Package: rosm (via r-universe)

October 20, 2026

Type Package

Title Plot Raster Map Tiles from Open Street Map and Other Sources

Version 0.3.2

Encoding UTF-8

Maintainer Dewey Dunnington <dewey@fishandwhistle.net>

Description Download and plot Open Street Map <https://www.openstreetmap.org/>, Bing Maps <https://www.bing.com/maps> and other tiled map sources. Use to create basemaps quickly and add hillshade to vector-based maps.

License GPL-2

Imports curl, jpeg, png, wk, glue, progress, rlang

Suggests sp, plyr, raster, testthat (>= 3.0.0), withr, sf, terra, abind, methods, jsonlite, tiff, vdiffr

URL https://github.com/paleolimbot/rosm

BugReports https://github.com/paleolimbot/rosm/issues

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/pak/sysreqs libjpeg-dev libpng-dev libssl-dev

Repository https://thinkr-open.r-universe.dev

Date/Publication 2026-09-02 03:11:37 UTC

RemoteUrl https://github.com/paleolimbot/rosm

RemoteRef HEAD

RemoteSha 33302debd3def0b464caca9cd7104c3186a4ead9

Contents

as.tile_source 2
has_internet 5
osm_native 5

osm_raster 6
osm_tile 7
osm_tile_covering 8
osm_url 9
osm_url_load_async 10
osm_url_spec 11
set_default_cachedir 12

Index **13**

`as.tile_source` *Deprecated interface*

Description

The previous interface for rosm was written to support idioms that are no longer prevalent in modren r-spatial code. These functions may continue to exist; however, their use is not encouraged and the functions may be removed in a future release.

Usage

```
as.tile_source(x, ..., api_key = NULL)

is.tile_source(x)

source_from_url_format(
  url_format,
  max_zoom = tile.maxzoom.default(),
  min_zoom = 0,
  attribution = NULL,
  extension = tools::file_ext(url_format[1]),
  ...
)

register_tile_source(...)

set_default_tile_source(x, ...)

get_default_tile_source()

osm.types()

bmaps.types()

bmaps.plot(bbox, type = "Aerial", key = NULL, ...)

extract_bbox(x, tolatlon = TRUE, ...)
```

```
osm.plot(
  bbox,
  zoomin = 0,
  zoom = NULL,
  type = NULL,
  forcedownload = FALSE,
  stoponlargerequest = TRUE,
  fusetiles = TRUE,
  cachedir = NULL,
  res = 150,
  project = TRUE,
  progress = c("text", "none"),
  quiet = TRUE,
  api_key = NULL,
  ...
)

osm.image(
  x,
  zoomin = 0,
  zoom = NULL,
  type = NULL,
  forcedownload = FALSE,
  cachedir = NULL,
  progress = c("text", "none"),
  quiet = TRUE,
  api_key = NULL
)

osm.raster(
  x,
  zoomin = 0,
  zoom = NULL,
  type = "osm",
  forcedownload = FALSE,
  cachedir = NULL,
  progress = c("text", "none"),
  quiet = TRUE,
  projection = NULL,
  crop = FALSE,
  filename = NULL,
  resample = "bilinear",
  api_key = NULL,
  ...
)

osm.points(x, y = NULL, epsg = 4326, toepsg = 3857, ...)
```

```
osm.segments(x0, y0, x1 = x0, y1 = y0, epsg = 4326, toepsg = 3857, ...)

osm.lines(x, y = NULL, epsg = 4326, toepsg = 3857, ...)

osm.polygon(x, y = NULL, epsg = 4326, toepsg = 3857, ...)

osm.text(x, y = NULL, labels = seq_along(x), epsg = 4326, toepsg = 3857, ...)

makebbox(n, e, s, w)

zoombbox(bbox, factor = 1, offset = c(0, 0))
```

Arguments

x, y, x0, y0, x1, y1, url_format, max_zoom, min_zoom, attribution, extension	Deprecated
...	Arguments passed to other methods
api_key	API key used for CARTO map types. If NULL, the value of the CARTO_API_KEY environment variable is used.
bbox	A bounding box as generated by sp::bbox()
type	A map type; one of that returned by osm.types. User defined types are possible by defining tile.url.TYPENAME <- function(xtile, ytile, zoom){} and passing TYPENAME as the type argument.
key, tolatlon, epsg, toepsg, labels, n, e, s, w, factor, offset	Deprecated
zoomin	The amount by which to adjust the automatically calculated zoom (or manually specified if the zoom parameter is passed). Use +1 to zoom in, or -1 to zoom out.
zoom	Manually specify the zoom level (not reccomended; adjust zoomin or res instead.
forcedownload	TRUE if cached tiles should be re-downloaded. Useful if some tiles are corrupted.
stoponlargerequest	By default osm.plot will only load 32 tiles at a time. If plotting at a higher resolution it may be necessary to pass true here.
fusetiles	TRUE if tiles should be fused into a single image. This is the default because white lines appear between tiles if it is set to FALSE. PDFs appear not to have this problem, so when plotting large, high resolution PDFs it may be faster (and more memory efficient) to use fusetiles=FALSE.
cachedir	The directory in which tiles should be cached. Defaults to getwd()/rosm.cache.
res	The resolution used to calculate scale.
project	TRUE if tiles should be projected to a pseudo-mercator projection, FALSE if lat/lon should be maintained. Becuase sp::plot adjusts the aspect according to latitude for lat/lon coordinates, this makes little difference at high zoom and may make plotting overlays more convenient. Defaults to TRUE.
progress	A progress bar to use, or "none" to suppress progress updates

quiet	Pass FALSE to see more error messages, particularly if your tiles do not download/load properly.
projection	A map projection in which to reproject the RasterStack as generated by CRS() or Spatial*@proj4string. If a Spatial* object is passed as the first argument, this argument will be ignored.
crop	TRUE if results should be cropped to the specified bounding box (see x), FALSE otherwise.
filename	A filename to which the raster should be written (see raster::writeRaster()). Use a ".tif" extension to write as a GeoTIFF.
resample	One of "ngb" (nearest neighbour) or "bilinear". Passed to projectRaster.

---

has_internet *Check for Internet*

Description

Used to skip tests and examples for this package when offline.

Usage

```
has_internet()
```

Value

TRUE if the internet is available, false otherwise

Examples

```
has_internet()
```

---

osm_native *Coordinate helpers*

Description

Coordinate helpers

Usage

```
osm_native(x, y)

osm_lnglat(lng, lat)

osm_crs_native()

osm_ensure_lnglat(pt)

osm_ensure_native(pt)
```

Arguments

x, y	Ordinate values in EPSG:3857 (Spherical Mercator in meters)
lng, lat	Coordinate values for longitude/latitude in degrees.
pt	A vector of points as coerced by [wk::as_xy()]. The CRS for these points is considered.

Value

- ‘osm_native()‘, ‘osm_lnglat()‘, ‘osm_ensure_native()‘, and ‘osm_ensure_lnglat()‘ return a [wk::xy()] with the appropriate crs - ‘osm_crs_native()‘ returns a value that can be used as the [wk::wk_crs()] of a vector.

Examples

```
osm_lnglat(-64, 45)
osm_ensure_native(osm_lnglat(-64, 45))
osm_ensure_lnglat(
  osm_ensure_native(osm_lnglat(-64, 45))
)
```

---

osm_raster *Load an Open Street Map image*

Description

Load an Open Street Map image

Usage

```
osm_raster(
  bbox,
  spec,
  zoom = osm_zoom_num_tiles(6),
  cache_spec = NULL,
  quiet = NA
)
```

Arguments

bbox	A [wk::rct()] or object with a [wk::wk_bbox()] method.
spec	An [osm_url_spec()]
zoom	A zoom level or an auto zoom specifier like [osm_zoom_num_tiles()].
cache_spec	An optional [osm_url_spec()] or character vector to be used as the cache.
quiet	Use 'TRUE' for fewer messages or 'FALSE' for more messages.

Value

A [wk::grd_rct()] whose data member is a nativeRaster.

Examples

```
bounds <- wk::rct(
  -7476083, 5349058,
  -6594103, 6243203,
  crs = osm_crs_native()
)

(grd <- osm_raster(bounds, osm_url_spec()))
plot(grd)
```

---

osm_tile *Low-level tile math*

Description

Open Street Map operates using a system of tiles whose value and bounds are easily calculated from WGS84 longitude/latitude. These functions convert between tile system coordinates and longitude/latitude.

Usage

```
osm_tile(pt, zoom)

osm_tile_quadkey(tile)

osm_tile_top_left(tile, crs = osm_crs_native())

osm_tile_envelope(tile, crs = osm_crs_native())
```

Arguments

pt	A vector of points as coerced by [wk::as_xy()]. The CRS for these points is considered.
zoom	A zoom level, generally between 0 and 21, with higher values representing a smaller (i.e., more detailed) tile.
tile	A 'data.frame()' with columns 'x', 'y', and 'zoom'.
crs	A target CRS. Either [wk::wk_crs_longlat()] or [osm_crs_native()].

Value

- 'osm_tile()': A 'data.frame()' with columns 'x', 'y', and 'zoom'. - 'osm_tile_top_left()': A [wk::xy()] of the top-left (northwest) corner of the tile. - 'osm_tile_envelope()': A [wk::rct()] of the tile bounds.

Examples

```
(tiles <- osm_tile(osm_lnglat(-64, 45), zoom = 0:5))
osm_tile_envelope(tiles)
```

---

osm_tile_covering *Get an OSM tile covering*

Description

Get an OSM tile covering

Usage

```
osm_tile_covering(bbox, zoom = osm_zoom_num_tiles(6))

osm_zoom_num_tiles(num_tiles)
```

Arguments

bbox	A [wk::rct()] or object with a [wk::wk_bbox()] method.
zoom	A zoom level or an auto zoom specifier like [osm_zoom_num_tiles()].
num_tiles	The minimum number of tiles to use when choosing a zoom level.

Value

- 'osm_tile_covering()' returns a 'data.frame()' with columns x, y, and zoom.

Examples

```
bounds <- wk::rct(
  -7514064, 5009380,
  -6261722, 6261715,
  crs = osm_crs_native()
)

osm_tile_covering(bounds)
```

`osm_url` *Resolve a tile into a URL*

Description

Resolve a tile into a URL

Usage

```
osm_url(tile, spec)
```

Arguments

`tile`	A ‘data.frame()‘ with columns ‘x‘, ‘y‘, and ‘zoom‘.
`spec`	An [osm_url_spec()]

Value

A character vector of URLs

Examples

```
bounds <- wk::rct(
  -7514064, 5009380,
  -6261722, 6261715,
  crs = osm_crs_native()
)

tiles <- osm_tile_covering(bounds, zoom = 6)
osm_url(tiles, osm_url_spec())
```

`osm_url_load_async` *Load tile URLs*

Description

Load tile URLs

Usage

```
osm_url_load_async(tile, spec, callback = NULL, cache_spec = NULL)
```

Arguments

`tile`	A ‘data.frame()‘ with columns ‘x‘, ‘y‘, and ‘zoom‘.
`spec`	An [osm_url_spec()]
`callback`	A function to be run for each tile fetch or NULL to do nothing. The callback is always called with two arguments: the first is the subset of ‘tile‘ for which this URL applies (typically one row but can be more than one in some corner cases); the second is the curl response object whose useful elements are url, status_code, type, and content.
`cache_spec`	An optional [osm_url_spec()] or character vector to be used as the cache.

Value

‘tile‘, invisibly.

Examples

```
bounds <- wk::rct(
  252185, 4815826, 739729, 5210280,
  crs = "EPSG:32620"
)

tiles <- osm_tile_covering(bounds, zoom = 5)

osm_url_load_async(
  tiles,
  osm_url_spec_example(),
  function(tile, res) {
    str(tile)
    str(res)
  }
)
```

`osm_url_spec` *Tile URL Specification*

Description

See <https://github.com/roblabs/xyz-raster-sources> for a number of useful values to use for 'server_url'.

Usage

```
osm_url_spec(
  server_url = "https://tile.openstreetmap.org/{z}/{x}/{y}.png",
  block_size = c(256, 256),
  min_zoom = 0,
  max_zoom = 18,
  content_type = NA_character_,
  name = NULL
)

osm_url_spec_example()

as_osm_url_spec(x, ..., name = NULL)
```

Arguments

`server_url`	A url using '{x}', '{y}', and '{z}' for the x, y, and zoom level to be replaced. This can be any URL; non-URLs are assumed to be local file paths relative to the current working directory at the time of the download.
`block_size`	The pixel size of each image
`min_zoom, max_zoom`	The min/max zoom that this tile specification can handle
`content_type`	A MIME type or NA to guess the type from 'server_url'.
`name`	A name for this spec. Useful for cache specifications.
`x`	An object to convert to an osm_url_spec
`...`	Passed to S3 methods

Value

An object of class osm_url_spec.

Examples

```
osm_url_spec()
```

`set_default_cachedir` *Set/Get the Default Tile Cache Location*

Description

The default tile cache location is the "rosm.cache" folder in the current working directory, but for a variety of reasons it may be desirable to use one cache directory for all calls in a script. This must be called every time the namespace is loaded.

Usage

```
set_default_cachedir(cachedir)

get_default_cachedir()
```

Arguments

`cachedir`	A path to use as the cache directory (relative to the working directory). Use NULL to reset to the default.

Value

The previous cache directory, invisibly.

Examples

```
set_default_cachedir(tempfile())
get_default_cachedir()
(set_default_cachedir(NULL))
```

Index

as.tile_source, 2
as_osm_url_spec *(*osm_url_spec*)*, 11

bmaps.plot *(*as.tile_source*)*, 2
bmaps.types *(*as.tile_source*)*, 2

extract_bbox *(*as.tile_source*)*, 2

get_default_cachedir
        *(*set_default_cachedir*)*, 12
get_default_tile_source
        *(*as.tile_source*)*, 2

has_internet, 5

is.tile_source *(*as.tile_source*)*, 2

makebbox *(*as.tile_source*)*, 2

osm.image *(*as.tile_source*)*, 2
osm.lines *(*as.tile_source*)*, 2
osm.plot *(*as.tile_source*)*, 2
osm.points *(*as.tile_source*)*, 2
osm.polygon *(*as.tile_source*)*, 2
osm.raster *(*as.tile_source*)*, 2
osm.segments *(*as.tile_source*)*, 2
osm.text *(*as.tile_source*)*, 2
osm.types, *4*
osm.types *(*as.tile_source*)*, 2
osm_crs_native *(*osm_native*)*, 5
osm_ensure_lnglat *(*osm_native*)*, 5
osm_ensure_native *(*osm_native*)*, 5
osm_lnglat *(*osm_native*)*, 5
osm_native, 5
osm_raster, 6
osm_tile, 7
osm_tile_covering, 8
osm_tile_envelope *(*osm_tile*)*, 7
osm_tile_quadkey *(*osm_tile*)*, 7
osm_tile_top_left *(*osm_tile*)*, 7
osm_url, 9
osm_url_load_async, 10
osm_url_spec, 11
osm_url_spec_example *(*osm_url_spec*)*, 11
osm_zoom_num_tiles *(*osm_tile_covering*)*,
        8

projectRaster, *5*

register_tile_source *(*as.tile_source*)*, 2

set_default_cachedir, 12
set_default_tile_source
        *(*as.tile_source*)*, 2
source_from_url_format
        *(*as.tile_source*)*, 2

zoombbox *(*as.tile_source*)*, 2