}}
Latitude \mrcformlat{12.3456789}.
```

Latitude 12.3457° North.

mermap/format east=⟨*code*⟩ (no default, initially `#1\,E`)

Defines the format ⟨*code*⟩ for a positive longitude. Use `#1` to place the number (without sign).

```
\mermapset{format east={#1\,O}}
\sisetup{output-decimal-marker={,}}
L\"angengrad \mrcformlon{12.3456789}.
```

Längengrad 12,3457° O.

mermap/format west=⟨*code*⟩ (no default, initially `#1\,W`)

Defines the format ⟨*code*⟩ for a negative longitude. Use `#1` to place the number.

```
\mermapset{format west={West: #1}}
Longitude \mrcformlon{-24.29}.
```

Longitude West: 24.2900°.

mermap/format NEWS numeric (no value)

Defines the format for north, east, west, and south as numeric value without N, E, W, S.

```
\mermapset{format NEWS numeric}
Longitude \mrcformlon{-24.29} and \mrcformlon{12.3456789}.\\
Latitude  \mrcformlat{-24.29} and \mrcformlat{12.3456789}.
```

Longitude −24.2900° and 12.3457°.
Latitude −24.2900° and 12.3457°.

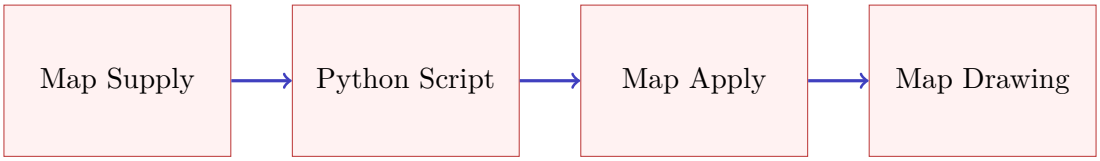
mermap/format NEWS absolute (no value)

Defines the format for north, east, west, and south as absolute value without N, E, W, S and without algebraic sign.

```
\mermapset{format NEWS absolute}
Longitude \mrcformlon{-24.29} and \mrcformlon{12.3456789}.\\
Latitude  \mrcformlat{-24.29} and \mrcformlat{12.3456789}.
```

Longitude 24.2900° and 12.3457°.
Latitude 24.2900° and 12.3457°.

4 Automated Map Definition and Map Tiles



As illustrated above, the script aided map definition is a process with several stages.

- Map Supply: `\mrcsupplymap` → P. 29 is the replacement of the manual setup by `\mrcdefinemap` → P. 18. Actually, it is quite similar to `\mrcdefinemap` → P. 18. With `\mrcsupplymap` → P. 29 directions for the following Python script are formulated.
- Python Script: The script is executed by `\mrcsupplymap` → P. 29 during compilation. It does some coordinate system computations and downloads map tiles from a Web server. Finally, it writes a map definition into a file ⟨*id*⟩`.def`.
- Map Apply: `\mrcapplymap` → P. 35 reads and applies the map definition from ⟨*id*⟩`.def`.
- Map Drawing: Afterwards, the map can be drawn by `\mrcdrawmap` → P. 47 and other commands.

A map can be applied more than once, e.g. reused later in the document. If this is not needed, map supply and map apply can be combined by `\mrcmap` → P. 35.

4.1 Script Activation

Remember to install Python beforehand, see Section 1.2 on page 6.

`\mrcactivatescript`

Use this inside the preamble of your document to activate the accompanying Python script. Without this command, the script is not executed! If the document is final (or the maps are final), this line could be removed and the document should be compilable without script.

`mermap/python`=⟨*python*⟩ (no default, initially `python`)

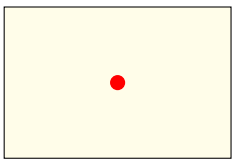
Names the Python 3 interpreter as ⟨*python*⟩. If your Python 3 interpreter is not called `python`, but e.g. `python3`, then use

```
\mermapset{python=python3}
```

4.2 Map Types

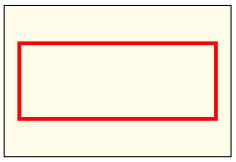
Currently, there are three methods provided how a map is computed by the accompanying Python script. The technical background is documented in [1, Section 5].

1. `mermap/supply/type`→P. 29`=reference`:



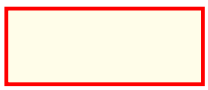
The default method determines the map dimensions from a reference position and given document map dimensions. Also, a zoom level `mermap/supply/zoom`→P. 30 is required which relates to the Web Mercator map tile covering of the Earth. A higher zoom level gives a growing number of smaller map tiles. Alternative to the zoom level, a ⟨*scale denominator*⟩ can be provided by `mermap/supply/flex area scale`→P. 33, `mermap/flex scale`→P. 49 or `mermap/supply/flex reference scale`→P. 33 which determines the zoom level implicitly. As default, the reference position is the center of the map, but can be aligned at the map borders. This method is quite safe to use and could be the preferred one for many applications like showing the neighborhood of a route target. Finding the best reference point for depicting a certain area could be more tricky.

2. `mermap/supply/type`→P. 29`=areafit`:



The map dimensions are determined by an area with latitude and longitude boundaries which is fitted into given document map dimensions. The zoom level is computed accordingly for a fixed document tile size or by `mermap/supply/flex area fit`→P. 34. In any case, the map contains the target area plus some protrusion. This method is also quite safe to use and may be the preferred one for many applications like showing a map which contains a bunch of markers (e.g. for locations of universities). If the map borders are required to exactly meet the boundaries, the third method can be regarded.

3. `mermap/supply/type`→P. 29`=boundaries`:



The most obvious method determines the map dimensions from latitude and longitude boundaries. For this, a corresponding zoom level `mermap/supply/zoom`→P. 30 is required which relates to the Web Mercator map tile covering of the Earth. Alternative to the zoom level, a ⟨*scale denominator*⟩ can be provided by `mermap/supply/flex area scale`→P. 33 or `mermap/supply/flex reference scale`→P. 33 which determines the zoom level implicitly. Note that a too high zoom level imposes the risk of downloading an unwanted high quantity of map tiles resulting in a much too large document map. Therefore, this most obvious method is *not recommended* for the beginner and may be explored after some experience.

4.3 Map Supply

`\mrcsupplymap[`⟨*options*⟩`]{`⟨*definition*⟩`}`

The ⟨*options*⟩ provide parameters for the Python 3 script to supply all materials for a map. All options share the common prefix `mermap/supply/`.
The map is identified by
⟨*id*⟩=`mermap/definition prefix`+⟨*definition*⟩
for later drawing. This identifier ⟨*id*⟩ has to be unique for the document. It corresponds to generated files ⟨*id*⟩`.def`, ⟨*id*⟩`.md5`, and possibly ⟨*id*⟩`.png`. Do not use spaces or special characters like umlauts for ⟨*definition*⟩.
If `\mrcactivatescript` → P. 27 is used inside the preamble, `\mrcsupplymap` executes the Python 3 script, otherwise nothing happens.

`\mermapsetsupply{`⟨*options*⟩`}`

Sets ⟨*options*⟩ for all following maps inside the current TEX group. All options share the common prefix `mermap/supply/`, e.g. for setting `mermap/supply/type` use

```
\mermapsetsupply{type=reference}
```

Also see `\mermapset` → P. 17 and `\mermapsetmarker` → P. 59.

`mermap/definition prefix=`⟨*definition prefix*⟩ (no default, initially `maps/`)

Prefix for map identifiers and generated map files, see `\mrcsupplymap` and `\mrcapplymap` → P. 35. Note that `mermap/definition prefix` is not to be used inside the option list for `\mrcsupplymap`.

`mermap/supply/type=`⟨*type*⟩ (no default, initially `reference`)

The ⟨*type*⟩ defines the basic computation for the map. Feasible values are

- `reference`: «map with reference position»
  The map is constructed from a given reference position `mermap/supply/latitude` → P. 31,
  `mermap/supply/longitude` → P. 31,
  a zoom level `mermap/supply/zoom` → P. 30,
  map dimensions
  `mermap/supply/width` → P. 31,
  `mermap/supply/height` → P. 31,
  and alignment `mermap/supply/align` → P. 31.
- `areafit`: «map fitting an area»
  The map is constructed from a given area boundaries
  `mermap/supply/west` → P. 30,
  `mermap/supply/east` → P. 30,
  `mermap/supply/north` → P. 30,
  `mermap/supply/south` → P. 30,
  map dimensions
  `mermap/supply/width` → P. 31,
  `mermap/supply/height` → P. 31,
  and alignment `mermap/supply/align` → P. 31.
- `boundaries`: «map with boundaries»
  The map is constructed from given boundaries
  `mermap/supply/west` → P. 30,
  `mermap/supply/east` → P. 30,
  `mermap/supply/north` → P. 30,
  `mermap/supply/south` → P. 30,
  and zoom level `mermap/supply/zoom` → P. 30.

mermap/supply/zoom=⟨*setup zoom*⟩ (no default, initially 9)

Map tile zoom factor alias z coordinate of the map tiles. Used for map types **boundaries** and **reference**.

mermap/supply/north=⟨*setup north latitude*⟩ (no default, initially 50)

Northern latitude degree, possibly negative for the southern hemisphere, lower than 90 but always larger than mermap/supply/south. Used for map types **boundaries** and **areafit**.

mermap/supply/south=⟨*setup south latitude*⟩ (no default, initially 48)

Southern latitude degree, possibly negative for the southern hemisphere, larger than −90 but always lower than mermap/supply/north. Used for map types **boundaries** and **areafit**.

mermap/supply/west=⟨*setup west longitude*⟩ (no default, initially 11)

Western longitude degree, possibly negative for the western hemisphere, possibly shifted periodically, but always lower than mermap/supply/east. Used for map types **boundaries** and **areafit**.

mermap/supply/east=⟨*setup east longitude*⟩ (no default, initially 13)

Eastern longitude degree, possibly negative for the western hemisphere, possibly shifted periodically, but always larger than mermap/supply/west. Used for map types **boundaries** and **areafit**.

mermap/supply/area={⟨*comma separated list of named positions*⟩} (no default)
mermap/supply/add area={⟨*comma separated list of named positions*⟩} (no default)

Sets mermap/supply/north, mermap/supply/south, mermap/supply/west, mermap/supply/east according to the given ⟨*comma separated list of named positions*⟩, i.e. the described area contains all these positions.

mermap/supply/area resets the current area which requires at least two points inside the list.

mermap/supply/add area does not reset the current area, i.e. the positions are added to the current area which possibly grows to fit all positions.

Also note to take special care, if the international dateline is on your resulting map, see Section 11.2 on page 86. Used for map types **boundaries** and **areafit** or in combination with mermap/supply/area to reference also for for map type **reference**.

N 2020-05-08 **mermap/supply/area from marker input**={⟨*file name*⟩} (no default)
N 2020-05-08 **mermap/supply/add area from marker input**={⟨*file name*⟩} (no default)

Sets mermap/supply/north, mermap/supply/south, mermap/supply/west, mermap/supply/east according to the given \mrcmarker→ P. 59 positions contained in a file with the given ⟨*file name*⟩.

mermap/supply/area from marker input resets the current area which requires at least two marker positions inside the file.

mermap/supply/add area from marker input does not reset the current area, i.e. the positions are added to the current area which possibly grows to fit all positions.

Also note to take special care, if the international dateline is on your resulting map, see Section 11.2 on page 86. Used for map types **boundaries** and **areafit** or in combination with mermap/supply/area to reference also for for map type **reference**.

mermap/supply/area to reference (no value, initially unset)

The map settings mermap/supply/north, mermap/supply/south, mermap/supply/west, mermap/supply/east are taken to compute the map center. This center position is saved to mermap/supply/latitude→ P. 31 and mermap/supply/longitude→ P. 31. Used for map type **reference**.

`mermap/supply/latitude`=⟨*setup latitude*⟩ (no default, initially `49`)

Latitude degree of a reference point, possibly negative for the southern hemisphere. Used for map type `reference`.

`mermap/supply/longitude`=⟨*setup longitude*⟩ (no default, initially `12`)

Longitude degree of a reference point, possibly negative for the western hemisphere. Used for map type `reference`.

`mermap/supply/position`=⟨*setup latitude*⟩`:`⟨*setup longitude*⟩ (no default, initially `49:12`)

Latitude degree and longitude of a reference point. Used for map type `reference`.

`mermap/supply/named position`=⟨*name*⟩ (style, no default)

The *named position* given by ⟨*name*⟩ describes a reference point, see Section 3.5 on page 22. Used for map type `reference`.

`mermap/supply/width`=⟨*setup width in tiles*⟩ (no default, initially `4`)

Width of the map as multiplicity of map tiles. Used for map types `reference` and `areafit`.

`mermap/supply/tex width`=⟨*width*⟩ (style, no default)

Width of the map as TEX dimension. This is a style to compute `mermap/supply/width` according to the current `mermap/tile size`→ P. 48.

`mermap/supply/height`=⟨*setup height in tiles*⟩ (no default, initially `4`)

Height of the map as multiplicity of map tiles. Used for map types `reference` and `areafit`.

`mermap/supply/tex height`=⟨*width*⟩ (style, no default)

Height of the map as TEX dimension. This is a style to compute `mermap/supply/height` according to the current `mermap/tile size`→ P. 48.

`mermap/supply/align`=⟨*setup alignment*⟩ (no default, initially `center`)

Alignment of reference point or area for map types `reference` and `areafit`. Feasible values are `northwest`, `north`, `northeast`, `west`, `center`, `east`, `southwest`,`south`, `southeast`.

`mermap/supply/target`=⟨*setup target*⟩ (no default, initially `tiles`)

Defines the type of output for the Python 3 script. Feasible values are:

- `none`: No tiles are downloaded and no merged map is generated, just map computation. This is the fastest method and needs no tile supplier.
- `tiles`: Download map tiles from a tile map service (TMS) `mermap/supply/url`→ P. 32. Compilation of a document with map tile takes longer than compilation with a merged map and transparency should not be used with tiles, but the resulting document is smaller than a document with merged maps.
- `mergedmap`: Download map tiles from a tile map service (TMS) `mermap/supply/url`→ P. 32 and merge them into a single map picture. This speeds compilation and allows transparency effects, but the resulting document is possibly larger than a document with map tiles, because map tiles often are optimized 8-bit image files while the merged image is a 24-bit PNG file. Additionally, synergy effects of using the same map tiles for different maps are lost. Also, since the pixel map is clipped to full pixels, the resulting map may differ (shift/size) from the more accurate tile representation by one pixel.
- `wmsmap`: Download a single map from a web map service (WMS) `mermap/supply/url`→ P. 32. Internally, the package treats a WMS like a tile map service including all tile calculations. Actually, a single file is downloaded.

N 2020-08-06 `mermap/fail on missing resource`=`true|false` (default `true`, initially `true`)

If set to `true`, compilation stops with an error, if `mermap/supply/target` and `mermap/mapdef/resource`→ P. 19 are different. Typically, this means that something went wrong while trying to download map tiles. Set this option temporarily to `false`, if the map tile service or the internet connection is expected to be unavailable only temporarily.

`mermap/supply/url`=⟨*setup URL*⟩ (no default, initially empty)

Here, the url format with placeholder `{z}{x}{y}` for map tile download is defined. **Be sure that you have the permission to download, save, and use the map tiles from that URL. Illegal downloads are not endorsed in any way.**

```
url={https://abc.efg.hij/{z}/{x}/{y}.png?apikey=12345678},
```

See Section 4.5 on page 36 for predefined URLs.

`mermap/supply/url with api key`={⟨*prefix*⟩}{⟨*name*⟩}{⟨*postfix*⟩} (no default)

This is an alternative version of `mermap/supply/url`. The URL is constructed from some fixed ⟨*prefix*⟩ and ⟨*postfix*⟩ with an API key in between. The API key is retrieved by ⟨*name*⟩ from a repository filled by `\mrcsetapikey`.

```
url={https://abc.efg.hij/{z}/{x}/{y}.png?apikey=}{myservice}{},
```

See Section 4.5 on page 36 for predefined URLs.

`\mrcsetapikey`{⟨*name*⟩}{⟨*value*⟩}

Stores an API key ⟨*value*⟩ for access with the given ⟨*name*⟩. Typically, ⟨*value*⟩ is a received ID from a map tile service provider after personal registration. ⟨*name*⟩ is a placeholder which is used inside `mermap/supply/url with api key` to mark the insertion point for the API key.

```
\mrcsetapikey{myservice}{....K942XY....}
```

`mermap/supply/attribution`=⟨*attribution text*⟩ (no default, initially empty)

Attribution text for the map source. Typically, it acknowledges the copyright of the map data provider. It may contain hyperlinks. It is used to set up `mermap/mapdef/attribution` → P. 19 afterwards and it is accessible as `\mrcmapattribution` (use read-only).

For technical reasons, do not use `"`. `\mrcumlaut` may be used for masking umlauts, e.g. use `\mrcumlaut{u}` instead of `\"{u}`, but umlauts can also be used directly, e.g. as UTF-8 coded characters.

`mermap/supply/attribution print`=⟨*attribution text*⟩ (no default, initially empty)

Attribution text for the map source. In contrast to `mermap/supply/attribution` it is intended for media that does not support hyperlinks like printed posters, books, etc. It is used to set up `mermap/mapdef/attribution print` → P. 19 afterwards and it is accessible as `\mrcmapattributionprint` (use read-only).

`mermap/supply/basename`=⟨*setup tile base name*⟩ (no default, initially `tiles/tile`)

Prefix for local tile files, e.g. '`tiles/map`' for '`tiles/map_10_10_10.png`'.

`mermap/supply/flex reference scale`=⟨*scale denominator*⟩ (no default)

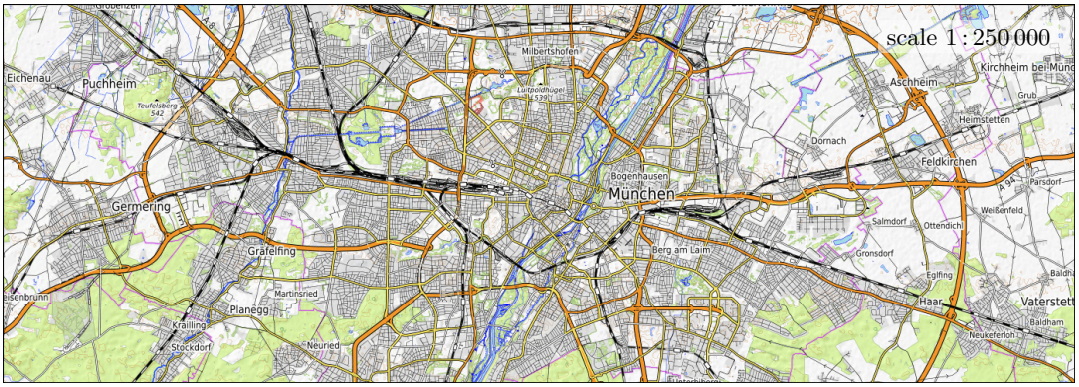
With the given ⟨*scale denominator*⟩, an appropriate `mermap/supply/zoom` → P. 30 and `mermap/tile size` → P. 48 is computed. Note that the ⟨*scale denominator*⟩ always applies to the current `mermap/supply/latitude` → P. 31 and is used for map type **boundaries** and **reference**. For example, if the reference point is on the north side of the map, also the ⟨*scale denominator*⟩ applies to the most northern latitude.

Note to take special care to the order of the options.

- The reference point has to be set *before* `mermap/supply/flex reference scale`, e.g. by `mermap/supply/latitude` → P. 31, `mermap/supply/position` → P. 31, `mermap/supply/named position` → P. 31.
- `mermap/supply/tex height` → P. 31, `mermap/supply/tex width` → P. 31 (only for map type **reference**) have to be set *after* `mermap/supply/flex reference scale`, because the `mermap/tile size` → P. 48 is adapted.

Also see `mermap/flex tile size` → P. 48, `mermap/flex zoom` → P. 48, and `mermap/flex scale` → P. 49.

```
\begin{tikzpicture}
  \mrcmap[type=reference,latitude=48.14,longitude=11.57,
    flex reference scale=250000,
    source=opentopomap,
    tex width=\linewidth,tex height=5cm]{}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
       {\mrcmapattribution};
  \mrcclipmap
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
  \node[below left=2mm,align=right,fill=white,fill opacity=0.5,
    text opacity=1] at (mrcmap.north east) {scale \mrcprettymapscale};
\end{tikzpicture}
```



Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)

`mermap/supply/flex area scale`=⟨*scale denominator*⟩ (no default)

This is a shortcut for setting `mermap/supply/area to reference` → P. 30 and `mermap/supply/flex reference scale`=⟨*scale denominator*⟩. Used for map type **boundaries** and **reference**.

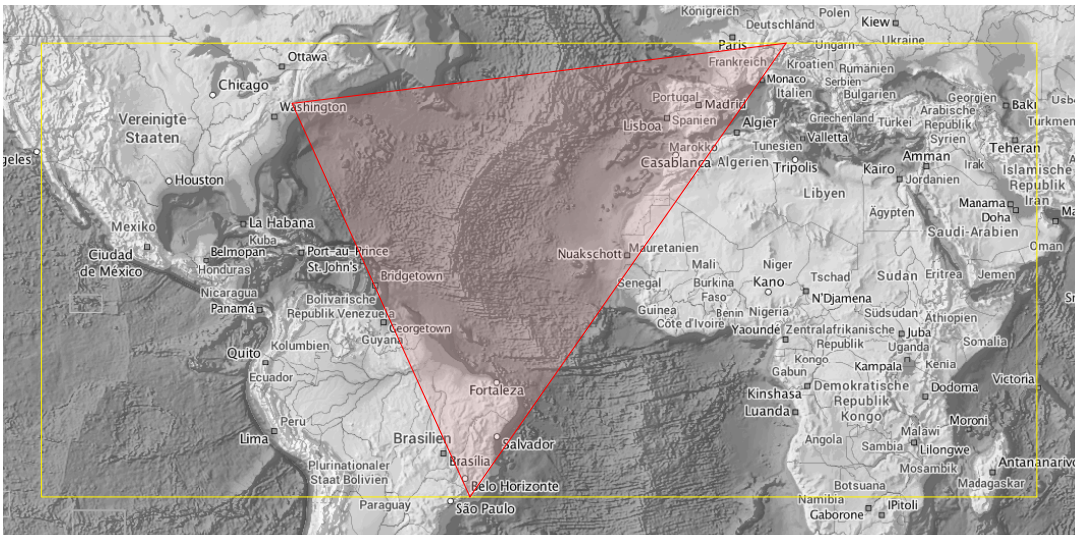
Note to take special care to the order of the options.

- The reference point has to be set *before* `mermap/supply/flex area scale`.
- `mermap/supply/tex height` → P. 31, `mermap/supply/tex width` → P. 31 (only for map type **reference**) have to be set *after* `mermap/supply/flex reference scale`.

`mermap/supply/flex area fit=`⟨*size*⟩ (default `0pt`)

This key can be used for map type `areafit` as *final* option *after* all other options. It applies a fine tuning to `mermap/tile size` → P. 48, `mermap/supply/width` → P. 31, and `mermap/supply/height` → P. 31 such that the defined area fits exactly into the map region. If a ⟨*size*⟩ is specified, width and height are reduced for the calculation by this ⟨*size*⟩, e.g. ⟨*size*⟩`=1cm` ensures a border of `5mm` on each side. Also see `mermap/flex tile size` → P. 48 and `mermap/flex zoom` → P. 48.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.14}{11.58}
  \mrcNPdef{rio}{-22.91}{-43.20}
  \mrcNPdef{newyork}{40.71}{-74.01}
  \mrcmap[  type = areafit, area = {munich,rio,newyork},
    source=topplusopen web grau,
    tex width=\linewidth, tex height=7cm,
    flex area fit=1cm ]{}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
       {\mrcmapattribution};
  \draw[yellow] ([xshift=5mm,yshift=5mm]mrcmap.south west) rectangle
                ([xshift=-5mm,yshift=-5mm]mrcmap.north east);
  \draw[red,fill=red!50!gray!50!white,fill opacity=0.25]
    (\mrcNPcs{newyork}) -- (\mrcNPcs{rio}) -- (\mrcNPcs{munich}) -- cycle;
\end{tikzpicture}
```



Kartendarstellung: © BKG (2024), Datenquellen

`mermap/supply/pixel`=⟨*setup pixel size*⟩ (no default, initially 256)

Pixel width (and height) of a tile. It is especially needed for target `mergedmap` and also `wmsmap`. For `wmsmap`, it is multiplied with a pseudo tile calculation to compute the actual picture size to download.

`mermap/supply/dpi`=⟨*dpi value*⟩ (style, no default)

This style sets `mermap/supply/pixel` such that the given ⟨*dpi value*⟩ is resulting (approximately).

- Note that this only applies for WMS Servers and not for TMS (Tile Map Service) Servers, because there `mermap/supply/pixel` is a fixed number depending on Server settings and cannot be chosen arbitrarily.
- A high ⟨*dpi value*⟩ results in large downloaded map files. If the server does not provide a high resolution map, you will get unnecessary large files with blurred content.
- The ⟨*dpi value*⟩ for TMS data can be changed by adapting `mermap/tile size` → P. 48 or `mermap/flex tile size` → P. 48.

```
\begin{tikzpicture}
 \mrcmap[type=reference,latitude=48.14,longitude=11.57,
    flex reference scale=250000,
    source=topplusopen p250, target=wmsmap, dpi=300,
    tex width=\linewidth,tex height=5cm]{dpi_value}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
       {\mrcmapattribution};
  \mrcclipmap
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
\end{tikzpicture}
```

Kartendarstellung: © BKG (2024), Datenquellen

4.4 Map Apply

`\mrcapplymap{`⟨*definition*⟩`}`

A map which is supplied by `\mrcsupplymap` → P. 29 is applied inside a `tikzpicture` environment by `\mrcapplymap` where ⟨*definition*⟩ identifies the map. `\mrcapplymap` replaces a manual setup by `\mrcdefinemap` → P. 18. The same map can be applied more than once inside a document. Note that applying a map does not mean to draw the map, but to prepare everything for drawing.

`\mrcmap[`⟨*options*⟩`]{`⟨*definition*⟩`}`

This is a combination of `\mrcsupplymap` → P. 29 with the given ⟨*options*⟩ followed immediately by `\mrcapplymap`. If ⟨*definition*⟩ is left empty, an automated unique identifier is inserted. If a map is to be used just once, `\mrcmap` may be preferred.

4.5 Map Tile Server

To use map tiles with this package you obviously need access to a map tile server. Thanks to all the many contributors to OpenStreetMap, map data is free for everyone to use. **But, map tile servers based on OpenStreetMap are not necessarily free**.

A list of online raster tile servers based on OpenStreetMap data is found here:
https://wiki.openstreetmap.org/wiki/Raster_tile_providers

- **I do not run a map tile server.**
- **I do not and cannot grant any permission to access a map tile server.**
- **I do not and cannot grant any permission to use map tiles in private, academic, free, or commercial publications.**
- **All operators of map tile servers require to mention an attribution for their maps.**

The following option allows easy usage of very few selected tile servers. The tile server of OpenStreetMap is not included because of its Tile Usage Policy. I am aware that the following list could be enlarged much more, but I do not want to add more to avoid any legal uncertainties.

- **I will remove an entry immediately, if the tile server operator asks for it.**
- If **YOU** operate a tile server and you want an entry here, I would be glad to add it to the following list.

`mermap/supply/source`=⟨*source*⟩ (style, no default)

This style sets `mermap/supply/url` → P. 32, `mermap/supply/attribution` → P. 32, `mermap/supply/attribution print` → P. 32 and `mermap/supply/basename` → P. 32.
Feasible values for ⟨*source*⟩ are:

- `dummy`:
  Dummy tile server at loopback `127.0.0.1` for test purposes.
- `opentopomap`:
  Tile server (TMS) of OpenTopoMap.
  - Usage (German language): https://opentopomap.org/about#verwendung
  - Required attribution (`\mrcmapattribution`):

    Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)

  - Required attribution for media without hyperlinks (`\mrcmapattributionprint`):

    Kartendaten: © www.openstreetmap.org/copyright, SRTM | Kartendarstellung: © opentopomap.org (CC-BY-SA)

- **thunderforest opencyclemap**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey → P. 32{thunderforest}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/opencyclemap
  - Required attribution (\mrcmapattribution):
    > Maps © Thunderforest, Data © OpenStreetMap contributors
  - Required attribution for media without hyperlinks (\mrcmapattributionprint):
    > Maps © www.thunderforest.com, Data © www.osm.org/copyright

Maps © Thunderforest, Data © OpenStreetMap contributors

- **thunderforest transport**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey → P. 32{thunderforest}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/transport
  - Required attribution (\mrcmapattribution):
    > Maps © Thunderforest, Data © OpenStreetMap contributors
  - Required attribution for media without hyperlinks (\mrcmapattributionprint):
    > Maps © www.thunderforest.com, Data © www.osm.org/copyright

Maps © Thunderforest, Data © OpenStreetMap contributors

- **thunderforest landscape**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey → P. 32{thunderforest}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/landscape
  - Required attribution (\mrcmapattribution):
    > Maps © Thunderforest, Data © OpenStreetMap contributors
  - Required attribution for media without hyperlinks (\mrcmapattributionprint):
    > Maps © www.thunderforest.com, Data © www.osm.org/copyright

Maps © Thunderforest, Data © OpenStreetMap contributors

- **thunderforest outdoors**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey→ P. 32{**thunderforest**}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/outdoors
  - Required attribution (**\mrcmapattribution**):
    Maps © Thunderforest, Data © OpenStreetMap contributors
  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):
    Maps © www.thunderforest.com, Data © www.osm.org/copyright

Maps © Thunderforest, Data © OpenStreetMap contributors

N 2024-07-29

- **thunderforest atlas**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey→ P. 32{**thunderforest**}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/atlas
  - Required attribution (**\mrcmapattribution**):
    Maps © Thunderforest, Data © OpenStreetMap contributors
  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):
    Maps © www.thunderforest.com, Data © www.osm.org/copyright

Maps © Thunderforest, Data © OpenStreetMap contributors

- **thunderforest transport-dark**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey→ P. 32{**thunderforest**}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/transport-dark
  - Required attribution (**\mrcmapattribution**):
    Maps © Thunderforest, Data © OpenStreetMap contributors
  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):
    Maps © www.thunderforest.com, Data © www.osm.org/copyright

Maps © Thunderforest, Data © OpenStreetMap contributors

- **thunderforest spinal-map**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey → P. 32{thunderforest}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/spinal-map
  - Required attribution (\mrcmapattribution):
    > Maps © Thunderforest, Data © OpenStreetMap contributors
  - Required attribution for media without hyperlinks (\mrcmapattributionprint):
    > Maps © www.thunderforest.com, Data © www.osm.org/copyright

Maps © Thunderforest, Data © OpenStreetMap contributors

- **thunderforest pioneer**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey → P. 32{thunderforest}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/pioneer
  - Required attribution (\mrcmapattribution):
    > Maps © Thunderforest, Data © OpenStreetMap contributors
  - Required attribution for media without hyperlinks (\mrcmapattributionprint):
    > Maps © www.thunderforest.com, Data © www.osm.org/copyright

Maps © Thunderforest, Data © OpenStreetMap contributors

- **thunderforest mobile-atlas**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey → P. 32{thunderforest}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/mobile-atlas
  - Required attribution (\mrcmapattribution):
    > Maps © Thunderforest, Data © OpenStreetMap contributors
  - Required attribution for media without hyperlinks (\mrcmapattributionprint):
    > Maps © www.thunderforest.com, Data © www.osm.org/copyright

Maps © Thunderforest, Data © OpenStreetMap contributors

- **thunderforest neighbourhood**:
  Tile server of Thunderforest.
  A registered ⟨*api-key*⟩ is needed (free plan available) which is applied by \mrcsetapikey → P. 32{thunderforest}{⟨*api-key*⟩}.
  - Usage: https://www.thunderforest.com/terms
  - API documentation: https://www.thunderforest.com/maps/neighbourhood
  - Required attribution (\mrcmapattribution):

    Maps © Thunderforest, Data © OpenStreetMap contributors

  - Required attribution for media without hyperlinks (\mrcmapattributionprint):

    Maps © www.thunderforest.com, Data © www.osm.org/copyright

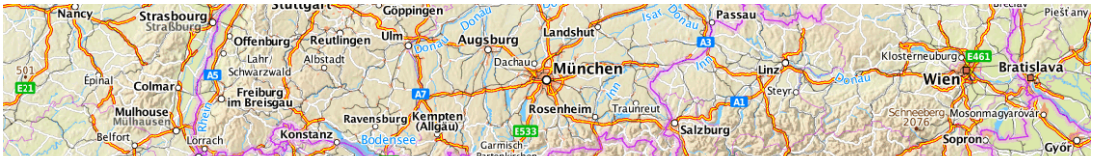
Maps © Thunderforest, Data © OpenStreetMap contributors

- **topplusopen web**:
  Tile server (TMS) of Bundesamt für Kartographie und Geodäsie.
  - Covers world / Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (\mrcmapattribution):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (\mrcmapattributionprint):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf



Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen web grau**:
  Tile server (TMS) of Bundesamt für Kartographie und Geodäsie.
  - Covers world / Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (\mrcmapattribution):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (\mrcmapattributionprint):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

N 2024-07-29

- **topplusopen web light**:
  Tile server (TMS) of Bundesamt für Kartographie und Geodäsie.
  - Covers world / Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):
    Kartendarstellung: © BKG (2024), Datenquellen
  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):
    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

N 2024-07-29

- **topplusopen web light grau**:
  Tile server (TMS) of Bundesamt für Kartographie und Geodäsie.
  - Covers world / Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):
    Kartendarstellung: © BKG (2024), Datenquellen
  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):
    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

The following sources are Web map server (WMS). Therefore, **mermap/supply/target** → P. 31 can only be set to **wmsmap** or **none**.

- `topplusopen p5`:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:5000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (`\mrcmapattribution`):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (`\mrcmapattributionprint`):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

- `topplusopen p5 grau`:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:5000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (`\mrcmapattribution`):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (`\mrcmapattributionprint`):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

- `topplusopen p10`:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:10000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (`\mrcmapattribution`):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (`\mrcmapattributionprint`):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen p10 grau**:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:10000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen p17.5**:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:17500, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen p17.5 grau**:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:17500, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen p25**:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:25000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen p25 grau**:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:25000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen p50**:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:50000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

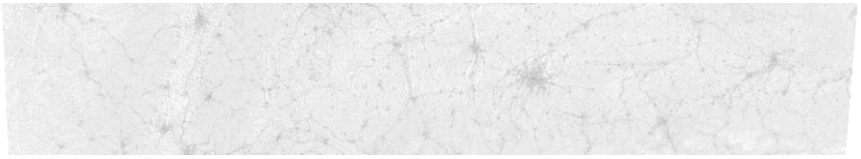
Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen p50 grau**:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:50000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf



Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen p100**:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:100000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

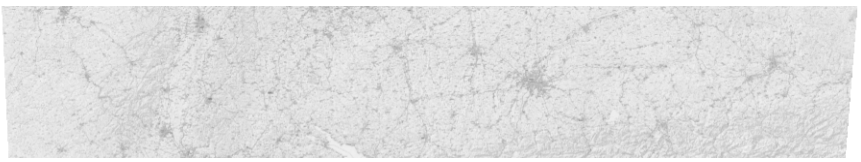
Kartendarstellung: © BKG (2024), Datenquellen

- **topplusopen p100 grau**:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:100000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (**\mrcmapattribution**):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (**\mrcmapattributionprint**):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf



Kartendarstellung: © BKG (2024), Datenquellen

- `topplusopen p250`:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:250000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (`\mrcmapattribution`):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (`\mrcmapattributionprint`):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf

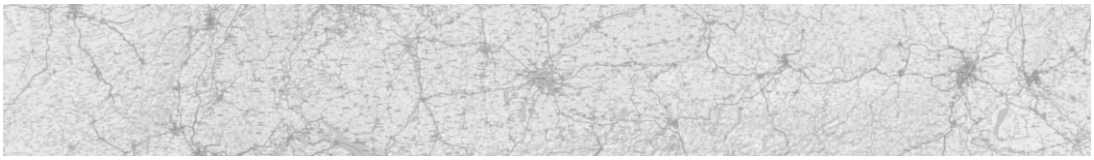
Kartendarstellung: © BKG (2024), Datenquellen

- `topplusopen p250 grau`:
  Web map server (WMS) of Bundesamt für Kartographie und Geodäsie.
  - 1:250000, covers Europe / Germany depending on zoom level
  - Usage (German language): https://gdz.bkg.bund.de/index.php/default/digitale-geodaten/topplusopen-produkte.html
  - Required attribution (`\mrcmapattribution`):

    Kartendarstellung: © BKG (2024), Datenquellen

  - Required attribution for media without hyperlinks (`\mrcmapattributionprint`):

    © BKG (2024), Datenquellen:
    https://sgx.geodatenzentrum.de/web_public/Datenquellen_TopPlus_Open.pdf



Kartendarstellung: © BKG (2024), Datenquellen

`\mrcnewsupplysource`{⟨*source*⟩}{⟨*options*⟩}

Adds a new ⟨*source*⟩ value to `mermap/supply/source` → P. 36. The ⟨*options*⟩ should set the keys `mermap/supply/url` → P. 32, `mermap/supply/attribution` → P. 32, `mermap/supply/attribution print` → P. 32 and `mermap/supply/basename` → P. 32. It is recommend to use `x ...` for ⟨*source*⟩ to avoid conflicts with future official additions to `mermap/supply/source` → P. 36.

```
\mrcnewsupplysource{x example}
{
  url              = https://127.0.0.1/dummy/{z}/{x}/{y}.png,
  attribution      = {Dummy tile server},
  attribution print = {Dummy tile server},
  basename         = tiles/dummy,
}
```

5 Map Drawing

5.1 Principal Drawing

`\mrcdrawmap[`⟨*options*⟩`]`

Inside a `tikzpicture` environment, `\mrcdrawmap` draws a map prepared by `\mrcdefinemap` → P. 18, `\mrcapplymap` → P. 35, or `\mrcmap` → P. 35. All ⟨*options*⟩ share the common prefix `mermap/`. This is the principal macro to draw a prepared map respectively the background of the map. The background consists of downloaded map tiles or just a color rectangle.

`mermap/draw=`⟨*tile draw*⟩ (no default, initially `auto`)

- `auto`: Draws the map according to `mermap/mapdef/resource` → P. 19, i.e. downloaded maps or tiles are used, if available.
- `path`: Draws the map according to the style given by `mermap/map path`. Existing map tiles or merged maps are ignored
- `tiles`: Draws the map with downloaded map tiles, if available.
- `mergedmap`: Draws the map with a merged picture, if available.
- `wmsmap`: Draws the map with a downloaded WMS picture, if available.

`mermap/map path=`⟨*options*⟩ (no default, initially `upper left=green!50, upper right=green!25, lower left=green!50!black!50, lower right=green!25`)

Defines a Ti*k*Z style for drawing the map without tiles. ⟨*options*⟩ are feasible Ti*k*Z path options.

`mermap/map clip=`⟨*code*⟩ (no default, initially `\mrcclipmap`)

Clipping options for the map. By default, the defined map is clipped with the full map rectangle. Use this option only, if you not want to clip the map to its specified size. ⟨*code*⟩ is some Ti*k*Z clipping code.

`mermap/map scope=`⟨*options*⟩ (no default, initially empty)

`\mrcdrawmap` uses a `scope` environment inside which takes the given Ti*k*Z ⟨*options*⟩.

`\mrcclipmap`

Clips all subsequent drawings against the applied map.
This is a shortcut macro identical to

```
\path[clip] (mrcmap.south west) rectangle (mrcmap.north east);
```

`\mrcboundmap`

Sets the picture bounding box according to the applied map.
This is a shortcut macro identical to

```
\path[use as bounding box]
  (mrcmap.south west) rectangle (mrcmap.north east);
```

5.2 Flexible Tile Size

Typically, the pixel size of a map tile is fixed and a map tile is a pixel graphics file. The actual size of such an included picture inside the document is freely selectable. Note that a very small `mermap/tile size` results in very small map lettering, while a very large `mermap/tile size` results results in very blurred images.

The general idea of a *flexible* tile size is to specify an aspired tile size called `mermap/flex tile size` and to give LaTeX the freedom to select `mermap/tile size` in *about the same size* as `mermap/flex tile size`. This freedom is used to achieve a *pseudo zoom* called `mermap/flex zoom` which is a nearly arbitrary rational number instead of `mermap/supply/zoom` → P. 30 which is a natural number.

This *pseudo zoom* is applied by several options which share *flex* in their names, e.g. `mermap/flex scale` → P. 49, `mermap/named flex scale` → P. 49, `mermap/supply/flex reference scale` → P. 33, `mermap/supply/flex area scale` → P. 33, `mermap/supply/flex area fit` → P. 34.

`mermap/tile size`=⟨*length*⟩ (no default, initially `32.512mm`)

Width and height of a drawn tile picture are set to ⟨*length*⟩. For standard tiles with 256 times 256 pixels a tile size of 32.512 mm = 1.28 in results in an approximate 200 dpi output for the document. For a `beamer` document, consider to use a `mermap/tile size` of 21.333 333 mm to get approximate 1:1 pixel input and output (depending on `beamer` settings and used hardware). Also see `mermap/mapdef/tile size` → P. 19.

`mermap/flex tile size`=⟨*length*⟩ (no default, initially `32.512mm`)

Aspired width and height of a tile picture are set to ⟨*length*⟩. This value is used while applying `mermap/flex zoom`.

`mermap/flex zoom`=⟨*pseudo zoom*⟩ (style, no default)

This style sets `mermap/supply/zoom` → P. 30 and `mermap/tile size` in combination.

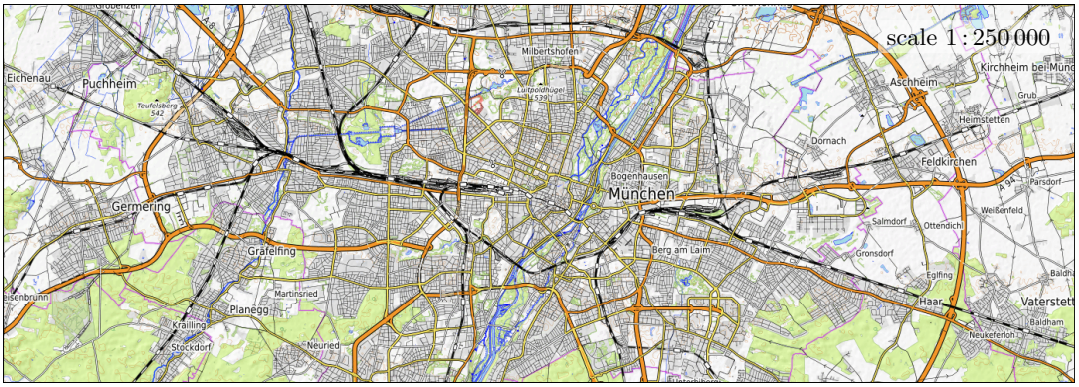
- If ⟨*pseudo zoom*⟩ is a natural number, `mermap/supply/zoom` → P. 30 is set to ⟨*pseudo zoom*⟩ and `mermap/tile size` is set to `mermap/flex tile size`.
- Otherwise, `mermap/supply/zoom` → P. 30 is set to the natural number closest to ⟨*pseudo zoom*⟩ and `mermap/tile size` is such enlarged or reduced that the ⟨*pseudo zoom*⟩ value is simulated, i.e. the *impression* of a rational zoom factor is given.

Note that `mermap/flex zoom` has to be used *before* `\mrcsupplymap` → P. 29 or `\mrcmap` → P. 35, because the zoom setup is adapted.

`mermap/flex scale`=⟨*scale denominator*⟩:⟨*latitude*⟩ (style, no default)

For different latitude scopes, an identical zoom factor produces maps of different scale. With `mermap/flex scale`, a `mermap/flex zoom` → P. 48 is computed to achieve the given ⟨*scale denominator*⟩ at a given ⟨*latitude*⟩. Note that this only applies to the center of a map. If the produced map is not centered at ⟨*latitude*⟩, the produced scale may differ from the intended one. Also see `mermap/supply/flex reference scale` → P. 33.

```
\begin{tikzpicture}
  \mermapset{flex scale=250000:48.14}
  \mrcmap[type=reference,latitude=48.14,longitude=11.57,
    source=opentopomap,
    tex width=\linewidth,tex height=5cm]{}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
       {\mrcmapattribution};
  \mrcclipmap
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
  \node[below left=2mm, align=right, fill=white, fill opacity=0.5,
      text opacity=1]
    at (mrcmap.north east) {scale \mrcprettymapscale};
\end{tikzpicture}
```



Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)

`mermap/named flex scale`=⟨*scale denominator*⟩:⟨*name*⟩ (style, no default)

Identical to `mermap/flex scale`, but used the *named position* ⟨*name*⟩ to provide a *latitude*, see Section 3.5 on page 22.

```
\mrcNPdef{munich}{48.137222}{11.575556}
\mermapset{named flex scale=250000:munich} % identical to the following
\mermapset{flex scale=250000:\mrcNPlat{munich}}
```

5.3 Geodetic Network

`\mrcdrawnetwork[`⟨*options*⟩`]`

Draws a geodetic network with meridians and parallels. All ⟨*options*⟩ share the common prefix `mermap/`. The displayed lines are selected automatically according to some tuning parameters. The map is sliced in *about* maximal `mermap/network pieces` in each direction. Meridians and parallels share a minimal distance of *about* `mermap/network distance`. The algorithm is allowed to violate these conditions *somewhat*.
Note that oversized maps are not supported, i.e. maps which are wider than 360° in longitude. Here, meridians are expected to be missing or misplaced.

```
% \mrcsetapikey{thunderforest}{YOUR-API-KEY}  % registered key
\begin{tikzpicture}
  \mrcmap[  type = boundaries,
    west = -20, east = 40, south = 36, north = 65,
    source=thunderforest landscape,
    flex area scale=40 000 000 ]{}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
       {\mrcmapattribution};
  \mrcclipmap
  \mrcdrawnetwork
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
\end{tikzpicture}
```

Maps © Thunderforest, Data © OpenStreetMap contributors

`mermap/network pieces=`⟨*number*⟩ (no default, initially `8`)

The map is sliced in *about* maximal ⟨*number*⟩ pieces in each direction. ⟨*number*⟩ may be exceeded *somewhat*. It is underrun to comply with `mermap/network distance`.

`mermap/network distance=`⟨*mesh width*⟩ (no default, initially `2cm`)

Meridians and parallels share a minimal distance of *about* ⟨*mesh width*⟩. ⟨*mesh width*⟩ may be underrun *somewhat*. It is exceeded to comply with `mermap/network pieces`. For parallels on small scale maps, it refers to an averaged mesh width.

`mermap/network font=`⟨*text*⟩ (no default, initially `\fontsize{4pt}{4pt}\sffamily`)

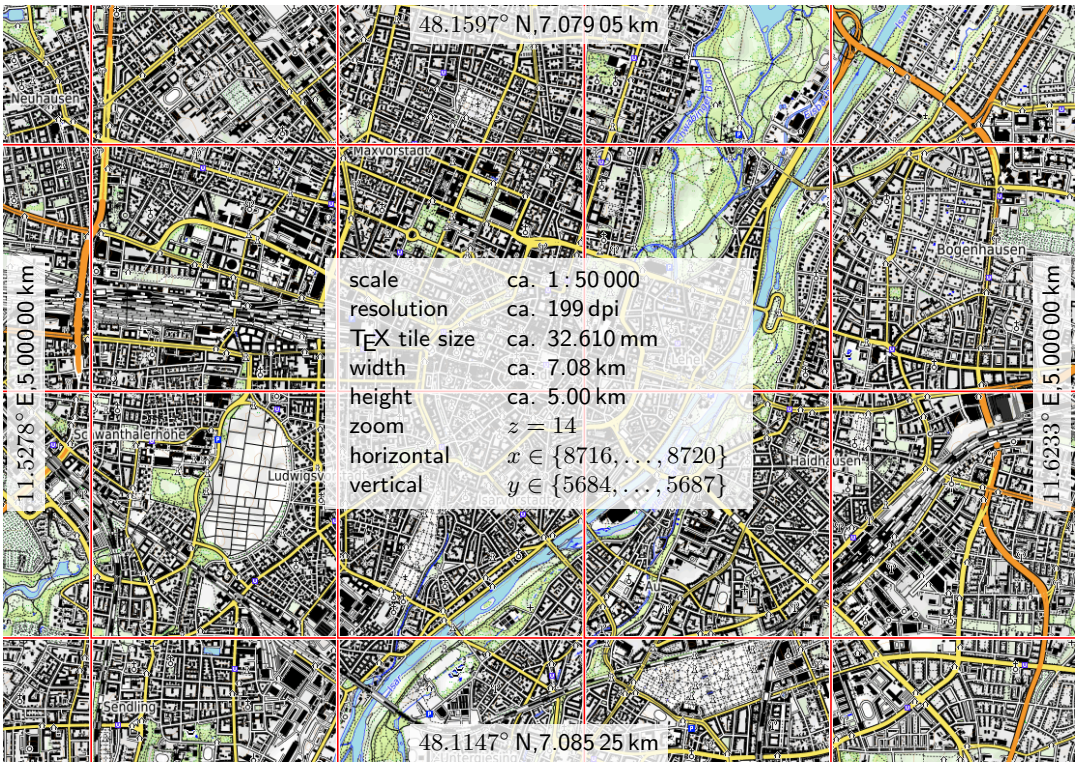
⟨*text*⟩ is some font setting for the latitude and longitude display.

5.4 Graphical Debug Overlay

`\mrcdrawinfo`

Draws some map information overlay for debugging purposes only.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.137222}{11.575556}
  \mrcmap[type=reference, named position=munich, source=opentopomap,
    flex reference scale=50000,
    tex width=\linewidth,tex height=10cm]{}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
       {\mrcmapattribution};
  \mrcclipmap
  \mrcdrawinfo
\end{tikzpicture}
```



Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)

6 Scales and Sizes

Inside a defined map several size values and scaling options are available. Please note that due to the nature of the Mercator projection and several simplifying assumptions all specifications for map scale, map width, map height, etc. are imprecise in the best case and even misleading in the worst case. They are suited for representative diagrams, but not for critical navigation purposes etc.

6.1 Map Sizes and Document Sizes

`\mrctexwidth`

TEX length denoting the document width of the current map.

`\mrctexheight`

TEX length denoting the document width of the current map.

`\mrcscale`

Scaling factor between map and real world. A TEX length given in `pt`, but stripped from that unit, multiplied by `\mrcscale` corresponds to a real world length given in kilometers. Note that this is *not* the map scale. Actually, it is reciprocal proportional to the map scale and proportional to the map scale denominator, see `\mrcmapscaledenominator`.

`\mrctextokm{`⟨*length*⟩`}`

Computes a given TEX ⟨*length*⟩ (with unit) into the corresponding real world length in kilometers (without unit).

```
Map width: \mrctextokm{\mrctexwidth}\,km
```

`\mrctextomile{`⟨*length*⟩`}`

Computes a given TEX ⟨*length*⟩ (with unit) into the corresponding real world length in miles (without unit).

```
Map width: \mrctextomile{\mrctexwidth}\,mi
```

`\mrckmtotex{`⟨*number*⟩`}`

Computes real world length ⟨*number*⟩ in kilometers (without unit) to a TEX length (with unit).

```
% draw a circle with radius 20km
\draw (mrcpos) circle (\mrckmtotex{20});
```

`\mrcmiletotex{`⟨*number*⟩`}`

Computes real world length ⟨*number*⟩ in miles (without unit) to a TEX length (with unit).

```
% draw a circle with radius 20 miles
\draw (mrcpos) circle (\mrcmiletotex{20});
```

`\mrcmapscaledenominator`

Approximate map scale denominator. 1 cm on the map corresponds approximately to `\mrcmapscaledenominator`·1 cm in the real world. Do not confuse with `\mrcscale`.

```
Map scale: 1:\mrcmapscaledenominator
```

6.2 Pretty Size Output

`\mrcprettymapscale`

Approximate map scale given with three valid digits with a representation like `1:1000`.

```
Map scale: \mrcprettymapscale
```

`\mrcprettymapwidth`

Approximate map width in kilometers (or meters) with three valid digits:

```
Map width: \mrcprettymapwidth
```

To create a pretty printing to your own liking, you can do like the following:

```
\newcommand{\myprettymapwidth}{\SI[round-mode=figures,round-precision=3]%
  {\mrctextokm{\mrctexwidth}}{km}}
```

`\mrcprettymapheight`

Approximate map height in kilometers (or meters) with three valid digits:

```
Map height: \mrcprettymapheight
```

`\mrcprettymapresolution`

Approximate map resolution in dpi (dots per inch):

```
Map resolution: \mrcprettymapresolution
```

`\mrcprettytilesize`

Approximate tile size inside the document (TEX size) in millimeters:

```
\TeX\ tile size: \mrcprettytilesize
```

```
\begin{tikzpicture}
  \mrcNPdef{rome}{41.89300}{12.48557}
  \mermapset{named flex scale=2000000:rome}
  \mrcmap[type=reference, named position=rome, source=topplusopen web,
    tex width=\linewidth, tex height=10cm]{scales_example}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
        {\mrcmapattribution};
  \mrcclipmap
  \path[draw] (\mrcNPcs{rome}) circle (\mrckmtotex{50});
  \node[above] at ([yshift=\mrckmtotex{50}]\mrcNPcs{rome}) {\SI{50}{km}};
  \node[above right,fill=white,fill opacity=0.5,text opacity=1,align=center,
    line width=0pt, inner sep=2mm]
    at (mrcmap.south west) { \begin{tabular}{rl}
      scale:           & \mrcprettymapscale\\
      map width:       & \mrcprettymapwidth\\
      map height:      & \mrcprettymapheight\\
      \TeX\ tile size: & \mrcprettytilesize\\
      resolution:      & \mrcprettymapresolution
      \end{tabular}\\
      \begin{tikzpicture}
      \mrcdrawscalebar[width-in-km=100]
      \path[every node/.style={above,inner sep=0.5mm,font=\sffamily\tiny}]
        (mrcscalebar.north west) -- (mrcscalebar.north east)
        node[pos=0]{0} node[pos=0.2]{20} node[pos=0.4]{40} node[pos=0.6]{60}
        node[pos=0.8]{80} node[pos=1]{100} node[pos=1,right,yshift=-1mm]{km};
      \end{tikzpicture}
    };
\end{tikzpicture}
```

Kartendarstellung: © BKG (2024), Datenquellen

6.3 Scale Bars

`\mrcdrawscalebar[`⟨*options*⟩`]`

Draws a scale bar according to the given ⟨*options*⟩. All ⟨*options*⟩ share the common prefix `mermap/scalebar/`. The most essential option is the *width* of the scale bar.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.137222}{11.575556}
  \mrcNPdef{vienna}{48.208333}{16.373056}
  \mrcmap[tex width=\linewidth, tex height=4cm, target=none,
    type=areafit, area={munich,vienna}, flex area fit=2cm]{}
  \mrcdrawmap
  \mrcmarker{named position=munich, contents={M\"unchen}}
  \mrcmarker{named position=vienna, contents={Wien}}
  \mrcdrawscalebar[width-in-km=100, solid,
    at={([xshift=-10mm,yshift=5mm]mrcmap.south east)},
    placement={above left},  ]
\end{tikzpicture}
```

The size, position, and appearance of the scale bar can be customized by setting the various ⟨*options*⟩. The shape of the scale bar is denoted by a TikZ node `mrcscalebar` which can be used for lettering.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.137222}{11.575556}
  \mrcNPdef{vienna}{48.208333}{16.373056}
  \mrcmap[tex width=\linewidth, tex height=4cm, target=none,
    type=areafit, area={munich,vienna}, flex area fit=2cm]{}
  \mrcdrawmap
  \mrcmarker{named position=munich, contents={M\"unchen}}
  \mrcmarker{named position=vienna, contents={Wien}}
  \mrcdrawscalebar[width-in-km=100,solid,south-west-inside=10mm;3mm ]
  \path[every node/.style={above,inner sep=0.5mm,font=\sffamily\tiny}]
    (mrcscalebar.north west) -- (mrcscalebar.north east)
    node[pos=0]{0} node[pos=0.2]{20} node[pos=0.4]{40} node[pos=0.6]{60}
    node[pos=0.8]{80} node[pos=1]{100} node[pos=1,right,yshift=-1mm]{km};
\end{tikzpicture}
```

mermap/scalebar/width-in-km=⟨*number*⟩ (no default, initially 0)

Sets the width of the scale bar to match the real word length ⟨*number*⟩ in kilometers.

mermap/scalebar/width-in-kilometer=⟨*number*⟩ (no default, initially 0)

Alias for mermap/scalebar/width-in-km

mermap/scalebar/width-in-meter=⟨*number*⟩ (no default, initially 0)

Sets the width of the scale bar to match the real word length ⟨*number*⟩ in meters.

mermap/scalebar/width-in-mile=⟨*number*⟩ (no default, initially 0)

Sets the width of the scale bar to match the real word length ⟨*number*⟩ in miles.

mermap/scalebar/width-in-yard=⟨*number*⟩ (no default, initially 0)

Sets the width of the scale bar to match the real word length ⟨*number*⟩ in yards.

mermap/scalebar/partitions=⟨*number*⟩ (no default, initially 5)

Determines the ⟨*number*⟩ of partitions for the scale bar. If ⟨*number*⟩ is set to 1, there is no partitioning.

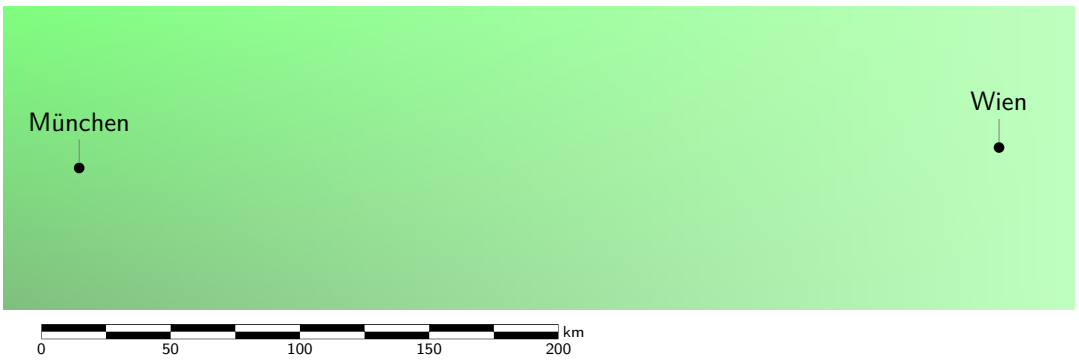
mermap/scalebar/height=⟨*length*⟩ (no default, initially 2mm)

Sets the height of the scale bar to the given TEX ⟨*length*⟩.

mermap/scalebar/at={⟨Ti*k*Z *coordinate*⟩} (no default, initially {(0,0)})

The scale bar is positioned at the given ⟨Ti*k*Z *coordinate*⟩. The placement is done with the mermap/scalebar/placement → P. 57 option. Both option correspond to the Ti*k*Z options for positioning nodes. The scale bar can be positioned outside the map (e.g. below), but remember to use \mrcclipmap → P. 47 *after* the scale bar in this case, if needed.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.137222}{11.575556}
  \mrcNPdef{vienna}{48.208333}{16.373056}
  \mrcmap[tex width=\linewidth, tex height=4cm, target=none,
    type=areafit, area={munich,vienna}, flex area fit=2cm]{}
  \mrcdrawmap
  \mrcmarker{named position=munich, contents={M\"unchen}}
  \mrcmarker{named position=vienna, contents={Wien}}
  \mrcdrawscalebar[width-in-km=200, partitions=8,
    at={([xshift=5mm,yshift=-2mm]mrcmap.south west)},
    placement=below right   ]
  \path[every node/.style={below,inner sep=0.5mm,font=\sffamily\tiny}]
    (mrcscalebar.south west) -- (mrcscalebar.south east)
    node[pos=0]{0} node[pos=0.25]{50} node[pos=0.5]{100} node[pos=0.75]{150}
    node[pos=1]{200} node[pos=1,right,yshift=1mm]{km};
\end{tikzpicture}
```



`mermap/scalebar/placement`=⟨Ti*k*Z *positioning*⟩ (no default, initially empty)

⟨Ti*k*Z *positioning*⟩ of a scale bar in combination with `mermap/scalebar/at` → P. 56. All Ti*k*Z placement options for nodes can be used, e.g. `above left` or `anchor=mid west`, etc. Actually, *any* node option could be applied here, but the intended use is for placement options only.

`mermap/scalebar/south-east-inside`=⟨*x shift*⟩;⟨*y shift*⟩ (default `0pt;0pt`, initially unset)

Shortcut for placing the scale bar at the south east corner of the map. The optional ⟨*x shift*⟩ and ⟨*y shift*⟩ denote the absolute shift values in each direction, i.e. the algebraic sign is automatically complemented. If only ⟨*x shift*⟩ is given, then ⟨*y shift*⟩ is set to the same value. `mermap/scalebar/at` → P. 56 and `mermap/scalebar/placement` are set by this option.

`mermap/scalebar/south-east-outside`=⟨*x shift*⟩;⟨*y shift*⟩ (default `0pt;0pt`, initially unset)

Shortcut for placing the scale bar below the south east corner of the map.

`mermap/scalebar/south-west-inside`=⟨*x shift*⟩;⟨*y shift*⟩ (default `0pt;0pt`, initially unset)

Shortcut for placing the scale bar at the south west corner of the map.

`mermap/scalebar/south-west-outside`=⟨*x shift*⟩;⟨*y shift*⟩ (default `0pt;0pt`, initially unset)

Shortcut for placing the scale bar below the south west corner of the map.

`mermap/scalebar/north-west-inside`=⟨*x shift*⟩;⟨*y shift*⟩ (default `0pt;0pt`, initially unset)

Shortcut for placing the scale bar at the north west corner of the map.

`mermap/scalebar/north-west-outside`=⟨*x shift*⟩;⟨*y shift*⟩ (default `0pt;0pt`, initially unset)

Shortcut for placing the scale bar above the north west corner of the map.

`mermap/scalebar/north-east-inside`=⟨*x shift*⟩;⟨*y shift*⟩ (default `0pt;0pt`, initially unset)

Shortcut for placing the scale bar at the north east corner of the map.

`mermap/scalebar/north-east-outside`=⟨*x shift*⟩;⟨*y shift*⟩ (default `0pt;0pt`, initially unset)

Shortcut for placing the scale bar above the north east corner of the map.

`mermap/scalebar/major style`={⟨Ti*k*Z *options*⟩} (no default, initially empty)

The *major* part of the scale bar is a single Ti*k*Z path object which can be customized by the given ⟨Ti*k*Z *options*⟩. The *major* part consists of the black area in the default case.

```
\begin{tikzpicture}
  \mrcdrawscalebar[scale=2000000, width-in-km=100,
      major style={left color=red,right color=blue}   ]
\end{tikzpicture}
```

`mermap/scalebar/minor style`={⟨Ti*k*Z *options*⟩} (no default, initially empty)

The *minor* part of the scale bar is a single Ti*k*Z path object which can be customized by the given ⟨Ti*k*Z *options*⟩. The *minor* part is seen as holes in the default case. `mermap/scalebar/minor style` has only an effect, if the minor part is drawn `mermap/scalebar/solid` → P. 58.

```
\begin{tikzpicture}
  \mrcdrawscalebar[scale=2000000, width-in-km=100, solid,
      minor style={yellow}  ]
\end{tikzpicture}
```

`mermap/scalebar/double=true|false` (default `true`, initially `true`)

If set to `true`, the scale bar is drawn as a double ruler.

`mermap/scalebar/single=true|false` (default `true`, initially `false`)

If set to `true`, the scale bar is drawn as a single ruler. `mermap/scalebar/single` is inverse to `mermap/scalebar/double`.

```
\begin{tikzpicture}
  \mrcdrawscalebar[scale=2000000, width-in-km=100, single, height=1mm]
\end{tikzpicture}
```


`mermap/scalebar/transparent=true|false` (default `true`, initially `true`)

If set to `true`, the *minor* part of the scale bar is drawn transparent, i.e. as holes inside the ruler.

`mermap/scalebar/solid=true|false` (default `true`, initially `false`)

If set to `true`, the *minor* part of the scale bar is drawn opaque. It is drawn white or according to `mermap/scalebar/minor style` → P. 57. `mermap/scalebar/solid` is inverse to `mermap/scalebar/transparent`.

`mermap/scalebar/scale=`⟨*scale denominator*⟩ (no default, initially unset)

Sets or overwrites the ⟨*scale denominator*⟩ setting. **Using this key is not needed and may even lead to erroneous displays inside a `tikzpicture` with a defined map setting.** This key is helpful, if a scale bar is used without a defined map.

7 Markers

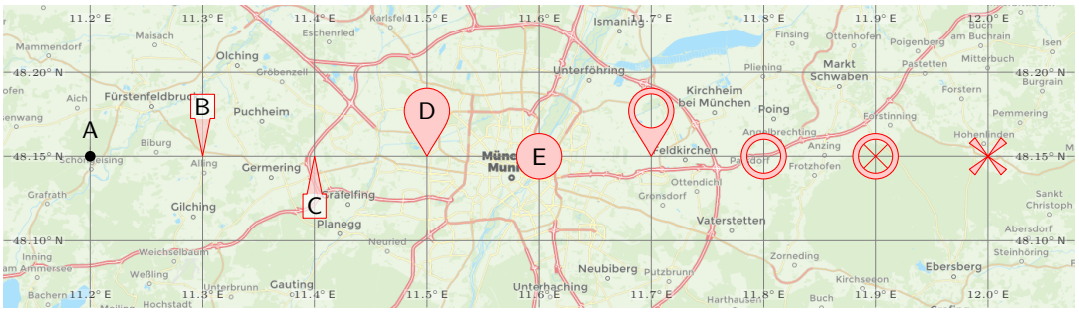
As described before, a map can be amended by arbitrary Ti*k*Z code using map coordinates. For highlighting places or adding markers, the `\mrcmarker` macro may be helpful which provides some predefined marker types.

7.1 Marker Settings

`\mrcmarker{`⟨*options*⟩`}`

Places a marker according to the given ⟨*options*⟩ on the map. All ⟨*options*⟩ share the common prefix `mermap/marker/`. Different `mermap/marker/type` → P. 63 settings are available which are more or less customizable.

```
% \mrcsetapikey{thunderforest}{YOUR-API-KEY} % registered key
\begin{tikzpicture}
  \mrcmap[type=reference, position=48.15:11.6,
    flex reference scale=500 000,
    source=thunderforest neighbourhood,
    tex width=\linewidth,tex height=4cm]{}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
    {\mrcmapattribution};
  \mrcdrawnetwork
  \mrcclipmap
  \mermapsetmarker{draw=red, fill=red!20!white, font=\sffamily\footnotesize}
  \mrcmarker{type=classic, position=48.15:11.2, contents={A}, radius=0.5mm}
  \mrcmarker{type=pin,     position=48.15:11.3, contents={B}}
  \mrcmarker{type=pinflip, position=48.15:11.4, contents={C}}
  \mrcmarker{type=drop,    position=48.15:11.5, contents={D}}
  \mrcmarker{type=knob,    position=48.15:11.6, contents={E}}
  \mrcmarker{type=pictodropring, position=48.15:11.7, contents={F}}
  \mrcmarker{type=pictoknobring, position=48.15:11.8, contents={G}}
  \mrcmarker{type=ringx,   position=48.15:11.9, contents={H}}
  \mrcmarker{type=markx,   position=48.15:12,   contents={I}}
\end{tikzpicture}
```



`\mermapsetmarker{`⟨*options*⟩`}`

Sets ⟨*options*⟩ for all following markers inside the current TEX group. All options share the common prefix `mermap/marker/`, e.g. for setting `mermap/marker/type` → P. 63 use

```
\mermapsetmarker{type=pin}
```

Also see `\mermapset` → P. 17 and `\mermapsetsupply` → P. 29.

mermap/marker/first options=⟨*options*⟩ (no default, initially unset)

The given list of ⟨*options*⟩ is used inside every \mrcmarker→ P. 59 *before* the options of \mrcmarker→ P. 59.

mermap/marker/last options=⟨*options*⟩ (no default, initially unset)

The given list of ⟨*options*⟩ is used inside every \mrcmarker→ P. 59 *after* the options of \mrcmarker→ P. 59.

mermap/marker/latitude=⟨*latitude*⟩ (no default, initially 12)

Latitude degree of the place marker. It is accessible as **\mrcmarkerlatitude** (use read-only).

mermap/marker/lat=⟨*latitude*⟩ (no default, initially 12)

Alias for mermap/marker/latitude.

mermap/marker/longitude=⟨*longitude*⟩ (no default, initially 49)

Longitude degree of the place marker. It is accessible as **\mrcmarkerlongitude** (use read-only).

mermap/marker/lon=⟨*longitude*⟩ (no default, initially 49)

Alias for mermap/marker/longitude.

mermap/marker/position=⟨*place latitude*⟩:⟨*place longitude*⟩ (no default, initially 12:49)

Sets the latitude degree and the longitude degree of the place marker.

mermap/marker/named position=⟨*name*⟩ (style, no default)

Sets the latitude degree and the longitude degree of the place marker to the *named position* denoted by ⟨*name*⟩, see Section 3.5 on page 22.

mermap/marker/use inside=⟨*area*⟩ (no default, initially map)

The place marker is used or ignored according to its belonging inside the given ⟨*area*⟩. Feasible values for ⟨*area*⟩ are:

- **map**: Use inside map.
- **vicinity**: Use inside map plus vicinity, see mermap/vicinity→ P. 23.

mermap/marker/contents=⟨*text*⟩ (no default, initially empty)

Sets ⟨*text*⟩ for displaying inside the marker, if the marker type supports such a thing. It is accessible as **\mrcmarkercontents** (use read-only).

mermap/marker/pictocontents=⟨*code*⟩ (no default, initially empty)

Sets Ti*k*Z ⟨*code*⟩ for displaying inside the marker, if the marker type supports such a thing. It is accessible as **\mrcmarkerpictocontents** (use read-only).

mermap/marker/alias=⟨*text*⟩ (no default, initially noname)

The position of the marker is available as Ti*k*Z coordinate by the given ⟨*text*⟩, e.g. to draw to or from the marker.

U 2020-05-04 **mermap/marker/uuid**=⟨*uuid*⟩ (no default, initially empty)

Sets a ⟨*uuid*⟩ for unique identification of markers. It is accessible as **\mrcmarkeruuid** (use read-only) and **\l_mermap_marker_uuid_tl** (use read-only). The ⟨*uuid*⟩ is provided for user applications.

N 2020-05-04 **mermap/marker/generic**=⟨*text*⟩ (no default, initially empty)

Sets ⟨*text*⟩ as generic content for user applications. It is accessible as **\mrcmarkergeneric** (use read-only) and **\l_mermap_marker_generic_tl** (use read-only).

mermap/marker/category=⟨*category*⟩ (no default, initially empty)

Sets a ⟨*category*⟩ to group markers. It is accessible as **\mrcmarkercategory** (use read-only).

`mermap/marker/show=true|false` (default `true`, initially `true`)

If set to `true`, the marker is shown, if it lies inside the map (or vicinity). Otherwise, the place marker is not used.

`mermap/marker/hide=true|false` (default `true`, initially `false`)

If set to `true`, the marker is not used. `mermap/marker/hide` is inverse to `mermap/marker/show`.

`mermap/marker/show category=`⟨*category*⟩ (style, no default)

Sets `mermap/marker/show` to `true`, if `mermap/marker/category` → P. 60 equals ⟨*category*⟩. Otherwise, nothing happens. `mermap/marker/category` → P. 60 has to be set *before*.

`mermap/marker/show all but category=`⟨*category*⟩ (style, no default)

Sets `mermap/marker/show` to `true`, if `mermap/marker/category` → P. 60 does not equal ⟨*category*⟩. Otherwise, nothing happens. `mermap/marker/category` → P. 60 has to be set *before*.

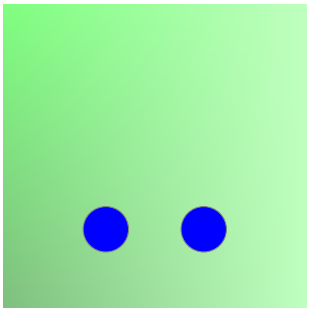
`mermap/marker/hide category=`⟨*category*⟩ (style, no default)

Sets `mermap/marker/show` to `false`, if `mermap/marker/category` → P. 60 equals ⟨*category*⟩. Otherwise, nothing happens. `mermap/marker/category` → P. 60 has to be set *before*.

`mermap/marker/hide all but category=`⟨*category*⟩ (style, no default)

Sets `mermap/marker/show` to `false`, if `mermap/marker/category` → P. 60 does not equal ⟨*category*⟩. Otherwise, nothing happens. `mermap/marker/category` → P. 60 has to be set *before*.

```
\begin{tikzpicture}
  \mrcmap[tex width=4cm, tex height=4cm,
    latitude=48.14, longitude=11.57, target=none]{}
  \mrcdrawmap[draw=path]
  \mermapsetmarker{last options={hide all but category=A}}
  \mrcmarker{type=knob,position=48.00:11.43,fill=blue,category=A}
  \mrcmarker{type=knob,position=48.28:11.43,fill=red,category=B}
  \mrcmarker{type=knob,position=48.00:11.71,fill=blue,category=A}
  \mrcmarker{type=knob,position=48.28:11.71,fill=red,category=B}
\end{tikzpicture}
```



The following options require the package **hyperref** to be loaded. If **hyperref** is not loaded, mermap/marker/url and mermap/marker/link are ignored.
Note that if mermap/marker/url and mermap/marker/link are applied concurrently, then mermap/marker/link will win.

N 2020-05-04 **mermap/marker/url**=⟨*URL*⟩ (no default, initially empty)

Sets ⟨*URL*⟩ as external link of the marker. The package is required for this option. For an own marker type, it is applied as Ti*k*Z option **mrchyperpath**.

```
\begin{tikzpicture}
  \mrcNPdef{unibwm}{48.0802826}{11.6381048}
  \mrcmap[type=reference, named position=unibwm,
    flex reference scale=100000,
    source=topplusopen web grau,
    tex width=\linewidth,tex height=4cm]{}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
    {\mrcmapattribution};
  \mrcdrawnetwork
  \mrcclipmap
  \mrcmarker{ named position = unibwm,
    type     = pin,
    draw     = red,
    fill     = red!20!white,
    url      = {https://www.unibw.de},
    contents = {Universit\"at der Bundeswehr M\"unchen},
  }
\end{tikzpicture}
```

Kartendarstellung: © BKG (2024), Datenquellen

N 2020-05-04 **mermap/marker/link**=⟨*name*⟩ (no default, initially empty)

Sets an internal link of the marker to a document target with the given ⟨*name*⟩. The package **hyperref** is required for this option. For an own marker type, it is applied as Ti*k*Z option **mrchyperpath**.

N 2020-05-04 **mermap/marker/use urls**=true|false (default **true**, initially **true**)
N 2020-05-04 **mermap/marker/ignore urls**=true|false (default **true**, initially **false**)

If mermap/marker/use urls is set to **false**, external links with mermap/marker/url are ignored.
mermap/marker/ignore urls is inverse to mermap/marker/use urls.

N 2020-05-04 **mermap/marker/use links**=true|false (default **true**, initially **true**)
N 2020-05-04 **mermap/marker/ignore links**=true|false (default **true**, initially **false**)

If mermap/marker/use links is set to **false**, internal links with mermap/marker/link are ignored.
mermap/marker/ignore links is inverse to mermap/marker/use links.

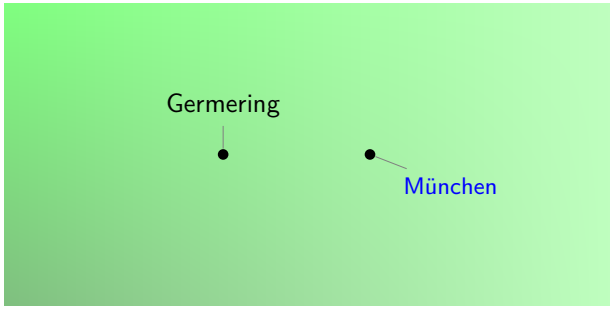
7.2 Marker Types

`mermap/marker/type`=⟨*type*⟩ (no default, initially `classic`)

Decides about the basic shape and style of the marker. Feasible values for ⟨*type*⟩ are listed in the following. More values can be defined by `\mrcnewmarkertype` → P. 69.

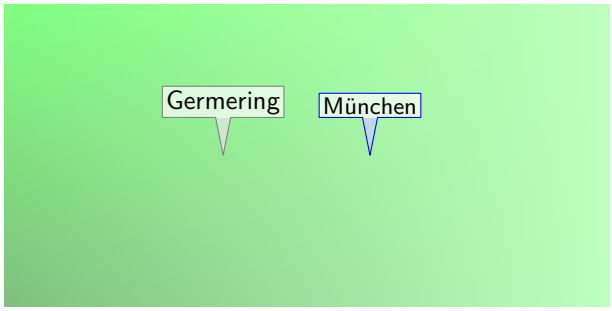
- `classic`:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=classic}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=classic, text=blue, font=\sffamily\footnotesize}
\mrcmarker{position=48.14:11.57, contents={M\"unchen}, angle=-30}
\end{tikzpicture}
```



- `pin`:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=pin}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=pin, draw=blue, fill=blue!20!white,
  font=\sffamily\footnotesize}
\mrcmarker{position=48.14:11.57, contents={M\"unchen}, alias={munich}}
\end{tikzpicture}
```

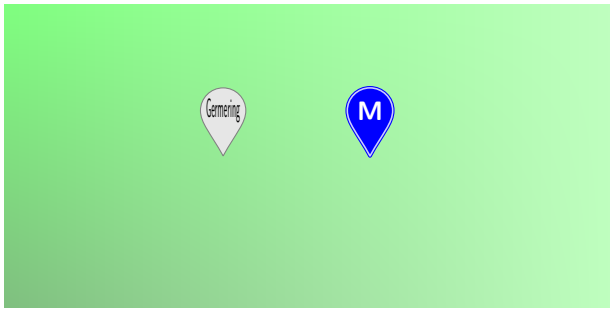


- pinflip:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=pinflip}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=pinflip, draw=red, fill=red!20!white,
  font=\sffamily\footnotesize}
\mrcmarker{position=48.14:11.57, contents={M\"unchen}, shift=4mm}
\end{tikzpicture}
```

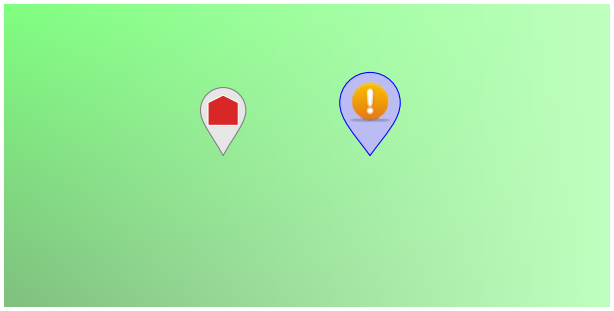
- drop:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=drop}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=drop, fill=blue, draw=blue!20!white, text=white,
  font=\sffamily\small\bfseries,
  path style={line join=round, thin, draw=mrcmarkerfill,
    double=mrcmarkerdraw,double distance=0.6pt}  }
\mrcmarker{position=48.14:11.57, contents={M}, alias={munich}}
\end{tikzpicture}
```



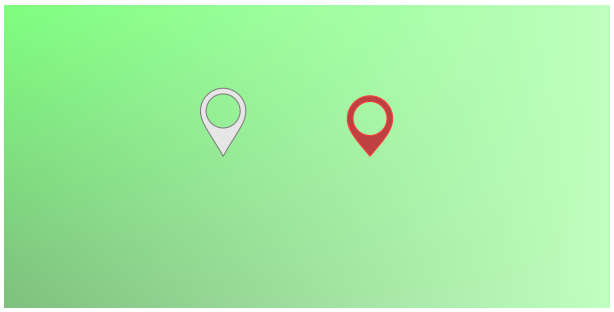
- pictodrop:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=pictodrop, pictocontents={
  \draw[fill=red!70!gray,draw=white]
    (-0.2,-0.2)--(0.2,-0.2)--(0.2,0.1)--(0,0.2)--(-0.2,0.1) -- cycle;
}}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=pictodrop,
  fill=blue!75!gray!30, draw=blue,
  radius=4mm, shift=-1mm}
\mrcmarker{position=48.14:11.57,
  pictocontents={
  \node {\includegraphics[width=6mm]{alertmessage-warning.png}};
}}
\end{tikzpicture}
```



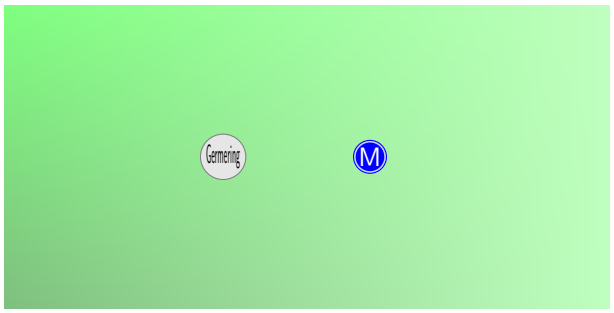
- pictodropring:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=pictodropring}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=pictodropring,
  fill=red!50!gray, draw=red!75!white,
  shift=-1mm}
\mrcmarker{position=48.14:11.57}
\end{tikzpicture}
```



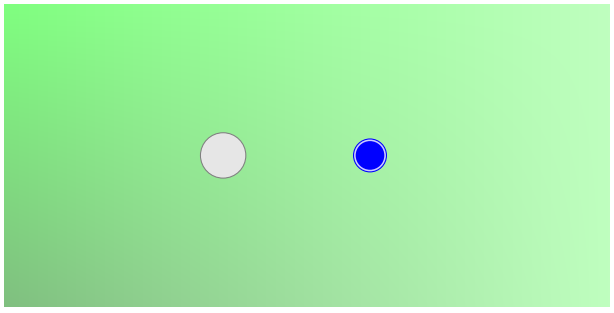
- knob:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=knob}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=knob, radius=2mm,
  fill=blue, draw=blue!20!white,
  text=white, path style={draw=mrcmarkerfill, double=mrcmarkerdraw,
    double distance=0.6pt}  }
\mrcmarker{position=48.14:11.57, contents={M}}
\end{tikzpicture}
```



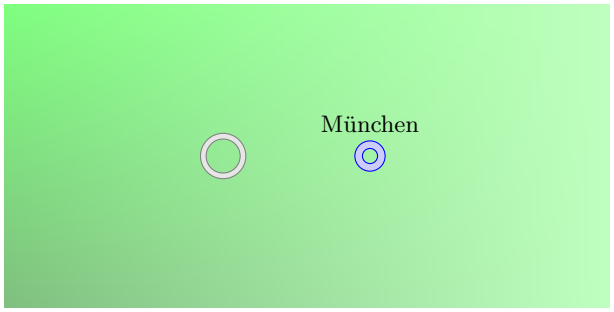
- pictoknob:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=pictoknob}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=pictoknob, radius=2mm,
  fill=blue, draw=blue!20!white,
  path style={draw=mrcmarkerfill, double=mrcmarkerdraw,
    double distance=0.6pt}  }
\mrcmarker{position=48.14:11.57, alias={munich}}
\end{tikzpicture}
```



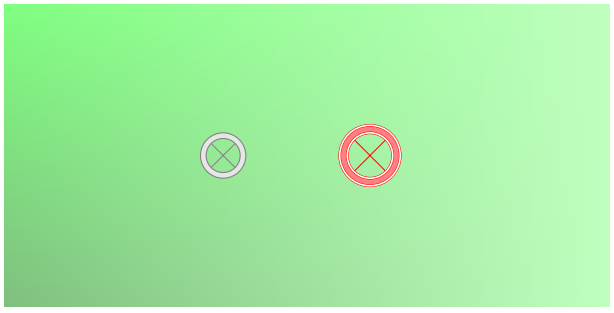
- pictoknobring:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=pictoknobring}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=pictoknobring, radius=2mm, inner radius=1mm,
  draw=blue, fill=blue!20!white,
  pictocontents={\node[above] at (0,\mrcmarkerradius)
    {\mrcmarkercontents};}
  }
\mrcmarker{position=48.14:11.57, contents={M\"unchen}}
\end{tikzpicture}
```



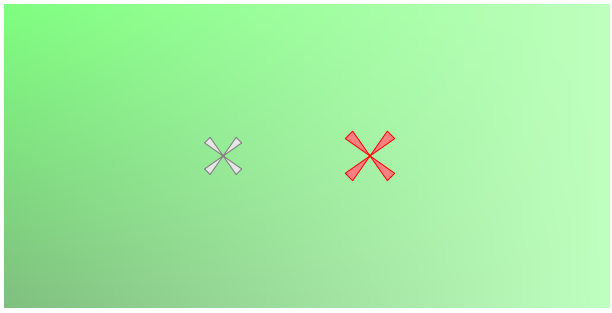
- ringx:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=ringx}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=ringx, radius=4mm, inner radius=3mm,
  draw=red, fill=red!50!white,
  path style={draw=mrcmarkerdraw, very thin, double=white,
    double distance=0.6pt}
    }
\mrcmarker{position=48.14:11.57, alias={munich}}
\end{tikzpicture}
```



- markx:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mermapsetmarker{type=markx}
\mrcmarker{position=48.14:11.36, contents={Germering}}
\mermapsetmarker{type=markx, radius=4mm, inner radius=3mm,
  draw=red, fill=red!50!white, path style={very thin}
    }
\mrcmarker{position=48.14:11.57, alias={munich}}
\end{tikzpicture}
```



The different marker types can be customized by some additional options:

mermap/marker/font=⟨*font*⟩ (no default, initially `\sffamily\small`)

Font of the marker text (**classic**, **pin**, **pinflip**, **drop**, **knob**). It is accessible as **\mrcmarkerfont** (use read-only).

mermap/marker/text=⟨*color*⟩ (no default, initially `black`)

Color of the marker text (**classic**, **pin**, **pinflip**, **drop**, **knob**). It is accessible as **mrcmarkertext** (use read-only).

mermap/marker/draw=⟨*color*⟩ (no default, initially `gray`)

Color of the marker frame (**pin**, **pinflip**, **drop**, **pictodrop**, **pictodropring**, **knob**, **pictoknob**, **pictoknobring**, **ringx**, **markx**). It is accessible as **mrcmarkerdraw** (use read-only).

mermap/marker/fill=⟨*color*⟩ (no default, initially `gray!20`)

Color of the marker interior (**pin**, **pinflip**, **drop**, **pictodrop**, **pictodropring**, **knob**, **pictoknob**, **pictoknobring**, **ringx**, **markx**). It is accessible as **mrcmarkerfill** (use read-only).

mermap/marker/angle=⟨*angle*⟩ (no default, initially `90`)

Angle of the marker (**classic**). It is accessible as **\mrcmarkerangle** (use read-only).

mermap/marker/shift=⟨*length*⟩ (no default, initially `0pt`)

Shift of the marker text (**pin**, **pinflip**). It also shifts the drop center (**drop**, **pictodrop**, **pictodropring**). It is accessible as **\mrcmarkershift** (use read-only).

N 2020-08-06 **mermap/marker/distance**=⟨*length*⟩ (no default, initially `5mm`)

Length of the marker needle (**pin**, **pinflip**). It is accessible as **\mrcmarkerdistance** (use read-only).

`mermap/marker/radius`=⟨*length*⟩ (no default, initially `3mm`)

Radius of the marker (`classic`, `drop`, `pictodrop`, `pictodropring`, `knob`, `pictoknob`, `pictoknobring`, `ringx`, `markx`). It is accessible as `\mrcmarkerradius` (use read-only).

`mermap/marker/inner radius`=⟨*length*⟩ (no default, initially `2.25mm`)

Inner radius of the marker (`pictodropring`, `pictoknobring`, `ringx`). It is accessible as `\mrcmarkerinnerradius` (use read-only).

`mermap/marker/path style`={⟨*options*⟩} (no default, initially empty)

Ti*k*Z ⟨*options*⟩ which are added to (some) path elements of of the marker (`pin`, `pinflip`, `drop`, `pictodrop`, `pictodropring`, `knob`, `pictoknob`, `pictoknobring`, `ringx`, `markx`). It is accessible as Ti*k*Z option `mrcpathstyle` (use read-only).

`mermap/marker/node style`={⟨*options*⟩} (no default, initially empty)

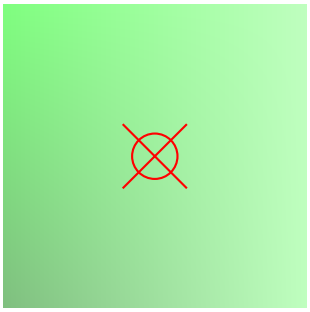
Ti*k*Z ⟨*options*⟩ which are added to the node element of of the marker (`classic`, `pin`, `pinflip`, `drop`, `knob`). It is accessible as Ti*k*Z option `mrcnodestyle` (use read-only).

7.3 New Marker Types

`\mrcnewmarkertype`{⟨*type name*⟩}{⟨Ti*k*Z *code*⟩}

Creates a new `mermap/marker/type` → P. 63 value ⟨*type name*⟩ using the given ⟨Ti*k*Z *code*⟩ for drawing a place marker. To avoid future name clashes, you should start a private ⟨*type name*⟩ with letter `x`. For ⟨Ti*k*Z *code*⟩ settings like `\mrcmarkercontents`, `\mrcmarkerfont`, or `mrcmarkerfill` may be used or ignored.

```
\mrcnewmarkertype{xmark}{
  \path[draw=mrcmarkerdraw,thick] circle (\mrcmarkerradius);
  \path[draw=mrcmarkerdraw,thick]
     (45:2*\mrcmarkerradius)--(225:2*\mrcmarkerradius)
     (135:2*\mrcmarkerradius)--(315:2*\mrcmarkerradius);
}
\begin{tikzpicture}
\mrcmap[tex width=4cm, tex height=4cm,
  latitude=48.14, longitude=11.57, target=none]{}
\mrcdrawmap[draw=path]
\mrcmarker{type=xmark,
  position=48.14:11.57,alias={munich},
  draw=red}
\end{tikzpicture}
```

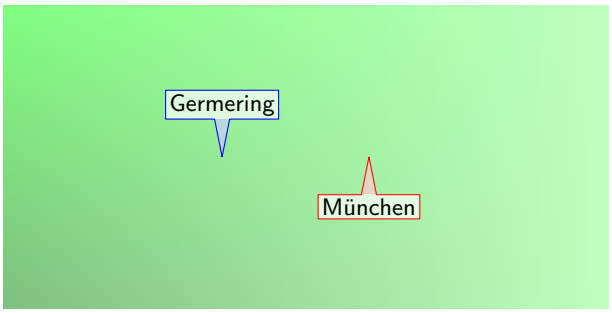


7.4 New Marker Styles

`\mrcnewmarkerstyle{`⟨*style name*⟩`}{`⟨*options*⟩`}`

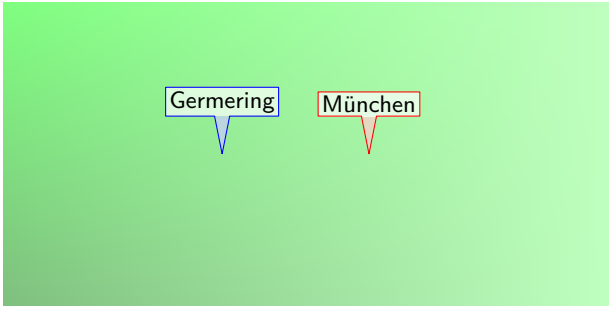
Creates a new `mermap/marker/style` value ⟨*style name*⟩ using the given ⟨*options*⟩ for drawing a place marker. All ⟨*options*⟩ share the common prefix `mermap/marker/`. Here, a *style* has the same concept as a Ti*k*Z style.

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mrcnewmarkerstyle{one}{type=pin, draw=blue, fill=blue!20!white,
  font=\sffamily\footnotesize}
\mrcnewmarkerstyle{two}{type=pinflip, draw=red, fill=red!20!white,
  font=\sffamily\footnotesize}
\mrcmarker{style=one, position=48.14:11.36, contents={Germering}}
\mrcmarker{style=two, position=48.14:11.57, contents={M\"unchen}}
\end{tikzpicture}
```



A *style* can have one parameter, but note the small difference in applying this parameter compared to Ti*k*Z:

```
\begin{tikzpicture}
\mrcmap[tex width=8cm, tex height=4cm, zoom=10,
  latitude=48.14, longitude=11.48, target=none]{}
\mrcdrawmap[draw=path]
\mrcnewmarkerstyle{color}{type=pin, draw=#1, fill=#1!20!white,
  font=\sffamily\footnotesize}
\mrcmarker{style/color=blue, position=48.14:11.36, contents={Germering}}
\mrcmarker{style/color=red, position=48.14:11.57, contents={M\"unchen}}
\end{tikzpicture}
```



`mermap/marker/style=`⟨*style name*⟩ (no default, initially unset)

Applies a given ⟨*style name*⟩, i.e. all options which were stored by `\mrcnewmarkerstyle` under this name.

8 Routes

Routes are Ti*k*Z `path` objects which can be drawn with the appropriate Ti*k*Z macros and the coordinate system documented in Section 3 on page 17. Nevertheless, in the following some alternatives are described which allow to specify a path by a sequence of points (`\mrcpoint`). The idea is that such a sequence of points is exported by a third-party application for inclusion in a LaTeX map, see Section 8.2 on page 73.

8.1 Route Path Definition

`\begin{mrcroute}[`⟨*options*⟩`]`
⟨*environment content*⟩
`\end{mrcroute}`

This environment creates a Ti*k*Z `path` with given Ti*k*Z ⟨*options*⟩. The ⟨*environment content*⟩ is a sequence of points made by `\mrcpoint`.

```
\begin{mrcroute}[red, very thick]
   \mrcpoint{48.137222}{11.575556}
   \mrcpoint{49.019479}{12.0976942}
   ...
\end{mrcroute}
```

Note that a map definition by `\mrcdefinemap` → P. 18, `\mrcmap` → P. 35, or `\mrcapplymap` → P. 35 is needed before a route `path` can be drawn.

`\begin{mrcroute*}[`⟨*options*⟩`]`
⟨*environment content*⟩
`\end{mrcroute*}`

Identical to `mrcroute`, but the created Ti*k*Z `path` is closed.

`\mrcrouteinput[`⟨*options*⟩`]{`⟨*filename*⟩`}`

Identical to `mrcroute`, but the sequence of points is included from a file with the given ⟨*filename*⟩.

`\mrcrouteinput*[`⟨*options*⟩`]{`⟨*filename*⟩`}`

Identical to `\mrcrouteinput`, but the created Ti*k*Z `path` is closed.

`\mrcpoint{`⟨*latitude*⟩`}{`⟨*longitude*⟩`}`

Specifies a single coordinate point with given ⟨*latitude*⟩ and ⟨*longitude*⟩ as part of sequence inside `mrcroute`.

This is a wrapper for `\pgfpathmoveto` respectively `\pgfpathlineto`.

`mermap/every route=`⟨*options*⟩ (no default, initially empty)

Sets Ti*k*Z ⟨*options*⟩ which are applied to every `mrcroute` and `\mrcrouteinput`.

```
% \mrcsetapikey{thunderforest}{YOUR-API-KEY}  % registered key
\begin{tikzpicture}
  \mrcmap[type=areafit,
    west=5,east=15,south=47,north=55,
    source=thunderforest neighbourhood,
    tex width=14cm, tex height=14cm,
    flex area fit=5mm
    ]{routes_example}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
    {\mrcmapattribution};
  \mrcclipmap
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
  \begin{mrcroute}[blue,line width=0.4mm,line cap=round,
        line join=round,double=blue!5!white,double distance=0.4mm]
     \mrcpoint{48.137222}{11.575556}
     \mrcpoint{49.019479}{12.0976942}
     \mrcpoint{49.45522}{11.07631}
     \mrcpoint{50.978056}{11.029167}
     \mrcpoint{52.518611}{13.408333}
  \end{mrcroute}
\end{tikzpicture}
```

Maps © Thunderforest, Data © OpenStreetMap contributors

8.2 Example Python Conversion Scripts for gpx

The following scripts are examples for conversions from a standard routes and tracks of gpx files to LaTeX include files (assuming a single route/track per file).

Python 3 script to convert `route.gpx` to a sequence of points file `route.inc`

```
import xmltodict

with open('route.gpx', encoding='utf-8') as gpx:
    doc = xmltodict.parse(gpx.read())
    with open('route.inc', "w", encoding="utf-8") as inc:
        for rtept in doc['gpx']['rte']['rtept']:
            lat = rtept['@lat']
            lon = rtept['@lon']
            inc.write(f'\\mrcpoint{{{lat}}}{{{lon}}}\n')
```

Python 3 script to convert `track.gpx` to a sequence of points file `track.inc`

```
import xmltodict

with open('track.gpx', encoding='utf-8') as gpx:
    doc = xmltodict.parse(gpx.read())
    with open('track.inc', "w", encoding="utf-8") as inc:
        for trkpt in doc['gpx']['trk']['trkseg']['trkpt']:
            lat = trkpt['@lat']
            lon = trkpt['@lon']
            inc.write(f'\\mrcpoint{{{lat}}}{{{lon}}}\n')
```

9 Orthodromes and Loxodromes

A loxodrome is a curve which crosses all meridians with a constant angle. On Mercator maps, loxodromes are depicted as straight lines and can be drawn by simple Ti*k*Z `path` elements.

On a sphere, the shortest path from one point to another runs along an orthodrome where an orthodrome is a great-circle.

The mathematical background and further information are found in [1].

9.1 Orthodrome Drawing

`mermap/samples`=⟨*number*⟩ (no default, initially `100`)

An orthodrome curve is approximated by a polygon trajectory with ⟨*number*⟩ pieces.

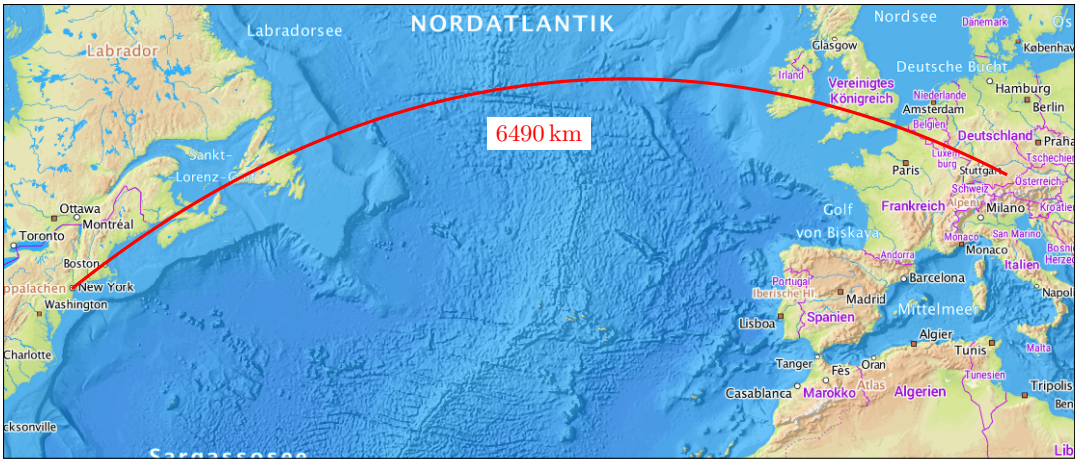
`/tikz/mermap samples`=⟨*number*⟩ (style, no default)

Ti*k*Z variant to set `mermap/samples`.

`\mrcdraworthodrome`[⟨*options*⟩]{⟨*lat1*⟩}{⟨*lon1*⟩}{⟨*lat2*⟩}{⟨*lon2*⟩}

Draws an orthodrome curve from a point with latitude ⟨*lat1*⟩ and longitude ⟨*lon1*⟩ to a point with latitude ⟨*lat2*⟩ and longitude ⟨*lon2*⟩. This is a Ti*k*Z path object where ⟨*options*⟩ are Ti*k*Z settings for this path. There are two orthodrome pieces connecting two positions (forming a great-circle). `\mrcdraworthodrome` does not necessarily choose the shorter one, see [1]. The drawn orthodrome is a spherical approximation instead of an ellipsoidal one.

```
\begin{tikzpicture}
  \mrcmap[type=areafit,
      south=40.7,north=48.2,west=-74.1,east=11.6,
      source=topplusopen web,
      tex width=\linewidth,tex height=6cm,
    ]{ortho_orthodrome1}
    \mrcdrawmap
    \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
      {\mrcmapattribution};
    \mrcclipmap
  \draw (mrcmap.south west) rectangle (mrcmap.north east);
  \mrcdraworthodrome[red,very thick,mermap samples=100]
     {48.14}{11.58}{40.71}{-74.01}
  \node[red,fill=white] at ([above=1.3cm]mrcmap) {
    \mrcprettyorthodistance{48.14}{11.58}{40.71}{-74.01} };
\end{tikzpicture}
```

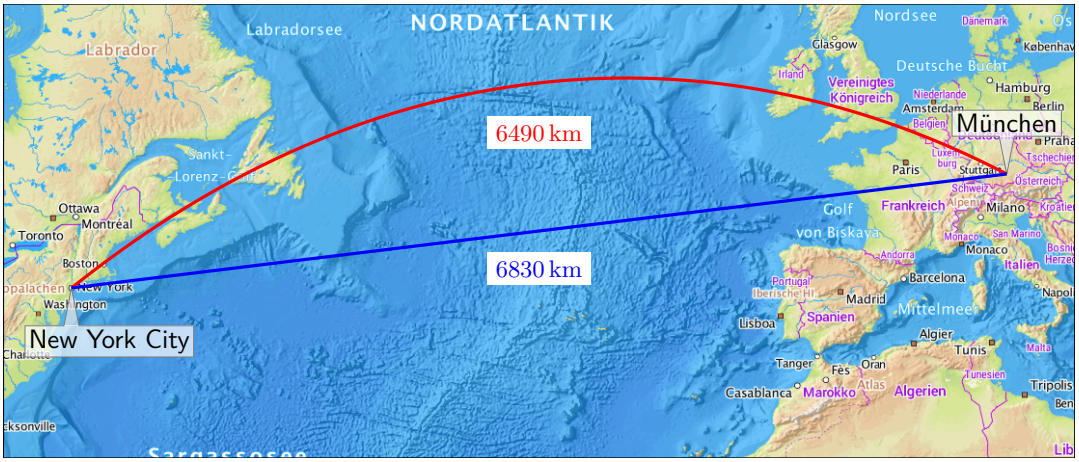


Kartendarstellung: © BKG (2024), Datenquellen

`\mrcNPdraworthodrome[⟨options⟩]{⟨name1⟩}{⟨name2⟩}`

Identical to `\mrcdraworthodrome` → P. 74, but the start and end point are described by named positions ⟨*name1*⟩ and ⟨*name2*⟩.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.14}{11.58}
  \mrcNPdef{newyork}{40.71}{-74.01}
  \mrcmap[type=areafit, area={munich,newyork},
      source=topplusopen web,
      tex width=\linewidth,tex height=6cm,
      ]{ortho_orthodrome2}
    \mrcdrawmap
    \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
      {\mrcmapattribution};
    \mrcclipmap
  \draw (mrcmap.south west) rectangle (mrcmap.north east);
  \mrcmarker{type=pin,named position=munich,contents={M\"unchen}}
  \mrcmarker{type=pinflip,shift=5mm,named position=newyork,
    contents={New York City}}
  \mrcNPdraworthodrome[red,very thick] {munich}{newyork}
  \draw[blue,very thick] (\mrcNPcs{munich}) -- (\mrcNPcs{newyork});
  \node[red,fill=white] at ([above=1.3cm]mrcmap) {
    \mrcNPprettyorthodistance{munich}{newyork} };
  \node[blue,fill=white] at ([below=5mm]mrcmap) {
    \mrcNPprettyloxodistance{munich}{newyork} };
\end{tikzpicture}
```



Kartendarstellung: © BKG (2024), Datenquellen

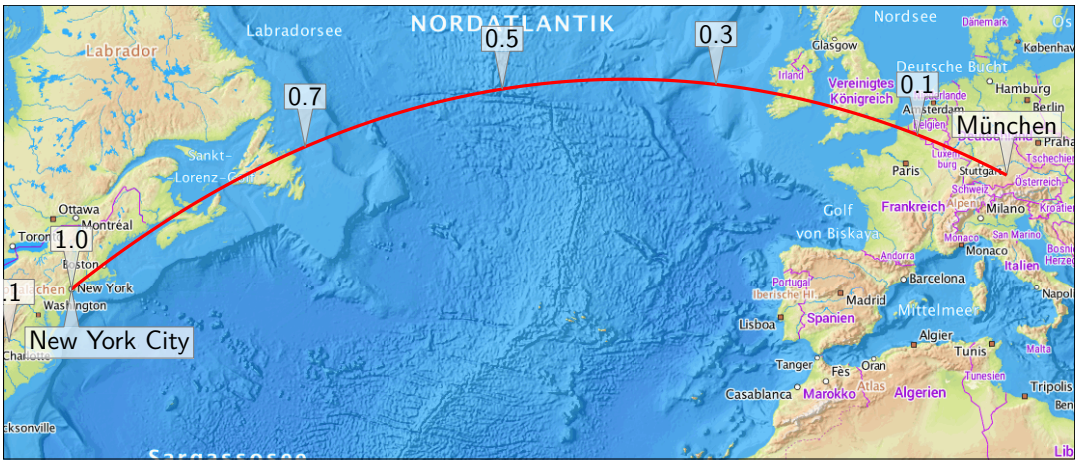
9.2 Orthodrome Points

N 2024-08-01 `\mrcNPfromOrthoFraction{`⟨*name*⟩`}{`⟨*lat1*⟩`}{`⟨*lon1*⟩`}{`⟨*lat2*⟩`}{`⟨*lon2*⟩`}{`⟨*fraction*⟩`}`

N 2024-08-01 `\mrcNPfromOrthoFractionNamed{`⟨*name*⟩`}{`⟨*name1*⟩`}{`⟨*name2*⟩`}{`⟨*fraction*⟩`}`

Defines a new named position ⟨*name*⟩ which is located on a ⟨*fraction*⟩ from the starting point of the orthodromic curve from a point with latitude ⟨*lat1*⟩ and longitude ⟨*lon1*⟩ (or described by a named position ⟨*name1*⟩) to a point with latitude ⟨*lat2*⟩ and longitude ⟨*lon2*⟩ (or described by a named position ⟨*name2*⟩). The given starting and final point of the orthodrome need to have a distance of more than 1 m.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.14}{11.58}
  \mrcNPdef{newyork}{40.71}{-74.01}
  \mrcmap[type=areafit, area={munich,newyork},
      source=topplusopen web,
      tex width=\linewidth,tex height=6cm,
      ]{ortho_orthodrome3}
    \mrcdrawmap
    \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
      {\mrcmapattribution};
    \mrcclipmap
  \draw (mrcmap.south west) rectangle (mrcmap.north east);
  \mrcmarker{type=pin,named position=munich,contents={M\"unchen}}
  \mrcmarker{type=pinflip,shift=5mm,named position=newyork,
    contents={New York City}}
  \mrcNPdraworthodrome[red,very thick] {munich}{newyork}
  \foreach \fraction in { 0.1,0.3,0.5,0.7,1.0,1.1 }
    {
      \mrcNPfromOrthoFractionNamed{waypoint}{munich}{newyork}{\fraction}
      \mrcmarker{type=pin,named position=waypoint,contents={\fraction}}
    }
\end{tikzpicture}
```



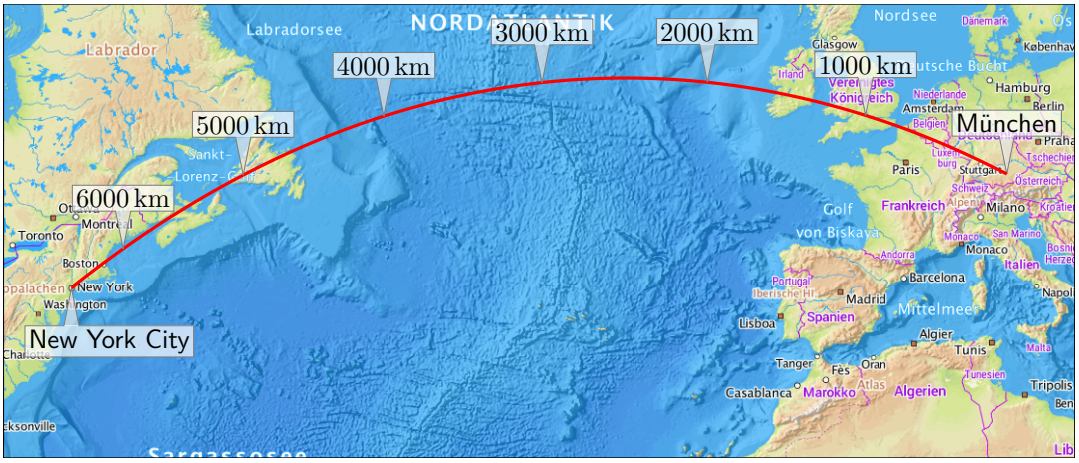
Kartendarstellung: © BKG (2024), Datenquellen

N 2024-08-01 `\mrcNPfromOrthoDistance{⟨name⟩}{⟨lat1⟩}{⟨lon1⟩}{⟨lat2⟩}{⟨lon2⟩}{⟨distance⟩}`

N 2024-08-01 `\mrcNPfromOrthoDistanceNamed{⟨name⟩}{⟨name1⟩}{⟨name2⟩}{⟨distance⟩}`

Defines a new named position ⟨*name*⟩ which is located on a ⟨*distance*⟩ from the starting point of the orthodromic curve from a point with latitude ⟨*lat1*⟩ and longitude ⟨*lon1*⟩ (or described by a named position ⟨*name1*⟩) to a point with latitude ⟨*lat2*⟩ and longitude ⟨*lon2*⟩ (or described by a named position ⟨*name2*⟩). The given starting and final point of the orthodrome need to have a distance of more than 1 m. The unit for the ⟨*distance*⟩ is kilometer.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.14}{11.58}
  \mrcNPdef{newyork}{40.71}{-74.01}
  \mrcmap[type=areafit, area={munich,newyork},
      source=topplusopen web,
      tex width=\linewidth,tex height=6cm,
      ]{ortho_orthodrome4}
    \mrcdrawmap
    \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
      {\mrcmapattribution};
    \mrcclipmap
  \draw (mrcmap.south west) rectangle (mrcmap.north east);
  \mrcmarker{type=pin,named position=munich,contents={M\"unchen}}
  \mrcmarker{type=pinflip,shift=5mm,named position=newyork,
    contents={New York City}}
  \mrcNPdraworthodrome[red,very thick] {munich}{newyork}
  \foreach \distance in { 1000,2000,3000,4000,5000,6000 }
    {
      \mrcNPfromOrthoDistanceNamed{waypoint}{munich}{newyork}{\distance}
      \mrcmarker{type=pin,named position=waypoint,
          contents={\SI{\distance}{\kilo\meter}}}
    }
\end{tikzpicture}
```



Kartendarstellung: © BKG (2024), Datenquellen

9.3 Orthodromic and Loxodromic Distances

`\mrcprettyorthodistance{`⟨*lat1*⟩`}{`⟨*lon1*⟩`}{`⟨*lat2*⟩`}{`⟨*lon2*⟩`}`

Approximate orthodromic distance between two points with latitude ⟨*lat1*⟩, longitude ⟨*lon1*⟩ and latitude ⟨*lat2*⟩, longitude ⟨*lon2*⟩ with three valid digits.

```
\mrcprettyorthodistance{48.14}{11.58}{40.71}{-74.01}
```

6490 km

`\mrcNPprettyorthodistance{`⟨*name1*⟩`}{`⟨*name2*⟩`}`

Approximate orthodromic distance between two named positions ⟨*name1*⟩ and ⟨*name2*⟩ with three valid digits.

```
\mrcNPdef{munich}{48.14}{11.58}
\mrcNPdef{newyork}{40.71}{-74.01}
\mrcNPprettyorthodistance{munich}{newyork}
```

6490 km

`\mrcstoreorthodistance{`⟨*macro*⟩`}{`⟨*lat1*⟩`}{`⟨*lon1*⟩`}{`⟨*lat2*⟩`}{`⟨*lon2*⟩`}`

Stores the approximate orthodromic distance (in kilometers) between two points with latitude ⟨*lat1*⟩, longitude ⟨*lon1*⟩ and latitude ⟨*lat2*⟩, longitude ⟨*lon2*⟩ to a given ⟨*macro*⟩.

```
\mrcstoreorthodistance\mydist{48.14}{11.58}{40.71}{-74.01}
\mydist
```

6488.72407474055

`\mrcprettyloxodistance{`⟨*lat1*⟩`}{`⟨*lon1*⟩`}{`⟨*lat2*⟩`}{`⟨*lon2*⟩`}`

Approximate loxodromic distance between two points with latitude ⟨*lat1*⟩, longitude ⟨*lon1*⟩ and latitude ⟨*lat2*⟩, longitude ⟨*lon2*⟩ with three valid digits.

```
\mrcprettyloxodistance{48.14}{11.58}{40.71}{-74.01}
```

6830 km

`\mrcNPprettyloxodistance{`⟨*name1*⟩`}{`⟨*name2*⟩`}`

Approximate loxodromic distance between two named positions ⟨*name1*⟩ and ⟨*name2*⟩ with three valid digits.

```
\mrcNPdef{munich}{48.14}{11.58}
\mrcNPdef{newyork}{40.71}{-74.01}
\mrcNPprettyloxodistance{munich}{newyork}
```

6830 km

`\mrcstoreloxodistance{`⟨*macro*⟩`}{`⟨*lat1*⟩`}{`⟨*lon1*⟩`}{`⟨*lat2*⟩`}{`⟨*lon2*⟩`}`

Stores the approximate loxodromic distance (in kilometers) between two points with latitude ⟨*lat1*⟩, longitude ⟨*lon1*⟩ and latitude ⟨*lat2*⟩, longitude ⟨*lon2*⟩ to a given ⟨*macro*⟩.

```
\mrcstoreloxodistance\mydist{48.14}{11.58}{40.71}{-74.01}
\mydist
```

6833.046494937649

10 Animations

An animation in the context of map drawing is considered to be a direct movement from a starting position to a final position with possible adaption of scale denominators.

This package does not provide animation production, but allows to create a PDF with a sequence of maps (frames) following such an animation path.

10.1 Animation Environment

N 2024-07-31

```
\begin{mrcAnimation}{⟨options⟩}
  ⟨environment content⟩
\end{mrcAnimation}
```

According to the given ⟨*options*⟩, the environment loops over the ⟨*environment content*⟩ several times generating *frames* for an external animation program. A detailed description for the ⟨*options*⟩ is found in Section 10.2 on page 82.

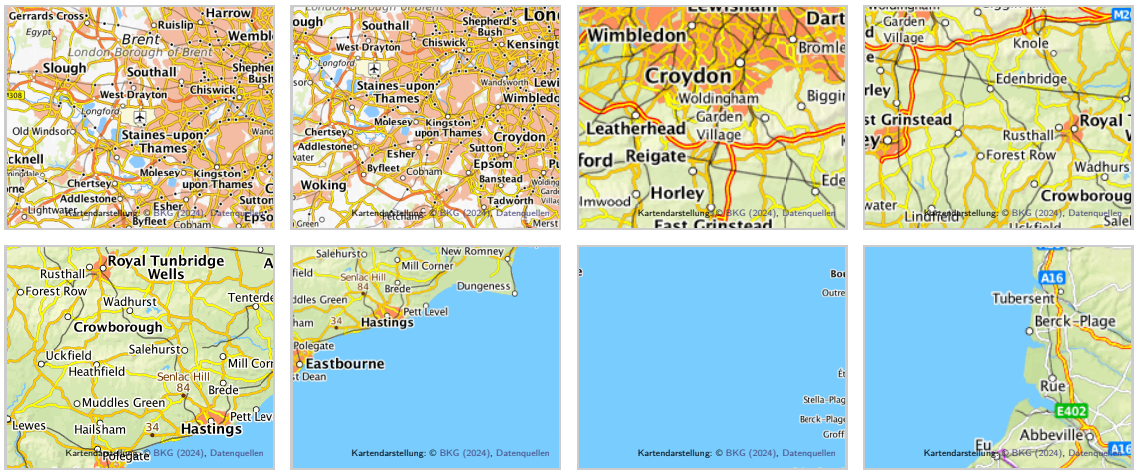
- Basically, the position is moved from a given `mermap/anim/start-position` → P. 82 with time index `0` following an orthodrome to a `mermap/anim/final-position` → P. 82 with time index `1`.
- This time interval $[0, 1]$ is divided into `mermap/anim/frames` → P. 82. The current frame number is denoted by `\mrcAnimFrame` → P. 84, the current time is denoted by `\mrcAnimTime` → P. 84, and the current position is denoted by `\mrcAnimLatitude` → P. 84 and `\mrcAnimLongitude` → P. 84, alternatively by the named positon **`AnimNP`**.
- During movement, a time dependent sequence of `mermap/anim/scale-denominators` → P. 82 defines the current scale denominator `\mrcAnimScaleDenom` → P. 84.
- The frames may have an equal time distance or have different time distances using a `mermap/anim/timewarp` → P. 83. This could be used to slow down movement near ground and to speed up with more height.

The generated sequence of frames is applicable for

- Movement along an orthodrome from starting to final position with an unchanged common scale denominator.
- Zoom in or out on a fixed position.
- Combination of both animation.

For more complex animations, several `mrcAnimation` environments may be used consecutively.

```
\mrcNPdef{heathrow}{51.4678}{-0.4548}
\mrcNPdef{fiumicino}{41.8151}{12.2508}

\begin{tcbraster}[raster height=22cm,raster columns=4,raster rows=7,
    tile,size=minimal,boxsep=1pt,colback=black!20 ]
\begin{mrcAnimation}
  {
    named-start-position = heathrow,
    named-final-position = fiumicino,
    frames = 36,
    scale-denominators = 0/1000000 - 0.3/8000000 - 0.7/8000000 - 1/1000000,
    timewarp-slow-start-final = 1.5,
  }
  \begin{tcolorbox}
    \begin{tikzpicture}
    \mermapset
      {
        named flex scale = \mrcAnimScaleDenom:AnimNP,
      }
    \mrcmap
      [
        type           = reference,
        named position = AnimNP,
        source         = topplusopen web,
        tex width      = \tcbtextwidth,
        tex height     = \tcbtextheight
      ]{london-roma-\mrcAnimFrame}
    \mrcdrawmap
    \node[above left,font=\fontsize{3.5pt}{3.5pt}\sffamily]
      at (mrcmap.south east) {\mrcmapattribution};
    \mrcclipmap
    \end{tikzpicture}
  \end{tcolorbox}
\end{mrcAnimation}
\end{tcbraster}
```




Berck-Plage
Rue
Abbeville
Flixecourt
Saint-Vaast-en-Chaussée
Villers-Bretonneux
Neufchâtel-en
Doullens
Amiens
Montdidier
Saint-Just-en-Chaussée
Beauvais
Laon
Compiègne
Meaux
Reims
Paris
Charleville-Mézières
Châlons-en-Champagne
Verdun-sur-Meuse
Saint-Dizier
Troyes
Pariser Becken
Chaumont
Nancy
Épinal
Dijon
Besançon
Colmar
Mulhouse
Belfort
Basel
Bern
SCHWEIZ
Jura
Genfersee
Genève
Thonon-les-Bains
Annecy
Chambéry
Mont Blanc
Aosta
Biella
Novara
Torino
Milano
Bergamo
Pavia
Piacenza
Alessandria
Cremona
Parma
Genova
La Spezia
Massa
Lucca
Pisa
Livorno
Ligurisches Meer
Pontedera
Cecina
Empoli
San Vincenzo
Follonica
Grosseto
Scansano
Manciano
Capalbio
Canino
Tuscania
Montebello
Tarquinia
Monte Maggiore
Civitavecchia
Vetralla
Blera
Manziana
Bracciano
Santa Marinella
Ladispoli
Fregene
Fiumicino-Isola Sacra
Lido di Ostia
Kartendarstellung: © BKG (2024), Datenquellen

10.2 Animation Options

N 2023-07-31 **mermap/anim/start-position**=⟨*latitude*⟩/⟨*longitude*⟩ (initially `51.4779/0`)
N 2023-07-31 **mermap/anim/named-start-position**=⟨*name*⟩

Starting position of the animation given by ⟨*latitude*⟩ and ⟨*longitude*⟩ or by a named position using ⟨*name*⟩.

N 2023-07-31 **mermap/anim/final-position**=⟨*latitude*⟩/⟨*longitude*⟩ (initially `51.4779/0`)
N 2023-07-31 **mermap/anim/named-final-position**=⟨*name*⟩

Final position of the animation given by ⟨*latitude*⟩ and ⟨*longitude*⟩ or by a named position using ⟨*name*⟩.

N 2023-07-31 **mermap/anim/position**=⟨*latitude*⟩/⟨*longitude*⟩ (initially `51.4779/0`)
N 2023-07-31 **mermap/anim/named-position**=⟨*name*⟩

Fixed position of the animation given by ⟨*latitude*⟩ and ⟨*longitude*⟩ or by a named position using ⟨*name*⟩. This sets `mermap/anim/start-position` and `mermap/anim/final-position` to the identical value.

N 2023-07-31 **mermap/anim/frames**=⟨*frame number*⟩ (initially `20`)

Integer ⟨*frame number*⟩ for the animation, at least `2`. For maps made with tiles, even a large ⟨*frame number*⟩ typically results in limited downloads, because the tiles a reused as far as possible. For WMS maps, every frame could give another download!

N 2023-07-31 **mermap/anim/drop-first-frame**=`true|false` (default `true`, initially `false`)
N 2023-07-31 **mermap/anim/drop-last-frame**=`true|false` (default `true`, initially `false`)

If several `mrcAnimation` → P. 79 environments are used consecutively, the end frame of one animation is identical to the start frame of the next animation. These options allow to remove one superfluous connecting frame.

N 2023-07-31 **mermap/anim/drop-no-frame** (no value, initially set)

Reset to drop no frames.

N 2023-07-31 **mermap/anim/scale-denominators**=`{`⟨*time and scale sequence*⟩`}` (initially `0/25000-1/25000`)

The ⟨*time and scale sequence*⟩ has to obey the following pattern:
⟨$time_1$⟩/⟨$scale_1$⟩ `-` ⟨$time_2$⟩/⟨$scale_2$⟩ `-` ... `-` ⟨$time_n$⟩/⟨$scale_n$⟩
The time values have to be taken from the interval $[0, 1]$ and have to be strictly monotonically increasing, i.e.
$0 \leq \langle time_1 \rangle < \langle time_2 \rangle < \ldots < \langle time_n \rangle \leq 1$
If not given, time and scale for time 0 and 1 are automatically added as constant continuation.
Depending on the current time value `\mrcAnimTime` → P. 84, the current scale denominator `\mrcAnimScaleDenom` → P. 84 is interpolated from this sequence using a logarithmic approach.

N 2023-07-31 **mermap/anim/common-scale-denominator**=`{`⟨*scale denominator*⟩`}` (initially `25000`)

This is a shortcut for
`mermap/anim/scale-denominators =`
`0/{`⟨*scale denominator*⟩`} - 1/{`⟨*scale denominator*⟩`}`
Thereby, a fixed `{`⟨*scale denominator*⟩`}` is set for the animation.

N 2023-07-31 **mermap/anim/timewarp=**⟨*macro*⟩ (initially `\mrcTimewarpIdentity`)

⟨*macro*⟩ has to be a fully expandable LaTeX macro with one parameter. The expansion has to be a valid `expl3` ⟨*floating point expression*⟩ denoting a strictly monotonically increasing function mapping from the interval $[0, 1]$ into $[0, 1]$ again. This timewarp may slow down and accelerate time over the span from 0 to 1.

N 2023-07-31 **\mrcTimewarpIdentity={**⟨*time*⟩**}**

Identity function from $[0, 1]$ into $[0, 1]$.

N 2023-07-31 **mermap/anim/timewarp-identity** (no value, initially set)

Sets `mermap/anim/timewarp` to `\mrcTimewarpIdentity`. This means that time flows constantly.

N 2023-07-31 **\mrcTimewarpSlowStart={**⟨*exponent*⟩**}{**⟨*time*⟩**}** (default 2)

Function term $t^{\langle exponent \rangle}$

N 2023-07-31 **mermap/anim/timewarp-slow-start=**⟨*exponent*⟩ (default 2)

Sets `mermap/anim/timewarp` to `\mrcTimewarpSlowStart{`⟨*exponent*⟩`}`.
If ⟨*exponent*⟩ > 1 is used, time flows slower at the begin of the interval $[0, 1]$. This can be used, if a low scale denominator is present at begin of a movement and is becoming larger later. Here, for smoothness, you may want this to have slower speed near ground and larger speed later.

N 2023-07-31 **\mrcTimewarpSlowFinal={**⟨*exponent*⟩**}{**⟨*time*⟩**}** (default 2)

Function term $1 - (1 - t)^{\langle exponent \rangle}$

N 2023-07-31 **mermap/anim/timewarp-slow-final=**⟨*exponent*⟩ (default 2)

Sets `mermap/anim/timewarp` to `\mrcTimewarpSlowFinal{`⟨*exponent*⟩`}`.
If ⟨*exponent*⟩ > 1 is used, time flows slower at the end of the interval $[0, 1]$. This can be used, if a large scale denominator is present at begin of a movement and is becoming lower later. Here, for smoothness, you may want this to have slower speed near ground at the end.

N 2023-07-31 **\mrcTimewarpSlowStartFinal={**⟨*exponent*⟩**}{**⟨*time*⟩**}** (default 2)

Combination of `\mrcTimewarpSlowStart` and `\mrcTimewarpSlowFinal` using case discrimination for the first and the second half of the intervall.

N 2023-07-31 **mermap/anim/timewarp-slow-start-final=**⟨*exponent*⟩ (default 2)

Sets `mermap/anim/timewarp` to `\mrcTimewarpSlowStartFinal{`⟨*exponent*⟩`}`.
If ⟨*exponent*⟩ > 1 is used, time flows slower at the begin and at the end of the interval $[0, 1]$. This can be used, if a low scale denominator is present at begin of a movement, becomes larger in the middle, and finally is becoming lower later. Here, for smoothness, you may want this to have slower speed near ground at the begin and the end.

10.3 Macros inside the Animation Environment

N 2023-07-31 `\mrcAnimFrame`
N 2023-07-31 `\l_mermap_anim_frame_int`

The current frame number ranging from 1 to `mermap/anim/frames` → P. 82.

N 2023-07-31 `\mrcAnimTime`
N 2023-07-31 `\l_mermap_anim_time_fp`

The current time ranging from 0 to 1.

N 2023-07-31 `\mrcAnimScaleDenom`
N 2023-07-31 `\l_mermap_anim_scaledenom_fp`

The current scale denominator according to `mermap/anim/scale-denominators` → P. 82.

N 2023-07-31 `\mrcAnimLatitude`
N 2023-07-31 `\l_mermap_anim_lat_fp`

The latitude of the current position.

N 2023-07-31 `\mrcAnimLongitude`
N 2023-07-31 `\l_mermap_anim_lon_fp`

The longitude of the current position.

N 2024-07-31 `AnimNP`: The current named position.

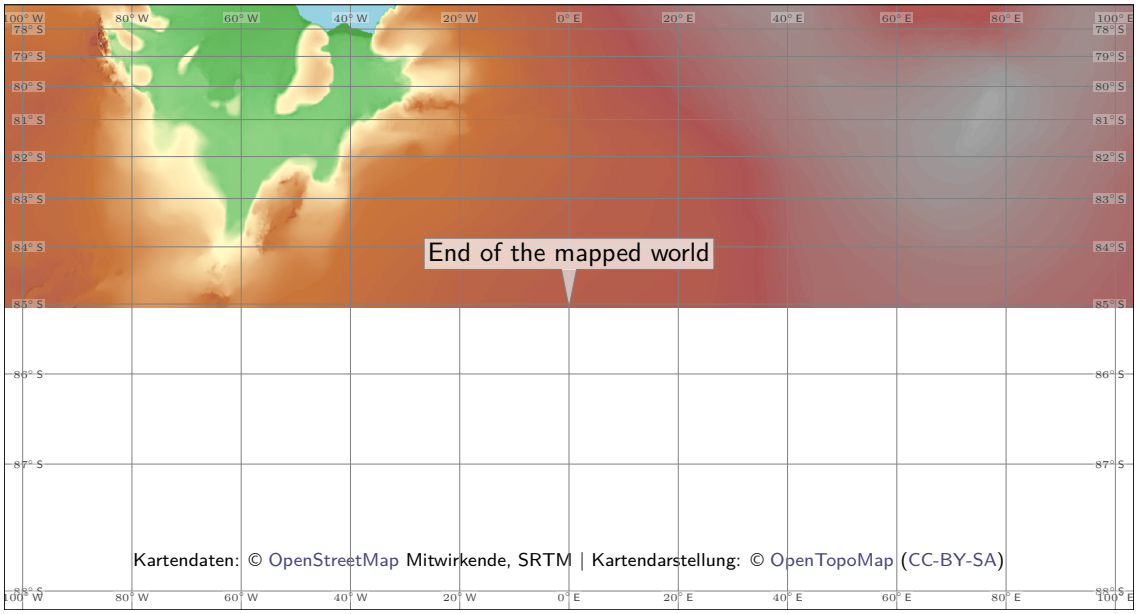
11 Limitations and Caveats

11.1 No Polar Regions

The standard Mercator projection is not suited for north or south polar regions. The Web Mercator projections only covers positions between 85.0511° S and 85.0511° N with map tiles. Equally, the `mercatormap` package does only provide support for this area.

If your map overlaps to south of 85.0511° S or to north of 85.0511° N, compiler errors are possible because of internal computation limitations. In any case, there are no map tiles.

```
\begin{tikzpicture}
  \mrcmap[type=reference,latitude=-85.0511,longitude=0,zoom=3,
    source=opentopomap,
    tex width=\linewidth, tex height=8cm]{}
  \mrcdrawmap
  \node[above=4mm,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
      {\mrcmapattribution};
  \mrcclipmap
  \mrcdrawnetwork[network pieces=10,network distance=10mm]
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
  \mrcmarker{type=pin,position=-85.0511:0,
    contents={End of the mapped world}}
\end{tikzpicture}
```

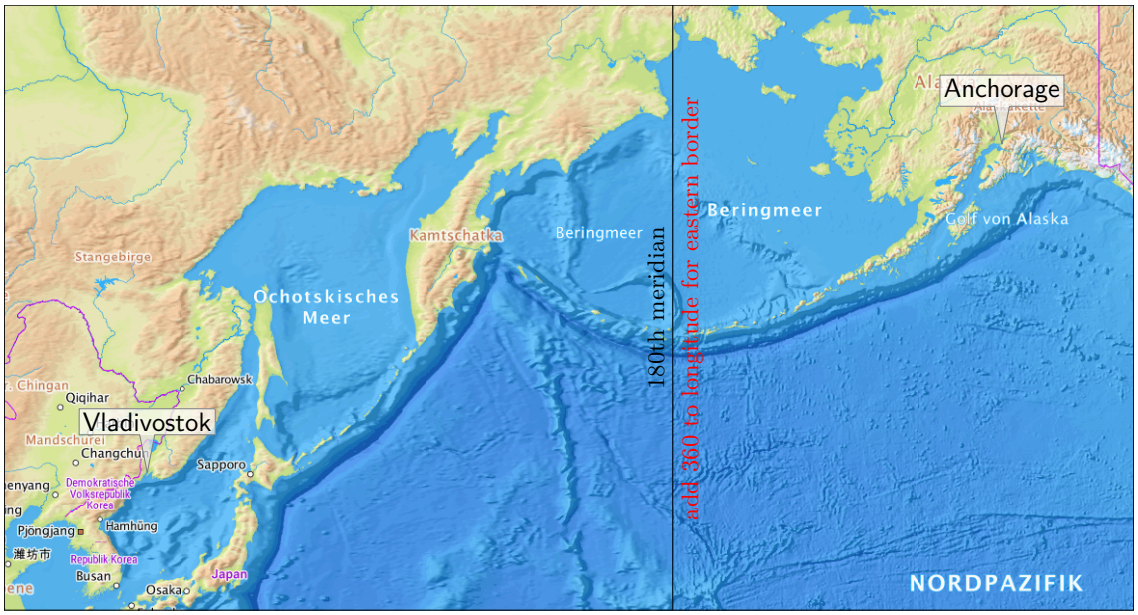


11.2 International Dateline (180th Meridian)

If your map includes the 180th meridian (more or less the international dateline), remember that the eastern hemisphere lies *west* and the western hemisphere lies *east* for your map. Consider the following example displaying Vladivostok and Anchorage. Here, `mermap/supply/west` → P. 30 is set to 130° E (`130`) and `mermap/supply/east` → P. 30 is set to 149° W (`211` instead of `-149`).

There is some automatic correction for positions to fit inside the defined map (see Anchorage in the example below), but *west* should be lower than *east*.

```
\begin{tikzpicture}
  \mrcmap[type=areafit,tex width=\linewidth,tex height=8cm,
    source=topplusopen web,
    south=42,north=62,west=130,east=-149+360]{}
    \mrcdrawmap
    \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
      {\mrcmapattribution};
    \mrcclipmap
  \draw (mrcmap.south west) rectangle (mrcmap.north east);
  \draw (mrc cs:lat=\mrcmapsouth,lon=180)
    -- node[sloped,above] {180th meridian}
       node[sloped,below,red] {add 360 to longitude for eastern border}
  (mrc cs:lat=\mrcmapnorth,lon=180);
  \mrcmarker{type=pin,position=61.22:-149.88,contents={Anchorage}}
  \mrcmarker{type=pin,position=43.12:131.9,  contents={Vladivostok}}
\end{tikzpicture}
```

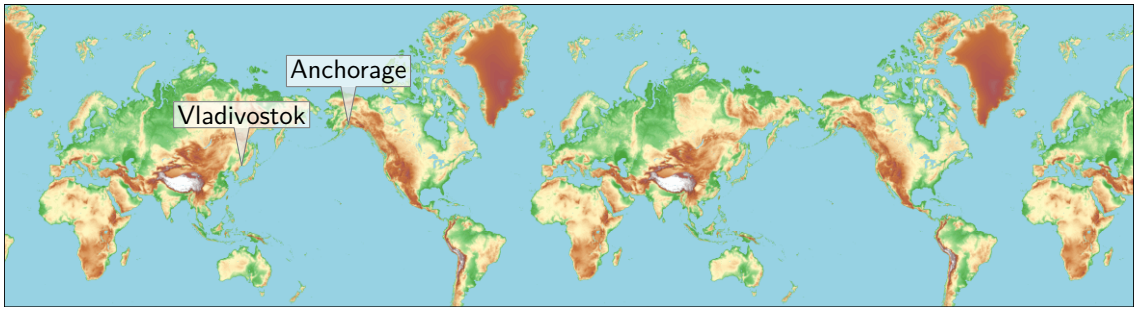


Kartendarstellung: © BKG (2024), Datenquellen

11.3 Very small Scale Maps

If a map has a very small scale, a coordinate point may appear more than once on the map. But, the map coordinates of this package will only pilot to a single point:

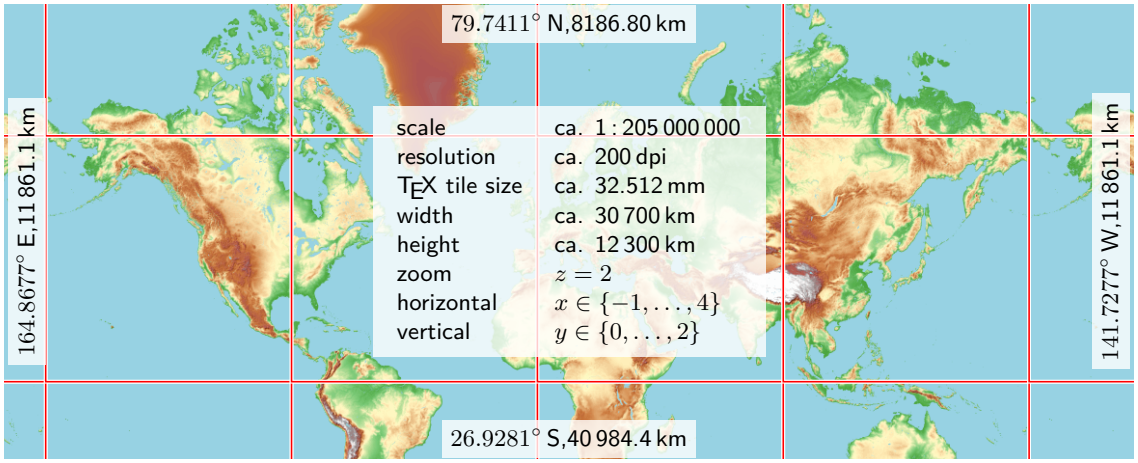
```
\begin{tikzpicture}
  \mrcmap[type=reference,latitude=48.14,longitude=11.57,
    source=opentopomap,
    tex width=\linewidth,tex height=4cm,zoom=1]{}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
    {\mrcmapattribution};   \mrcclipmap
  \path[draw] (mrcmap.south west) rectangle (mrcmap.north east);
  \mrcmarker{type=pin,position=61.22:-149.88,contents={Anchorage}}
  \mrcmarker{type=pin,position=43.12:131.9,contents={Vladivostok}}
\end{tikzpicture}
```



Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)

You cannot trust too much in figures from \mrcprettymapscale → P. 53, \mrcprettymapwidth → P. 53, \mrcprettymapheight → P. 53, etc for large and medium scale maps. For small scale maps, these figures are worse and even misleading. Better do not use them:

```
\mermapset{supply/source=opentopomap}
\begin{tikzpicture}
  \mrcmap[type=reference,latitude=48.14,longitude=11.57,
    tex width=\linewidth,tex height=6cm,zoom=2]{}
  \mrcdrawmap
  \node[below,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south)
    {\mrcmapattribution};   \mrcclipmap  \mrcdrawinfo
\end{tikzpicture}
```

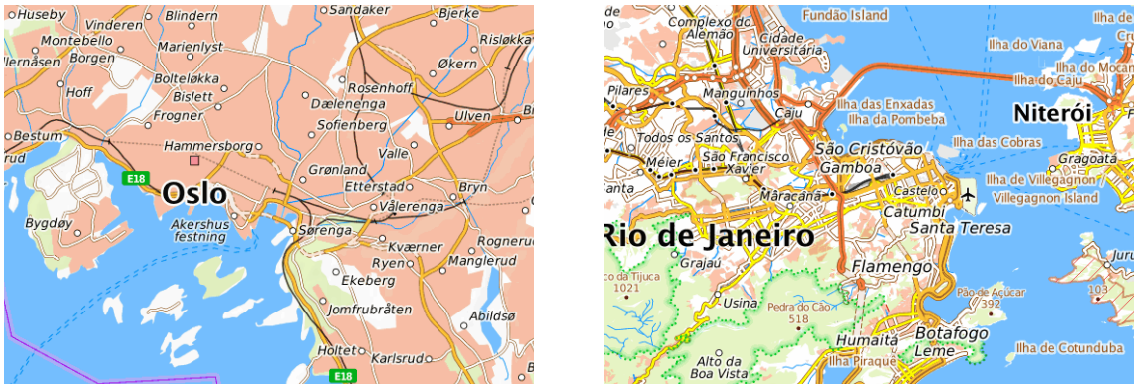


Kartendaten: © OpenStreetMap Mitwirkende, SRTM | Kartendarstellung: © OpenTopoMap (CC-BY-SA)

11.4 Zoom is not Scale

Using the same zoom value for different latitudes can give completely different scales. The following example for Oslo and Rio de Janeiro at zoom 12 scales to 1:151000 and otherwise 1:277000.

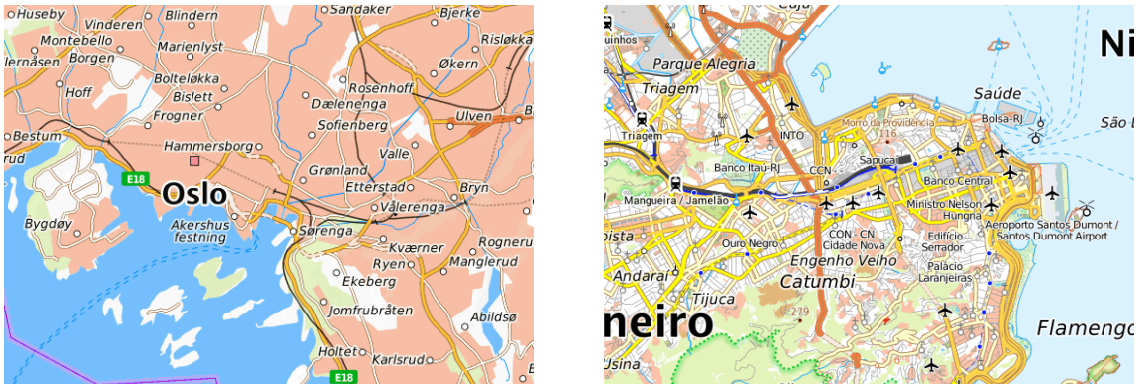
```
\mermapsetsupply{source=topplusopen web, type=reference, zoom=12,
  tex width=7cm, tex height=5cm}
\begin{tikzpicture}
  \mrcmap[latitude=59.91,longitude=10.75]{oslo1}   \mrcdrawmap
\end{tikzpicture}\hfill\begin{tikzpicture}
  \mrcmap[latitude=-22.91,longitude=-43.20]{rio1}  \mrcdrawmap
\end{tikzpicture}
\begin{center}\fontsize{7pt}{7pt}\sffamily\mrcmapattribution\end{center}
```



Kartendarstellung: © BKG (2024), Datenquellen

Same example again, but with `mermap/flex scale` → P. 49 which gives 1:150000 for both cities and comparable maps.

```
\mermapsetsupply{source=topplusopen web, type=reference}
\begin{tikzpicture}
  \mrcmap[latitude=59.91,longitude=10.75, flex reference scale=150000,
    tex width=7cm, tex height=5cm]{oslo2}    \mrcdrawmap
\end{tikzpicture}\hfill\begin{tikzpicture}
  \mrcmap[latitude=-22.91,longitude=-43.20, flex reference scale=150000,
    tex width=7cm, tex height=5cm]{rio2}     \mrcdrawmap
\end{tikzpicture}
\begin{center}\fontsize{7pt}{7pt}\sffamily\mrcmapattribution\end{center}
```



Kartendarstellung: © BKG (2024), Datenquellen

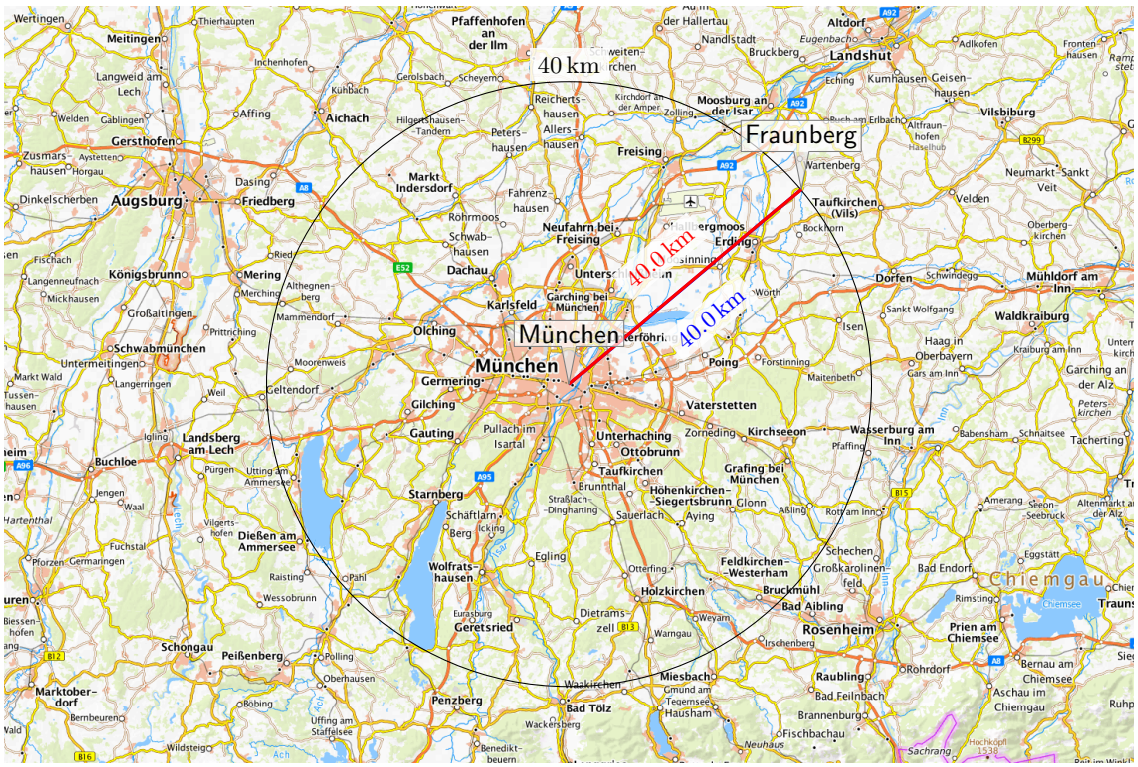
11.5 Distances

There are at least three kinds of distance measures between two points on the map:

- Measure the distance with a ruler on the printed (or displayed) map and multiply with the `\mrcmapscaledenominator` → P. 52. Note that the map scale denominator is only (approximately) correct for the map center.
- Compute the loxodromic distance, e.g. with `\mrcstoreloxodistance` → P. 78 which gives the (approximated) distance following a loxodrome. On our Mercator map, this would be equal to the ruler method from above, if the map scale would be constant.
- Compute the orthodromic distance, e.g. with `\mrcstoreorthodistance` → P. 78 which gives the (approximated) shortest distance following an orthodrome.

Apart from the approximate nature of the implementation, these distances are expected to be quite identical for large scale maps, but not for (very) small scale maps.

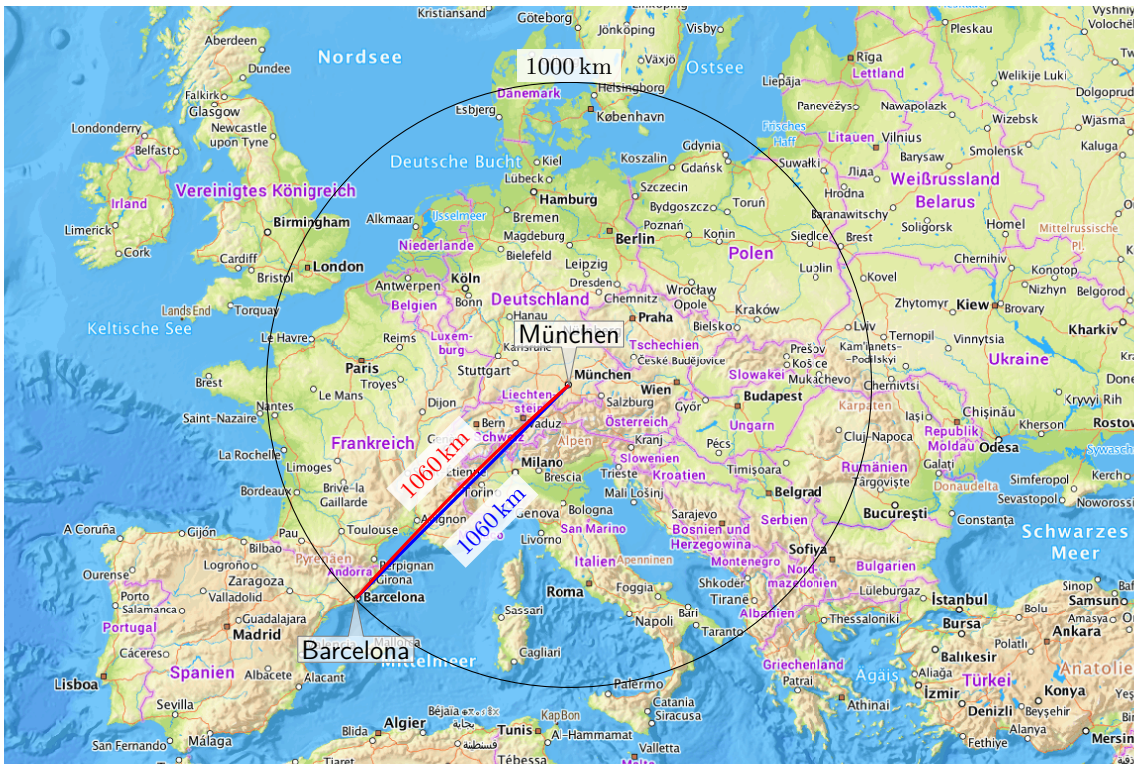
```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.137222}{11.575556}
  \mrcNPdef{fraunberg}{48.3685075}{11.9894209}
  \mrcmap[type=reference, named position=munich,
    flex reference scale=1000000,
    source=topplusopen web,
    tex width=\linewidth, tex height=10cm]{}
  \mrcdrawmap
  \node[below left,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south east)
       {\mrcmapattribution};
  \node[below right,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south west)
       {\mrcprettymapscale};
  \mrcclipmap
  \path[draw] (\mrcNPcs{munich}) circle (\mrckmtotex{40});
  \node[above,fill=white,opacity=0.85,text opacity=1]
    at ([yshift=\mrckmtotex{40}]\mrcNPcs{munich}) {\SI{40}{km}};
  \draw[blue,very thick] (\mrcNPcs{munich}) -- (\mrcNPcs{fraunberg});
  \mrcNPdraworthodrome[red,very thick] {munich}{fraunberg}
  \path (\mrcNPcs{munich}) --
    node[sloped,above=3mm,red,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyorthodistance{munich}{fraunberg} }
    node[sloped,below=3mm,blue,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyloxodistance{munich}{fraunberg} }
    (\mrcNPcs{fraunberg});
  \mrcmarker{type=pin,named position=munich,contents={M\"unchen}}
  \mrcmarker{type=pin,named position=fraunberg,contents={Fraunberg}}
\end{tikzpicture}
```



1 : 1 000 000 Kartendarstellung: © BKG (2024), Datenquellen

The next example gives notable different distances. The aberration is dependent from the center angle.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.137222}{11.575556}
  \mrcNPdef{barcelona}{41.3947688}{2.0787285}
  \mrcmap[type=reference, named position=munich,
    flex reference scale=25000000,
    source=topplusopen web,
    tex width=\linewidth, tex height=10cm]{}
  \mrcdrawmap
  \node[below left,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south east)
       {\mrcmapattribution};
  \node[below right,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south west)
       {\mrcprettymapscale};
  \mrcclipmap
  \path[draw] (\mrcNPcs{munich}) circle (\mrckmtotex{1000});
  \node[above,fill=white,opacity=0.85,text opacity=1]
    at ([yshift=\mrckmtotex{1000}]\mrcNPcs{munich}) {\SI{1000}{km}};
  \draw[blue,very thick] (\mrcNPcs{munich}) -- (\mrcNPcs{barcelona});
  \mrcNPdraworthodrome[red,very thick] {munich}{barcelona}
  \path (\mrcNPcs{munich}) --
    node[sloped,above=3mm,red,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyorthodistance{munich}{barcelona} }
    node[sloped,below=3mm,blue,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyloxodistance{munich}{barcelona} }
    (\mrcNPcs{barcelona});
  \mrcmarker{type=pin,named position=munich,contents={M\"unchen}}
  \mrcmarker{type=pinflip,named position=barcelona,contents={Barcelona}}
\end{tikzpicture}
```

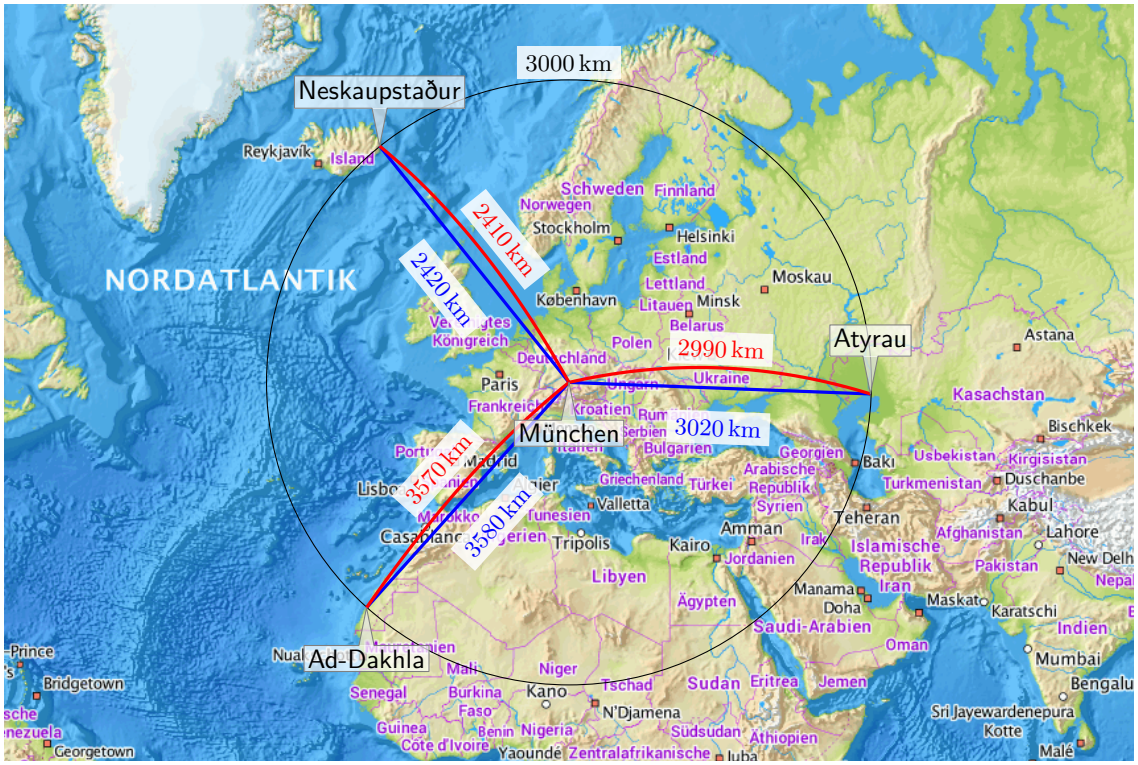


1 : 25 000 000

Kartendarstellung: © BKG (2024), Datenquellen

The following small scale example contains unacceptable deviations from the displayed circle radius of 3000 km. The aberration is highly dependent from the center angle. For such small scale maps, displaying a distance circle should be avoided.

```
\begin{tikzpicture}
  \mrcNPdef{munich}{48.137222}{11.575556}
  \mrcNPdef{neskaup}{65.1446431}{-13.7420082}
  \mrcNPdef{atyrau}{47.0971204}{51.866263}
  \mrcNPdef{dakhla}{24.2026691}{-15.4883971}
  \mrcmap[type=reference, named position=munich,
    flex reference scale=75000000,
    source=topplusopen web,
    tex width=\linewidth, tex height=10cm]{}
  \mrcdrawmap
  \node[below left,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south east)
       {\mrcmapattribution};
  \node[below right,font=\fontsize{7pt}{7pt}\sffamily] at (mrcmap.south west)
       {\mrcprettymapscale};
  \mrcclipmap
  \path[draw] (\mrcNPcs{munich}) circle (\mrckmtotex{3000});
  \node[above,fill=white,opacity=0.85,text opacity=1]
    at ([yshift=\mrckmtotex{3000}]\mrcNPcs{munich}) {\SI{3000}{km}};
  \draw[blue,very thick] (\mrcNPcs{munich}) -- (\mrcNPcs{neskaup});
  \mrcNPdraworthodrome[red,very thick] {munich}{neskaup}
  \path (\mrcNPcs{munich}) --
    node[sloped,above=3mm,red,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyorthodistance{munich}{neskaup} }
    node[sloped,below=3mm,blue,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyloxodistance{munich}{neskaup} }
    (\mrcNPcs{neskaup});
  \draw[blue,very thick] (\mrcNPcs{munich}) -- (\mrcNPcs{atyrau});
  \mrcNPdraworthodrome[red,very thick] {munich}{atyrau}
  \path (\mrcNPcs{munich}) --
    node[sloped,above=3mm,red,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyorthodistance{munich}{atyrau} }
    node[sloped,below=3mm,blue,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyloxodistance{munich}{atyrau} }
    (\mrcNPcs{atyrau});
  \draw[blue,very thick] (\mrcNPcs{munich}) -- (\mrcNPcs{dakhla});
  \mrcNPdraworthodrome[red,very thick] {munich}{dakhla}
  \path (\mrcNPcs{munich}) --
    node[sloped,above=3mm,red,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyorthodistance{munich}{dakhla} }
    node[sloped,below=3mm,blue,fill=white,opacity=0.85,text opacity=1]
         { \mrcNPprettyloxodistance{munich}{dakhla} }
    (\mrcNPcs{dakhla});
  \mrcmarker{type=pinflip,named position=munich,contents={M\"unchen}}
  \mrcmarker{type=pin,named position=neskaup,contents={Neskaupsta\dh ur}}
  \mrcmarker{type=pin,named position=atyrau,contents={Atyrau}}
  \mrcmarker{type=pinflip,named position=dakhla,contents={Ad-Dakhla}}
\end{tikzpicture}
```



1 : 75 000 000

Kartendarstellung: © BKG (2024), Datenquellen

References

[1] Thomas F. Sturm. *Some Principles of Web Mercator Maps and their Computation*. Monograph. Neubiberg: Universität der Bundeswehr München, Mar. 5, 2020. 44 pages. DOI: 10.18726/2020_3.

[2] Till Tantau. *The TikZ and PGF Packages. Manual for version 3.1.10*. Jan. 15, 2023. https://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf.

[3] The LaTeX3 Project. *The LaTeX3 Interfaces*. May 27, 2024.

[4] Joseph Wright. *siunitx – A comprehensive (SI) units package*. June 24, 2024. https://mirrors.ctan.org/macros/latex/contrib/siunitx/siunitx.pdf.

Index