

Package: fwatlasbc (via r-universe)

October 15, 2026

Title Freshwater Atlas British Columbia

Version 0.0.1.9028

Description Queries the Freshwater Atlas (FWA) of British Columbia through the 'fwapg' API
<<https://www.hillcrestgeo.ca/fwapg/index.html>> via the 'fwapgr' package and returns spatial data as simple features ('sf').
Also provides tools to convert streams to and from river meters, snap points to streams, and split, prune and join stream networks.

License MIT + file LICENSE

URL <https://poissonconsulting.github.io/fwatlasbc/>

Depends R (>= 4.1)

Imports chk, digest, dplyr, fwapg (>= 0.1.0.9013), glue, lifecycle, lwgeom, purrr, rlang, sf, stats, stringr, tibble, tidyplus, tidyr, tidyselect, units

Suggests covr, elevatr, googleway, hms, httr, mapview, progress, readr, testthat (>= 3.0.0)

Remotes poissonconsulting/fwapg

Config/Needs/website poissonconsulting/poissontemplate

Config/roxygen2/version 8.1.0.9000

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://poissonconsulting.r-universe.dev>

Date/Publication 2026-10-15 07:25:20 UTC

RemoteUrl <https://github.com/poissonconsulting/fwatlasbc>

RemoteRef HEAD**RemoteSha** a192327661ffd2809660faf8e88b64a1ba6ac0fd**Contents**

fwa_add_blk_to_lon_lat	3
fwa_add_blks_to_stream_name	4
fwa_add_collection_to_polygon	4
fwa_add_cut_to_rms	6
fwa_add_downstream_split_to_rms	6
fwa_add_elevatr_elevation_to_point	7
fwa_add_gm_elevation_to_point	8
fwa_add_intersection_to_geometry	9
fwa_add_new_blk_rm_to_blk_rm	9
fwa_add_parent_to_streams	10
fwa_add_point_to_stream_measure	11
fwa_add_rms_to_blk	12
fwa_add_section_to_rms	13
fwa_add_stream_names_to_blk	14
fwa_add_upstream_split_to_rms	14
fwa_add_watershed_to_blk	15
fwa_collection_name	16
fwa_convert_rms_to_streams	16
fwa_convert_stream_names_to_blks	17
fwa_convert_stream_network_to_rms	17
fwa_convert_streams_to_rms	18
fwa_find_stream_names	19
fwa_flip_streams	20
fwa_get_section_from_rms	21
fwa_get_segment_from_rms	21
fwa_join_stream_segments	22
fwa_mapview	23
fwa_mapview_rms	23
fwa_mapview_rms_to_rms	24
fwa_mapview_streams	25
fwa_order_segments	26
fwa_parent_blk_rms	27
fwa_parent_stream_name_rms	27
fwa_prune_rms	28
fwa_reverse_streams	28
fwa_snap_rm_to_point	29
fwa_snap_rm_to_rms	30
fwa_snap_rms_to_rms	31
fwa_snap_stream_measure_to_point	32
fwa_stitch_segments	33
fwa_stream_distances_to_matrix	34
fwa_stream_name	35

fwa_add_blk_to_lon_lat 3

fwa_swap_branches_rms 35

Index 37

fwa_add_blk_to_lon_lat

Add Blue Line Key to Longitude and Latitude

Description

Adds integer blue line key (blk) and numeric river meter (rm) column and sfc point (geometry) column of the closest point on the stream network (by default within 5 km) to the point specified by the lon and lat (WGS84).

Usage

```
fwa_add_blk_to_lon_lat(  
  x,  
  tolerance = 5000,  
  limit = 1,  
  epsg = getOption("fwa.epsg", 3005),  
  nocache = getOption("fwa.nocache", FALSE)  
)
```

Arguments

<i>x</i>	A data frame with numeric longitude (long) and latitude (lat) columns.
<i>tolerance</i>	A number of the tolerance in m.
<i>limit</i>	A positive whole number indicating the maximum number of features to return.
<i>epsg</i>	A positive whole number of the EPSG projection for the geometry.
<i>nocache</i>	A flag specifying whether or not to cache results.

Details

If a match isn't found the row is dropped.

Value

An sf tibble with the columns of *x* plus integer column *blk*, real columns *rm* and *distance_to_lon_lat* and sfc point column *geometry*

Examples

```
## Not run:  
fwa_add_blk_to_lon_lat(data.frame(lon = -132.26, lat = 53.36))  
  
## End(Not run)
```

fwa_add_blks_to_stream_name

Add Blue Line Key(s) to Stream Name

Description

Adds blue line keys (blk) to stream names. There may be more than one stream with the same name.

Usage

```
fwa_add_blks_to_stream_name(x, stream_name = fwatlasbc::fwa_stream_name)
```

Arguments

x A data frame with character column stream_name.
stream_name A data frame with whole numeric column blk and character column stream_name.

Value

A tibble with the columns of x plus an integer column blk.

Examples

```
fwa_add_blks_to_stream_name(data.frame(stream_name = "Sangan River"))
```

fwa_add_collection_to_polygon

Add Collection to Polygon

Description

Adds collection to a polygon such as a watershed. If the active sfc polygon column is called geometry it is replaced by the geometry column of the collection. If the collection includes a blue_line_key column the values are copied to column blk replacing any existing values.

Usage

```
fwa_add_collection_to_polygon(
  x,
  collection = "stream_network",
  intersect = FALSE,
  filter = NULL,
  limit = 10000,
  offset = 0,
```

```

    properties = NULL,
    transform = NULL,
    epsg = getOption("fwa.epsg", 3005),
    nocache = getOption("fwa.nocache", FALSE)
  )

fwa_add_collection_to_watershed(
  x,
  collection = "stream_network",
  intersect = FALSE,
  filter = NULL,
  limit = 10000,
  offset = 0,
  properties = NULL,
  transform = NULL,
  epsg = getOption("fwa.epsg", 3005),
  nocache = getOption("fwa.nocache", FALSE)
)

```

Arguments

x	A sf object with an active sfc polygon column.
collection	A character string of the collection.
intersect	A logical vector specifying whether to intersect the individual features with the polygon as opposed to just including the features that intersect it.
filter	A named vector or list of the filter(s) to apply, where the list names correspond to column names and the list values correspond to the desired value, e.g. <code>list(gnis_name = "Sangan River")</code> .
limit	A positive whole number indicating the maximum number of features to return.
offset	A positive whole number indicating the offset of start of returned results.
properties	A vector of strings of the column names to include. If NULL (default), all columns are retained.
transform	A character vector with the name of the <i>valid transform</i> function followed by the parameter values (e.g. <code>c("ST_Simplify", 100)</code>)
epsg	A positive whole number of the epsg to transform features to.
nocache	A flag specifying whether or not to cache results.

Value

An sf object

Functions

- `fwa_add_collection_to_watershed()`: **[Superseded]**
Deprecated for `fwa_add_collection_to_polygon()`.

See Also

```
fwapgr::fwa_collection().
```

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001))
fwa_add_collection_to_polygon(watershed)

## End(Not run)
```

fwa_add_cut_to_rms	<i>Add Cut to River Meter</i>
--------------------	-------------------------------

Description

Adds value(s) in cut column in y to to column of same name in x based on blk and rm_start and rm_end in y.

Usage

```
fwa_add_cut_to_rms(x, y, cut = "cut")
```

Arguments

- x A data frame with integer columns blk and rm.
- y A data frame with integer columns blk, rm_start, rm_end and column specified in cut.
- cut A string of the name of the column in y with the values.

Value

A tibble of x with cut column from y.

fwa_add_downstream_split_to_rms	<i>Add Split to River Meter</i>
---------------------------------	---------------------------------

Description

Splits river meters with parent_blk and parent_rm columns into upstream (TRUE) versus not (FALSE).

Usage

```
fwa_add_downstream_split_to_rms(x, y)
```

Arguments

x	A data frame with integer columns blk, rm, parent_blk and parent_rm.
y	A data frame with integer columns blk and rm and character column name.

Value

A tibble of x with a logical column for each name in name.

```
fwa_add_elevatr_elevation_to_point
```

Add elevatr Elevation to Point

Description

Adds elevations from the Amazon Web Services Terrain Tiles using `elevatr::get_elev_point()`. Unlike `fwa_add_gm_elevation_to_point()` no API key is required.

Usage

```
fwa_add_elevatr_elevation_to_point(x, zoom = 12)
```

Arguments

x	An sf object of spatial points.
zoom	A whole number between 1 and 14 of the zoom level of the Terrain Tiles. Higher zoom levels have a finer resolution but take longer to download.

Details

The connection timeout (s) and the number of attempts if the download times out are set by the options `fwa.elevatr_timeout` (by default 10) and `fwa.elevatr_tries` (by default 3). For example `options(fwa.elevatr_timeout = 120, fwa.elevatr_tries = 5)`.

Value

An updated version of x with numeric column elevation.

See Also

`elevatr::get_elev_point()` and `fwa_add_gm_elevation_to_point()`

Examples

```
## Not run:
rm <- fwa_add_rms_to_blk(data.frame(blk = 356308001))
fwa_add_elevatr_elevation_to_point(rm)

## End(Not run)
```

`fwa_add_gm_elevation_to_point`*Add Google Maps Elevation to Point*

Description

Add Google Maps Elevation to Point

Usage

```
fwa_add_gm_elevation_to_point(  
  x,  
  chunk_size = 300L,  
  digits = 7,  
  key = Sys.getenv("GOOGLE_MAPS_ELEVATION_API_KEY")  
)
```

Arguments

<code>x</code>	An sf object of spatial points.
<code>chunk_size</code>	The number of rows to include in each API query.
<code>digits</code>	The number of digits to round the latitude and longitude by before querying the elevation from the API
<code>key</code>	A string of the Google Maps Elevation API key.

Value

An updated version of `x` with numeric column elevation.

See Also

```
googleway::google_elevation()
```

Examples

```
## Not run:  
rm <- fwa_add_rms_to_blk(data.frame(blk = 356308001))  
fwa_add_gm_elevation_to_point(rm)  
  
## End(Not run)
```

```
fwa_add_intersection_to_geometry
```

Add Intersection to Geometry

Description

Adds a logical column for each name in name in y indicating whether each element of x intersects with the element of y.

Usage

```
fwa_add_intersection_to_geometry(x, y)
```

Arguments

x	An sf data frame.
y	An sf data frame with character column name.

Value

A copy of x with a logical column for each name in name indicating whether each element of x intersects with the element of y.

```
fwa_add_new_blk_rm_to_blk_rm
```

Add New Blue Line Key and River Meter to Blue Line Key and River Meter

Description

Adds new blk and rm values to existing blk and rm values based on look up in second data frame.

Usage

```
fwa_add_new_blk_rm_to_blk_rm(
  x,
  y,
  blk = "blk",
  rm = "rm",
  blk2 = "blk",
  rm2 = "rm",
  new_blk = "new_blk",
  new_rm = "new_rm",
  new_blk_to = new_blk,
  new_rm_to = new_rm
)
```

Arguments

x	A tibble (or sf object) with blk and rm columns.
y	A tibble (or sf object) with unique blk and rm columns and new blk (optional) and rm columns.
blk	A string of the name of the blk column in x.
rm	A string of the name of the rm column in x.
blk2	A string of the name of the blk column in y.
rm2	A string of the name of the rm column in y.
new_blk	An optional string of the name of the new blk column in y. If NULL it is not used in the join.
new_rm	A string of the name of the new rm column in y.
new_blk_to	A string specifying the name to rename the new blk column with.
new_rm_to	A string specifying the name to rename the new rm column with.

Value

An updated version of x with additional columns for the new blk and rm columns.

See Also

[fwa_add_new_rm_to_blk_rm()]

fwa_add_parent_to_streams

Add Parent to Streams

Description

Adds the parent blue line key (parent_blk) and the river meter on the parent where the stream joins it (parent_rm) to stream linestrings together with the distance (m) from the mouth to the parent (distance_to_parent). The parent is the nearest other stream to the mouth of each stream. If the mouth is more than gap m from the nearest other stream then parent_blk, parent_rm and distance_to_parent are NA.

Usage

```
fwa_add_parent_to_streams(
  x,
  gap = 1,
  elevation = FALSE,
  source = NULL,
  zoom = 12
)
```

Arguments

x	An sf tibble with a column blk and linestrings of streams (or stream segments).
gap	A positive real number specifying the maximum gap (m) between the mouth of stream and its parent stream to be considered connected.
elevation	A flag specifying whether to use the elevation to determine stream direction (or use the direction of the provided linestrings)
source	A string of the source of the elevations if elevation = TRUE. Possible values are "elevatr" (no API key required) and "google" (requires a Google Maps Elevation API key). If NULL (the default) and elevation = TRUE "google" is used but this is [Deprecated] and source should be set explicitly.
zoom	A whole number between 1 and 14 of the zoom level of the Terrain Tiles. Higher zoom levels have a finer resolution but take longer to download. Ignored if elevation = FALSE or source = "google".

Details

The stream segments for each blk are first joined using `fwa_join_stream_segments()`. The start of each joined stream is taken to be its mouth and river meters on the parent are measured from its start. Unless elevation = TRUE, streams digitized from source to mouth must first be reversed using `fwa_reverse_streams()`. The geometry of x is not altered.

Value

A copy of x with integer column parent_blk and numeric columns parent_rm and distance_to_parent.

See Also

`fwa_convert_streams_to_rms()`

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001, rm = 1000))
network <- fwa_add_collection_to_polygon(watershed)
network <- select(network, blk = blue_line_key)
fwa_add_parent_to_streams(network)

## End(Not run)
```

fwa_add_point_to_stream_measure

Add Point to Stream Measure

Description

Adds point geometry for stream measure in m based on the blue line key (blk) and the proportion of the stream measure along the stream. If the proportion is ≥ 1 then the geometry is set to the top of the stream.

Usage

```
fwa_add_point_to_stream_measure(x, streams, ...)
```

Arguments

x	A data frame with columns blk and stream_measure.
streams	An sf object of spatial linestrings with blk column.
...	Additional columns to group by when assigning.

Value

An updated version of x with numeric column proportion giving the proportion of the stream measure along the stream and a geometry column.

See Also

[fwa_snap_stream_measure_to_point\(\)](#)

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001, rm = 1000))
network <- fwa_add_collection_to_polygon(watershed)
network$blk <- network$blue_line_key
streams <- fwa_join_stream_segments(network)
points <- fwa_add_rms_to_blk(data.frame(blk = 356308001))
points <- fwa_snap_stream_measure_to_point(points, streams)
points <- points[c("blk", "stream_measure")]
fwa_add_point_to_stream_measure(points, streams)

## End(Not run)
```

fwa_add_rms_to_blk	<i>Add River Meters to Blue Line Key</i>
--------------------	--

Description

Adds distances (rm) and spatial coordinates (geometry) of regularly spaced points along blue line key (blk). All distances which are in meters are from the river mouth.

Usage

```
fwa_add_rms_to_blk(
  x,
  interval = 1000,
  start = 0,
  end = Inf,
  epsg = getOption("fwa.epsg", 3005),
```

```
  nocache = getOption("fwa.nocache", FALSE)
)
```

Arguments

x	A data frame with integer column blk.
interval	A whole numeric of the distance between points.
start	A whole numeric of the start distance.
end	An integer of the end distance.
epsg	A positive whole number of EPSG projection for the coordinates.
nocache	A flag specifying whether or not to cache results.

Value

An sf tibble with the columns of x plus integer column rm and sf column geometry.

Examples

```
## Not run:
fwa_add_rms_to_blk(data.frame(blk = 356308001))

## End(Not run)
```

```
fwa_add_section_to_rms
```

Add Section to River Meter

Description

Adds section column in y to x based on blk and end rm in y. All rms in x up to and including the end rm but before the previous end rm are assigned the section column value (which can be missing).

Usage

```
fwa_add_section_to_rms(x, y, section = "section")
```

Arguments

x	A data frame with integer columns blk and rm.
y	A data frame with integer columns blk and rm and column specified in section.
section	A string of the name of the column in y (can include missing values).

Value

A tibble of x with section column from y.

fwa_add_stream_names_to_blk

Add Blue Line Key(s) to Stream Name

Description

Adds blue line keys (blk) to stream names. There may be more than one stream with the same name.

Usage

```
fwa_add_stream_names_to_blk(x, stream_name = fwatlasbc::fwa_stream_name)
```

Arguments

x A data frame with whole numeric column blk.
stream_name A data frame with whole numeric column blk and character column stream_name.

Value

A tibble with the columns of x plus an integer column blk.

Examples

```
fwa_add_stream_names_to_blk(data.frame(blk = 360886335L))
```

fwa_add_upstream_split_to_rms

Add Split to River Meter

Description

Splits river meters with parent_blk and parent_rm columns into upstream (TRUE) versus not (FALSE).

Usage

```
fwa_add_upstream_split_to_rms(x, y)
```

```
fwa_add_split_to_rms(x, y)
```

Arguments

x A data frame with integer columns blk, rm, parent_blk and parent_rm.
y A data frame with integer columns blk and rm and character column name.

Value

A tibble of x with a logical column for each name in name.

Functions

- fwa_add_split_to_rms(): Soft-deprecated **[Deprecated]**

fwa_add_watershed_to_blk

Add Watershed to Blue Line Key

Description

Adds polygon (geometry) of aggregated fundamental watersheds to blue line key (blk). The rm distances which is in meters is from the river mouth.

Usage

```
fwa_add_watershed_to_blk(
  x,
  exclude = FALSE,
  epsg = getOption("fwa.epsg", 3005),
  nocache = getOption("fwa.nocache", FALSE)
)
```

Arguments

x	An sf object with a polygon sfc column specifying watersheds and an optional rm column specifying the river meter. The rm is set to be 0 if missing.
exclude	A logical vector specifying whether to exclude the fundamental watershed in which the start falls.
epsg	A positive whole number of the epsg to transform features to.
nocache	A flag specifying whether or not to cache results.

Value

An sf tibble with the columns of x plus sf column geometry.

A sf object

See Also

[fwapgr::fwa_watershed_at_measure\(\)](#).

Examples

```
## Not run:
fwa_add_watershed_to_blk(data.frame(blk = 356308001))

## End(Not run)
```

fwa_collection_name	<i>Collection Name</i>
---------------------	------------------------

Description

A tibble of collection names.

Usage

```
fwa_collection_name
```

Format

An object of class spec_tbl_df (inherits from tbl_df, tbl, data.frame) with 12 rows and 2 columns.

Examples

```
fwa_collection_name
```

fwa_convert_rms_to_streams	<i>Convert River Metres to Streams</i>
----------------------------	--

Description

Casts river metre points to linestrings by blk.

Usage

```
fwa_convert_rms_to_streams(x)
```

Arguments

x An point sf object with whole numeric columns blk and rm.

Value

A linestring sf object with whole numeric columns blk and rm.

`fwa_convert_stream_names_to_blks`*Converts Stream Names to Blue Line Keys*

Description

Each stream name is converted to a blue line key by calculating its integer hash.

Usage

```
fwa_convert_stream_names_to_blks(names)
```

Arguments

names	A character vector of stream names.
-------	-------------------------------------

Details

This function is only expected to be used when a blue line key does not already exist.

Value

A positive integer vector of blue line keys.

Examples

```
fwa_convert_stream_names_to_blks(c("a stream name", "a stream name2"))
```

`fwa_convert_stream_network_to_rms`*Convert Stream Network to River Meters*

Description

Converts a tibble of a BC stream network to river meters.

Usage

```
fwa_convert_stream_network_to_rms(x, interval = 5, tolerance = 0.1)
```

Arguments

x	An sf tibble of a stream network.
interval	A whole number of the distance between points.
tolerance	A number of the acceptable discrepancy in meters in the network lengths.

Value

An sf tibble with the columns of x plus integer column rm and sf column geometry.

See Also

[fwa_convert_streams_to_rms\(\)](#)

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001, rm = 1000))
network <- fwa_add_collection_to_polygon(watershed)
fwa_convert_stream_network_to_rms(network, interval = 100)

## End(Not run)
```

fwa_convert_streams_to_rms

Convert Streams to River Meters

Description

Converts a tibble of streams to river meters. Unlike [fwa_convert_stream_network_to_rms\(\)](#) it only requires the linestrings and the unique integer identifier for each stream.

Usage

```
fwa_convert_streams_to_rms(
  x,
  interval = 5,
  gap = 1,
  end = NULL,
  elevation = FALSE,
  reverse = integer(),
  source = NULL,
  zoom = 12
)
```

Arguments

x	An sf tibble with a column blk and linestrings of streams.
interval	A positive whole number of the distance (m) between points.
gap	A positive real number specifying the maximum gap (m) between the mouth of stream and its parent stream to be considered connected.
end	A positive whole number indicating how far (m) the end of the stream linestring has to be from the last interval to be included. To default end = NULL (equivalent to end = interval + 1) excludes ends.

elevation	A flag specifying whether to use the elevation to determine stream direction (or use the direction of the provided linestrings)
reverse	A whole numeric vector of streams to reverse direction ignoring elevation.
source	A string of the source of the elevations if elevation = TRUE. Possible values are "elevatr" (no API key required) and "google" (requires a Google Maps Elevation API key). If NULL (the default) and elevation = TRUE "google" is used but this is [Deprecated] and source should be set explicitly.
zoom	A whole number between 1 and 14 of the zoom level of the Terrain Tiles. Higher zoom levels have a finer resolution but take longer to download. Ignored if elevation = FALSE or source = "google".

Value

An sf tibble with the columns blk, integer column rm and sf column point geometry.

See Also

[fwa_convert_stream_network_to_rms\(\)](#)

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001, rm = 1000))
network <- fwa_add_collection_to_polygon(watershed)
network <- select(network, blk = blue_line_key)
fwa_convert_streams_to_rms(network, interval = 100)

## End(Not run)
```

fwa_find_stream_names *Find Stream Names*

Description

Finds gnis stream names that match regular expression.

Usage

```
fwa_find_stream_names(pattern = ".*", ignore_case = TRUE)
```

Arguments

pattern	A string of a regular expression.
ignore_case	A flag specifying whether to ignore case when matching the regular expression to gnis stream names.

Value

A tibble with character column stream_name of the names of all the streams that match the regular expression.

Examples

```
fwa_find_stream_names("sangan")
```

fwa_flip_streams	<i>Flip Streams</i>
------------------	---------------------

Description

Ensures that stream linestrings are oriented in an upstream direction, i.e., that the start (mouth) of each linestring is lower than its end. The elevations of the start and end of each linestring are obtained using [fwa_add_elevatr_elevation_to_point\(\)](#) or [fwa_add_gm_elevation_to_point\(\)](#) and linestrings whose start is more than min_difference m higher than their end are reversed. All other linestrings are not altered.

Usage

```
fwa_flip_streams(x, source = NULL, zoom = 12, min_difference = 5)
```

Arguments

x	An sf tibble with a column blk and linestrings of streams (or stream segments).
source	A string of the source of the elevations. Possible values are "elevatr" (no API key required) and "google" (requires a Google Maps Elevation API key). If NULL (the default) "google" is used but this is [Deprecated] and source should be set explicitly.
zoom	A whole number between 1 and 14 of the zoom level of the Terrain Tiles. Higher zoom levels have a finer resolution but take longer to download. Ignored if source = "google".
min_difference	A non-negative number of the minimum difference (m) between the elevations of the start and end of a linestring for it to be reversed.

Value

A copy of x with the linestrings oriented upstream.

See Also

```
fwa\_add\_elevatr\_elevation\_to\_point\(\), fwa\_add\_gm\_elevation\_to\_point\(\) and fwa\_join\_stream\_segments\(\)
```

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001, rm = 1000))
network <- fwa_add_collection_to_polygon(watershed)
network <- select(network, blk = blue_line_key)
fwa_flip_streams(network)

## End(Not run)
```

`fwa_get_section_from_rms`*Get Section from River Meter*

Description

Gets sf tibble of section blk, rm, length and geometry where sections are defined by the their most upstream point.

Usage

```
fwa_get_section_from_rms(x, section = "section")
```

Arguments

x	A data frame with integer columns blk and rm.
section	A string of the name of the column in x that specifies the sections.

Value

A sf tibble with integer columns blk, rm, length and a geometry where rm and geometry are the upstream end of the section.

`fwa_get_segment_from_rms`*Get Segment from River Meter*

Description

Gets sf tibble of section blk, rm_start, rm_end, segment and line geometry.

Usage

```
fwa_get_segment_from_rms(x, segment = "segment")
```

Arguments

x	A data frame with integer columns blk and rm.
segment	A string of the name of the column in x, which must be a character or factor vector, that specifies which segment each rms belongs to.

Value

A sf tibble with integer columns blk, rm_start, rm_end, character/factor segment and a line geometry.

fwa_join_stream_segments

Join Stream Segments

Description

Converts a tibble of stream segment linestrings to stream linestrings.

Usage

```
fwa_join_stream_segments(
  x,
  elevation = FALSE,
  reverse = integer(),
  source = NULL,
  zoom = 12
)
```

Arguments

x	An sf tibble with a column blk and linestrings of stream segments.
elevation	A flag specifying whether to use the elevation to determine stream direction (or use the direction of the provided linestrings)
reverse	A whole numeric vector of streams to reverse direction ignoring elevation.
source	A string of the source of the elevations if elevation = TRUE. Possible values are "elevatr" (no API key required) and "google" (requires a Google Maps Elevation API key). If NULL (the default) and elevation = TRUE "google" is used but this is [Deprecated] and source should be set explicitly.
zoom	A whole number between 1 and 14 of the zoom level of the Terrain Tiles. Higher zoom levels have a finer resolution but take longer to download. Ignored if elevation = FALSE or source = "google".

Value

An sf tibble with the columns blk and sfc column point geometry.

See Also

[fwa_convert_streams_to_rms\(\)](#)

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001, rm = 1000))
network <- fwa_add_collection_to_polygon(watershed)
network <- select(network, blk = blue_line_key)
fwa_join_stream_segments(network)

## End(Not run)
```

fwa_mapview	<i>Map View</i>
-------------	-----------------

Description

A wrapper on `mapview::mapview.sf()` that allows the user to layer by a column and coerces hms columns to character (to avoid being dropped).

Usage

```
fwa_mapview(x, layer = NULL, zcol = NULL, legend = FALSE, ...)
```

Arguments

- x An sf data frame.
- layer A string of the column to layer the points by.
- zcol A string of the column to color points by.
- legend A flag specifying whether to plot a legend.
- ... Additional arguments passed to `mapview::mapview()`.

fwa_mapview_rms	<i>Map View River Meters</i>
-----------------	------------------------------

Description

Map View River Meters

Usage

```
fwa_mapview_rms(
  x,
  layer = NULL,
  zcol = "rm",
  legend = FALSE,
  npoint = 250,
  ...
)
```

Arguments

x	An sf data frame with unique integer columns blk and rm.
layer	A string of the column to layer the points by.
zcol	A string of the column to color points by.
legend	A flag specifying whether to plot a legend.
npoint	An indication of the total number of points to plot.
...	Additional arguments passed to mapview::mapview().

fwa_mapview_rms_to_rms

Map View River Meters

Description

Maps two alternative stream networks by adding links from each point in x to matching point in y.

Usage

```
fwa_mapview_rms_to_rms(x, y, zcol = "rm", npoint = 250)
```

Arguments

x	An sf data frame with unique integer columns blk and rm and integer column new_rm.
y	An sf data frame with unique integer columns blk and rm.
zcol	A string of the column to color points by.
npoint	An indication of the total number of points to plot.

fwa_mapview_streams *Map View Streams*

Description

Plots stream linestrings in black together with the start of each linestring as a red point labelled "down" and the end as a blue point labelled "up", so that the direction of the streams can be seen. If `half_streams = TRUE` the downstream half (the start) of each linestring is instead colored red and the upstream half (the end) blue. Streams oriented upstream (as produced by [fwa_flip_streams\(\)](#)) have their "down" point at their mouth and "up" point at their source.

Usage

```
fwa_mapview_streams(x, layer = NULL, legend = FALSE, half_streams = FALSE, ...)
```

Arguments

<code>x</code>	An sf data frame with a column <code>blk</code> and linestrings of streams (or stream segments).
<code>layer</code>	A string of the column to layer the points by.
<code>legend</code>	A flag specifying whether to plot a legend.
<code>half_streams</code>	A flag specifying whether to split each linestring into downstream (red) and upstream (blue) halves.
<code>...</code>	Additional arguments passed to <code>mapview::mapview()</code> when plotting the linestrings.

See Also

[fwa_flip_streams\(\)](#)

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001, rm = 1000))
network <- fwa_add_collection_to_polygon(watershed)
network <- select(network, blk = blue_line_key)
fwa_mapview_streams(network)

## End(Not run)
```

fwa_order_segments	<i>Get Stream Segments in Order</i>
--------------------	-------------------------------------

Description

Orders stream segments so they are in order to be stitched together.

Usage

```
fwa_order_segments(x)
```

Arguments

x A spatial data frame.

Details

The function works by locating the longest segment and then finding the nearest segment to it and continues this pattern until all segments have been iterated through.

Value

A spatial data frame.

Examples

```
## Not run:
test_stream <- sf::st_sfc(
  sf::st_multilinestring(
    c(sf::st_linestring(matrix(c(0, 0, 1, 1), ncol = 2, byrow = TRUE)),
      sf::st_linestring(matrix(c(1.05, 1.05, 2, 2), ncol = 2, byrow = TRUE)),
      sf::st_linestring(matrix(c(2.1, 2.1, 3, 3), ncol = 2, byrow = TRUE))
    )
  ),
  crs = 26911
)

df <- data.frame(
  name = "stream 1",
  blk = 1,
  geometry = test_stream
) |>
sf::st_set_geometry("geometry")

fwa_order_segments(df)

## End(Not run)
```

fwa_parent_blk_rms	<i>Parent Blue Line Key</i>
--------------------	-----------------------------

Description

Gets parent blue line key.

Usage

```
fwa_parent_blk_rms(x, rms)
```

Arguments

x	A whole numeric vector of one or more blue line keys.
rms	A data frame with integer columns blk and parent_blk. There must be only one parent_blk for each blk.

Value

A whole numeric vector of the parent blue line keys.

fwa_parent_stream_name_rms	<i>Parent Stream Name</i>
----------------------------	---------------------------

Description

Gets parent stream name.

Usage

```
fwa_parent_stream_name_rms(x, rms, stream_name = fwatlasbc::fwa_stream_name)
```

Arguments

x	A character vector of one or more stream names.
rms	A data frame with integer columns blk and parent_blk. There must be only one parent_blk for each blk.
stream_name	A data frame with character column stream_name and integer column blk. There must be no more than one blk for the stream names specified.

Value

A character vector of the parent blue line keys.

fwa_prune_rms	<i>Prune River Meter</i>
---------------	--------------------------

Description

Removes river meters above rm values in y.

Usage

```
fwa_prune_rms(x, y)
```

Arguments

x	A data frame with integer columns blk, rm, parent_blk and parent_rm.
y	A data frame with integer columns blk and rm.

Value

A tibble of x with rows above the rms in y removed.

fwa_reverse_streams	<i>Reverse Streams</i>
---------------------	------------------------

Description

Reverses the direction of the linestrings of the streams with the specified blue line keys. All other linestrings are not altered.

Usage

```
fwa_reverse_streams(x, blk)
```

Arguments

x	An sf tibble with a column blk and linestrings of streams (or stream segments).
blk	A whole numeric vector of the unique blue line keys of the streams to reverse. All values must be in x\$blk.

Value

A copy of x with the linestrings of the specified streams reversed.

See Also

[fwa_flip_streams\(\)](#) and [fwa_mapview_streams\(\)](#)

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001, rm = 1000))
network <- fwa_add_collection_to_polygon(watershed)
network <- select(network, blk = blue_line_key)
fwa_reverse_streams(network, 356308001)

## End(Not run)
```

fwa_snap_rm_to_point *Snap River Meter to Point*

Description

Assigns closest river meter to each spatial point. If the blue line key (blk) is missing then it is also assigned together with the distance to the river meter (distance_to_rm) in m.

Usage

```
fwa_snap_rm_to_point(x, rm, ...)
```

Arguments

x	An sf object of spatial points with optional integer column blk.
rm	An sf object of spatial point with blk and rm columns.
...	Additional columns to group by when assigning. Elements with missing values in rm are assigned to any value in x.

Value

An updated version of x with integer columns blk and rm and numeric column distance_to_rm.

Examples

```
rm <- fwa_add_rms_to_blk(data.frame(blk = 356308001))
x <- rm[rm$rm %in% c(0, 2000, 5000, 6000, 7000), ]
rm <- rm[rm$rm %in% c(1000, 3000, 4000, 8000, 9000, 10000), ]
fwa_snap_rm_to_point(x, rm)
```

fwa_snap_rm_to_rms	<i>Snap River Meter to River Meters</i>
--------------------	---

Description

Assigns closest river meter to river meters based on blue line keys. If `x` already includes `new_rm` column then non-missing values are preserved. The non-missing `new_rm` values must be ordered (with respect to `x$rm`) and must be present in `rm$rm`. If `x` already includes `new_blk` then river meters can be assigned to a creek with a different blue line key and/or river meters from multiple creeks can be assigned to the same creek (for example in the case of a previously unmapped side channel and the mainstem).

Usage

```
fwa_snap_rm_to_rms(x, rm, snap_mouths = FALSE)
```

Arguments

<code>x</code>	An sf object of spatial points with <code>blk</code> and <code>rm</code> columns and optional <code>new_rm</code> integer and <code>new_blk</code> columns.
<code>rm</code>	An sf object of spatial point with <code>blk</code> and <code>rm</code> columns.
<code>snap_mouths</code>	A flag specifying whether to snap pairs of streams at their mouths (<code>rm = 0</code>) where <code>new_rm</code> is not already set.

Details

The closest river meter is snapped to each `rm` (by `blk = new_blk`) and missing `new_rm` values are replaced with the corresponding `rm` value. The `new_rm` values are then ordered by adjusting the values so that firstly all previous values are not greater than each provided `new_rm` value and then all subsequent values are not less than the maximum previous value. Next all runs of two or more identical `new_rm` values that do not include a provided `new_rm` are interpolated between the previous and subsequent `new_rm` values based on the original `rm` spacing and then snapped to the closest `rm` value in `rm`.

To ensure that pairs of streams snap at their mouths set the `new_rm` to be 0 where the `rm` is 0 or set `snap_mouths = TRUE`

Value

An updated version of `x` with integer columns `blk`, `rm`, `new_blk`, `new_rm` and numeric column `distance_to_new_rm`.

Examples

```
rm <- fwa_add_rms_to_blk(data.frame(blk = 356308001))
x <- rm[rm$rm %in% c(0, 2000, 5000, 6000, 7000), ]
rm <- rm[rm$rm %in% c(1000, 3000, 4000, 8000, 9000, 10000), ]
fwa_snap_rm_to_rms(x, rm)
```

fwa_snap_rms_to_rms	<i>Snap River Meters to River Meters</i>
---------------------	--

Description

Assigns closest river meters to river meters by blue line keys using `fwa_snap_rm_to_rms()` `rm` must not have an existing `new_rm` column.

Usage

```
fwa_snap_rms_to_rms(x, rm, snap_mouths = FALSE)
```

Arguments

<code>x</code>	An sf object of spatial points with <code>blk</code> and <code>rm</code> columns and optional <code>new_blk</code> and <code>new_rm</code> integer column.
<code>rm</code>	An sf object of spatial point with <code>blk</code> and <code>rm</code> columns.
<code>snap_mouths</code>	A flag specifying whether to snap pairs of streams at their mouths (<code>rm = 0</code>) where <code>new_rm</code> is not already set.

Details

`x` is first snapped to `rm` then `rm` is snapped to `x` while ensuring that the links between `x` and `rm` are bidirectional as much as possible.

Value

A named list with an updated versions of `x` and `rm` with integer columns `blk`, `new_blk`, `rm` and `new_rm` and numeric column `distance_to_new_rm`.

See Also

[fwa_snap_rm_to_rms\(\)](#)

Examples

```
rm <- fwa_add_rms_to_blk(data.frame(blk = 356308001))
x <- rm[rm$rm %in% c(0, 2000, 5000, 6000, 7000), ]
rm <- rm[rm$rm %in% c(1000, 3000, 4000, 8000, 9000, 10000), ]
fwa_snap_rms_to_rms(x, rm)
```

fwa_snap_stream_measure_to_point
Snap Stream Measure to Point

Description

Assigns closest stream measure in m to each spatial point. If the blue line key (blk) is missing then it is also assigned together with the distance to the stream (distance_to_stream) in m.

Usage

```
fwa_snap_stream_measure_to_point(x, streams, ...)
```

Arguments

x	An sf object of spatial points with optional integer column blk.
streams	An sf object of spatial linestrings with blk column.
...	Additional columns to group by when assigning.

Value

An updated version of x with integer columns blk and stream_measure and numeric column distance_to_stream.

See Also

[fwa_snap_rm_to_point\(\)](#)

Examples

```
## Not run:
watershed <- fwa_add_watershed_to_blk(data.frame(blk = 356308001, rm = 1000))
network <- fwa_add_collection_to_polygon(watershed)
network$blk <- network$blue_line_key
streams <- fwa_join_stream_segments(network)
points <- fwa_add_rms_to_blk(data.frame(blk = 356308001))
fwa_snap_stream_measure_to_point(points, streams)

## End(Not run)
```

fwa_stitch_segments *Stitch Segments*

Description

Adds segments between the disconnected parts of a MULTILINESTRING.

Usage

```
fwa_stitch_segments(x, ..., tolerance = 5)
```

Arguments

x	A spatial data frame.
...	Unused.
tolerance	A numeric value for the maximum euclidean distance between the end a segment to the start of a segment that will create a stitched segment. If the distance is greater then the tolerance then a segment will not be created.

Details

The fwa_stitch_segments() assumes that the segments within the MULTILINESTRING are in order and the correct direction to be stitched together.

If the segments could not be joined they will be returned as a MULTILINESTRING, if the segments have no gaps left the row will be returned as a LINESTRING.

Value

A data frame with the stitched segments added into the geometries.

Examples

```
## Not run:
test_stream <- sf::st_sfc(
  sf::st_multilinestring(
    c(sf::st_linestring(matrix(c(0, 0, 1, 1), ncol = 2, byrow = TRUE)),
      sf::st_linestring(matrix(c(1.05, 1.05, 2, 2), ncol = 2, byrow = TRUE)),
      sf::st_linestring(matrix(c(2.1, 2.1, 3, 3), ncol = 2, byrow = TRUE))
    )
  ),
  crs = 26911
)

df <- data.frame(
  name = "stream 1",
  blk = 1,
  geometry = test_stream
) |>
```

```
sf::st_set_geometry("geometry")

# to create two 2 segments to connect the whole line
fwa_stitch_segments(df)

# tolerance is set so only 1 segment is added and the line is still not connected
fwa_stitch_segments(df, tolerance = 0.1)

## End(Not run)
```

```
fwa_stream_distances_to_matrix
```

Stream Distances to Matrix

Description

Calculates the distances along the stream network between stream distances (river meters). The distance is Inf if the river meters are on streams that are not connected.

Usage

```
fwa_stream_distances_to_matrix(x, streams, id = NULL)
```

Arguments

x	A data frame (including sf objects) with whole numeric column blk and numeric column rm of the river meters. All blk values must be in streams\$blk and all rm values must be non-negative.
streams	A data frame with whole numeric column blk, whole numeric column parent_blk and numeric column parent_rm. The stream network must not be circular.
id	A string of the name of a column in x with unique non-missing values that are converted to character and used as the row and column names of the matrix or NULL.

Details

The stream network is defined by the parent blue line key (parent_blk) and the river meter on the parent where each stream joins it (parent_rm) as returned by [fwa_add_parent_to_streams\(\)](#). Parents that are not in streams\$blk are treated as root streams.

Value

A numeric matrix of the distances in m with one row and one column for each row in x.

See Also

[fwa_add_parent_to_streams\(\)](#)

Examples

```
streams <- data.frame(
  blk = c(1, 2, 3),
  parent_blk = c(NA, 1, NA),
  parent_rm = c(NA, 400, NA)
)
rms <- data.frame(
  site = c("a", "b", "c", "d"),
  blk = c(1, 1, 2, 3),
  rm = c(100, 500, 50, 10)
)
fwa_stream_distances_to_matrix(rms, streams)
fwa_stream_distances_to_matrix(rms, streams, id = "site")
```

fwa_stream_name	<i>Stream Name</i>
-----------------	--------------------

Description

A tibble of stream names with blue line keys.

Usage

```
fwa_stream_name
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 11606 rows and 2 columns.

Examples

```
fwa_stream_name
```

fwa_swap_branches_rms	<i>Swap Branches of River Meters</i>
-----------------------	--------------------------------------

Description

Swaps two branches of river meters.

Usage

```
fwa_swap_branches_rms(x, y, adjust_points = TRUE)
```

Arguments

- | | |
|----------------------------|---|
| <code>x</code> | A data frame with integer columns <code>blk</code> , <code>rm</code> , <code>parent_blk</code> and <code>parent_rm</code> . |
| <code>y</code> | A data frame with integer column <code>blk</code> specifying the blue line key that currently begins at the confluence of the two branches. |
| <code>adjust_points</code> | A flag specifying whether to adjust the coordinates of points which move. |

Value

A copy of `x` with the branches swapped.

Index

* datasets

- [fwa_collection_name](#), [16](#)
 - [fwa_stream_name](#), [35](#)
- [elevatr::get_elev_point\(\)](#), [7](#)
- [fwa_add_blk_to_lon_lat](#), [3](#)
- [fwa_add_blks_to_stream_name](#), [4](#)
- [fwa_add_collection_to_polygon](#), [4](#)
- [fwa_add_collection_to_watershed](#)
 ([fwa_add_collection_to_polygon](#)),
 [4](#)
- [fwa_add_cut_to_rms](#), [6](#)
- [fwa_add_downstream_split_to_rms](#), [6](#)
- [fwa_add_elevatr_elevation_to_point](#), [7](#)
- [fwa_add_elevatr_elevation_to_point\(\)](#),
 [20](#)
- [fwa_add_gm_elevation_to_point](#), [8](#)
- [fwa_add_gm_elevation_to_point\(\)](#), [7](#), [20](#)
- [fwa_add_intersection_to_geometry](#), [9](#)
- [fwa_add_new_blk_rm_to_blk_rm](#), [9](#)
- [fwa_add_parent_to_streams](#), [10](#)
- [fwa_add_parent_to_streams\(\)](#), [34](#)
- [fwa_add_point_to_stream_measure](#), [11](#)
- [fwa_add_rms_to_blk](#), [12](#)
- [fwa_add_section_to_rms](#), [13](#)
- [fwa_add_split_to_rms](#)
 ([fwa_add_upstream_split_to_rms](#)),
 [14](#)
- [fwa_add_stream_names_to_blk](#), [14](#)
- [fwa_add_upstream_split_to_rms](#), [14](#)
- [fwa_add_watershed_to_blk](#), [15](#)
- [fwa_collection_name](#), [16](#)
- [fwa_convert_rms_to_streams](#), [16](#)
- [fwa_convert_stream_names_to_blks](#), [17](#)
- [fwa_convert_stream_network_to_rms](#), [17](#)
- [fwa_convert_stream_network_to_rms\(\)](#),
 [18](#), [19](#)
- [fwa_convert_streams_to_rms](#), [18](#)
- [fwa_convert_streams_to_rms\(\)](#), [11](#), [18](#), [23](#)
- [fwa_find_stream_names](#), [19](#)
- [fwa_flip_streams](#), [20](#)
- [fwa_flip_streams\(\)](#), [25](#), [28](#)
- [fwa_get_section_from_rms](#), [21](#)
- [fwa_get_segment_from_rms](#), [21](#)
- [fwa_join_stream_segments](#), [22](#)
- [fwa_join_stream_segments\(\)](#), [11](#), [20](#)
- [fwa_mapview](#), [23](#)
- [fwa_mapview_rms](#), [23](#)
- [fwa_mapview_rms_to_rms](#), [24](#)
- [fwa_mapview_streams](#), [25](#)
- [fwa_mapview_streams\(\)](#), [28](#)
- [fwa_order_segments](#), [26](#)
- [fwa_parent_blk_rms](#), [27](#)
- [fwa_parent_stream_name_rms](#), [27](#)
- [fwa_prune_rms](#), [28](#)
- [fwa_reverse_streams](#), [28](#)
- [fwa_reverse_streams\(\)](#), [11](#)
- [fwa_snap_rm_to_point](#), [29](#)
- [fwa_snap_rm_to_point\(\)](#), [32](#)
- [fwa_snap_rm_to_rms](#), [30](#)
- [fwa_snap_rm_to_rms\(\)](#), [31](#)
- [fwa_snap_rms_to_rms](#), [31](#)
- [fwa_snap_stream_measure_to_point](#), [32](#)
- [fwa_snap_stream_measure_to_point\(\)](#), [12](#)
- [fwa_stitch_segments](#), [33](#)
- [fwa_stream_distances_to_matrix](#), [34](#)
- [fwa_stream_name](#), [35](#)
- [fwa_swap_branches_rms](#), [35](#)
- [fwapgr::fwa_collection\(\)](#), [6](#)
- [fwapgr::fwa_watershed_at_measure\(\)](#), [15](#)