

Package ‘bomrang’

December 1, 2020

Type Package

Title Australian Government Bureau of Meteorology ('BOM') Data Client

Version 0.7.3

Description Provides functions to interface with Australian Government Bureau of Meteorology ('BOM') data, fetching data and returning a data frame of precis forecasts, historical and current weather data from stations, agriculture bulletin data, 'BOM' 0900 or 1500 weather bulletins and downloading and importing radar and satellite imagery files. Data (c) Australian Government Bureau of Meteorology Creative Commons (CC) Attribution 3.0 licence or Public Access Licence (PAL) as appropriate. See <http://www.bom.gov.au/other/copyright.shtml> for further details.

URL <https://github.com/ropensci/bomrang>,
<https://docs.ropensci.org/bomrang/>

BugReports <https://github.com/ropensci/bomrang/issues>

License MIT + file LICENSE

Depends R (>= 3.5.0)

Imports curl (>= 2.8.1), crayon, data.table (>= 1.10.4), dplyr (>= 0.7.0), foreign, httr (>= 1.2.1), hoardr, janitor (>= 1.0.0), jsonlite (>= 1.5), lubridate, magick, magrittr (>= 1.5), readr (>= 1.1.1), rvest, terra, tidyr (>= 0.6.3), tools, utils, xml2 (>= 1.1.1)

Encoding UTF-8

LazyData true

Suggests ASGS.foyer, covr, testthat, knitr, rmarkdown, ggplot2, ggthemes, grid, gridExtra, maps, mapproj, rappdirs, roxygen2 (>= 6.1.0)

Language en-AU

RoxygenNote 7.1.1

NeedsCompilation no

ByteCompile TRUE

VignetteBuilder knitr

X-schema.org-applicationCategory Tools

X-schema.org-keywords bom, meteorological-data, weather-forecast,
australia, weather, weather-data, meteorology,
australia-bureau-of-meteorology

X-schema.org-isPartOf <https://ropensci.org>

Author Adam H. Sparks [aut, cre] (<<https://orcid.org/0000-0002-0061-8359>>),
Jonathan Carroll [aut] (<<https://orcid.org/0000-0002-1404-5264>>),
James Goldie [aut] (<<https://orcid.org/0000-0002-5024-6207>>),
Dean Marchiori [aut],
Paul Melloy [aut] (<<https://orcid.org/0000-0003-4253-7167>>),
Mark Padgham [aut, rev] (<<https://orcid.org/0000-0003-2172-5265>>),
Hugh Parsonage [aut] (<<https://orcid.org/0000-0003-4055-0835>>),
Keith Pembleton [aut] (<<https://orcid.org/0000-0002-1896-4516>>),
James Balamuta [ctb] (<<https://orcid.org/0000-0003-2826-8458>>),
Brooke Anderson [rev] (<<https://orcid.org/0000-0002-5012-9035>>)

Maintainer Adam H. Sparks <adamhsparks@gmail.com>

Repository CRAN

Date/Publication 2020-12-01 12:20:09 UTC

R topics documented:

get_ag_bulletin	3
get_available_imagery	4
get_available_radar	6
get_coastal_forecast	7
get_current_weather	8
get_historical_weather	10
get_precis_forecast	12
get_radar_imagery	14
get_satellite_imagery	15
get_weather_bulletin	17
manage_cache	18
parse_ag_bulletin	19
parse_coastal_forecast	21
parse_precis_forecast	22
sweep_for_forecast_towns	24
sweep_for_stations	25
update_forecast_towns	25
update_station_locations	26

Index

28

get_ag_bulletin	<i>Get BOM agriculture bulletin information for select stations</i>
-----------------	---------------------------------------------------------------------

Description

Fetch the BOM agricultural bulletin information and return it in a data frame

Usage

```
get_ag_bulletin(state = "AUS")
```

Arguments

state	Australian state or territory as full name or postal code. Fuzzy string matching via agrep is done. Defaults to "AUS" returning all state bulletins, see Details for more.
-------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Details

Allowed state and territory postal codes, only one state per request or all using AUS.

ACT Australian Capital Territory (will return NSW)

NSW New South Wales

NT Northern Territory

QLD Queensland

SA South Australia

TAS Tasmania

VIC Victoria

WA Western Australia

AUS Australia, returns forecast for all states, NT and ACT

Value

A data frame as a [data.table](#) object of Australia BOM agricultural bulletin information. For full details of fields and units returned see Appendix 3 in the [bomrang](#) vignette, use `vignette("bomrang", package = "bomrang")` to view.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com> and Paul Melloy <paul@melloy.com.au>

References

Agricultural observations are retrieved from the Australian Bureau of Meteorology (BOM) Weather Data Services Agriculture Bulletins,

<http://www.bom.gov.au/catalogue/observations/about-agricultural.shtml>

and

Australian Bureau of Meteorology (BOM) Weather Data Services Observation of Rainfall,

<http://www.bom.gov.au/climate/how/observations/rain-measure.shtml>

Station location and other metadata are sourced from the Australian Bureau of Meteorology (BOM) webpage, Bureau of Meteorology Site Numbers:

<http://www.bom.gov.au/climate/cdo/about/site-num.shtml>

See Also

[parse_ag_bulletin](#)

Examples

```
ag_bulletin <- get_ag_bulletin(state = "QLD")
ag_bulletin
```

`get_available_imagery` *Get a listing of available BOM satellite GeoTIFF imagery*

Description

Fetch a listing of BOM 'GeoTIFF' satellite imagery from <ftp://ftp.bom.gov.au/anon/gen/gms/> to determine which files are currently available for download. Files are available at ten minute update frequency with a 24 hour delete time. Useful to know the most recent files available and then specify in the [get_satellite_imagery](#) function.

Usage

```
get_available_imagery(product_id = "all")
```

Arguments

`product_id` Character. BOM product ID of interest for which a list of available images will be returned. Defaults to all images currently available.

Details

Valid BOM satellite Product IDs for 'GeoTIFF' files include:

- IDE00420** AHI cloud cover only 2km FD GEOS GIS
- IDE00421** AHI IR (Ch13) greyscale 2km FD GEOS GIS
- IDE00422** AHI VIS (Ch3) greyscale 2km FD GEOS GIS
- IDE00423** AHI IR (Ch13) Zehr 2km FD GEOS GIS
- IDE00425** AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km FD GEOS GIS
- IDE00426** AHI VIS (true colour) / IR (Ch13 greyscale) composite 2km FD GEOS GIS
- IDE00427** AHI WV (Ch8) 2km FD GEOS GIS
- IDE00430** AHI cloud cover only 2km AUS equirect. GIS
- IDE00431** AHI IR (Ch13) greyscale 2km AUS equirect. GIS
- IDE00432** AHI VIS (Ch3) greyscale 2km AUS equirect. GIS
- IDE00433** AHI IR (Ch13) Zehr 2km AUS equirect. GIS
- IDE00435** AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km AUS equirect. GIS
- IDE00436** AHI VIS (true colour) / IR (Ch13 greyscale) composite 2km AUS equirect. GIS
- IDE00437** AHI WV (Ch8) 2km AUS equirect. GIS
- IDE00439** AHI VIS (Ch3) greyscale 0.5km AUS equirect. GIS

Value

A vector of all available files for the requested Product ID(s).

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

References

Australian Bureau of Meteorology (BOM) high-definition satellite images <http://www.bom.gov.au/australia/satellite/index.shtml>

Examples

```
# Check availability of AHI VIS (true colour) / IR (Ch13 greyscale) composite
# 1km FD GEOS GIS images
imagery <- get_available_imagery(product_id = "IDE00425")
```

get_available_radar *Get a listing of available BOM radar imagery*

Description

Fetch a listing of available BOM RADAR imagery from <ftp://ftp.bom.gov.au/anon/gen/radar/> to determine which files are currently available for download. The files available are the most recent RADAR imagery for each location, which are updated approximately every 6 to 10 minutes by the BOM.

Usage

```
get_available_radar(radar_id = "all")
```

Arguments

radar_id Numeric. BOM radar ID of interest for which a list of available images will be returned. Defaults to all images currently available.

Details

Valid BOM RADAR ID for each location required.

Value

A data frame of all selected RADAR locations with location information and *product_ids*.

Author(s)

Dean Marchiori, <deanmarchiori@gmail.com>

References

Australian Bureau of Meteorology (BOM) radar image <http://www.bom.gov.au/australia/radar/>

Examples

```
# Check availability radar imagery for Wollongong (radar_id = 3)
imagery <- get_available_radar(radar_id = 3)
```

get_coastal_forecast *Get BOM coastal waters forecast*

Description

Fetch the BOM daily Coastal Waters Forecast and return a data frame of the forecast regions for a specified state or region.

Usage

```
get_coastal_forecast(state = "AUS")
```

Arguments

state Australian state or territory as full name or postal code. Fuzzy string matching via [agrep](#) is done. Defaults to "AUS" returning all state forecasts, see details for further information.

Details

Allowed state and territory postal codes, only one state per request or all using AUS.

ACT Australian Capital Territory (will return NSW)

NSW New South Wales

NT Northern Territory

QLD Queensland

SA South Australia

TAS Tasmania

VIC Victoria

WA Western Australia

AUS Australia, returns forecast for all states, NT and ACT

Value

A [data.table](#) of an Australia BOM Coastal Waters Forecast. For full details of fields and units returned see Appendix 5 in the [bomrang](#) vignette, use `vignette("bomrang", package = "bomrang")` to view.

Author(s)

Dean Marchiori, <deanmarchiori@gmail.com> and Paul Melloy <paul@melloy.com.au>

References

Forecast data come from Australian Bureau of Meteorology (BOM) Weather Data Services

<http://www.bom.gov.au/catalogue/data-feeds.shtml>

Location data and other metadata come from the BOM anonymous FTP server with spatial data

<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/>, specifically the DBF file portion of a shapefile,

<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/IDM00003.dbf>

See Also

[parse_coastal_forecast](#)

Examples

```
coastal_forecast <- get_coastal_forecast(state = "NSW")
coastal_forecast
```

get_current_weather *Get current weather observations of a BOM station*

Description

Get current weather observations of a BOM station

Usage

```
get_current_weather(
  station_name,
  strict = FALSE,
  latlon = NULL,
  emit_latlon_msg = TRUE
)
```

Arguments

station_name	The name of the weather station. Fuzzy string matching via agrep is done.
strict	(logical) If TRUE, <i>station_name</i> must match the station name exactly, except that <i>station_name</i> need not be upper case. Note this may be different to <i>full_name</i> in the response. See Details .
latlon	A length-2 numeric vector giving the decimal degree latitude and longitude (in that order), <i>e.g.</i> , <code>latlon = c(-34, 151)</code> for Sydney. When given instead of <i>station_name</i> , the nearest station (in this package) is used, with a message indicating the nearest such station. (See also sweep_for_stations .) Ignored if used in combination with <i>station_name</i> , with a warning.

emit_latlon_msg

Logical. If TRUE (the default), and latlon is selected, a message is emitted before the table is returned indicating which station was actually used (*i.e.*, which station was found to be nearest to the given coordinate).

Details

Station names are not consistently named within the Bureau, so the response may contain a different full_name to the one matched, even if *strict = TRUE*. For example, `get_current_weather("CASTLEMAINE PRISON")[["full_name"]][1]` is Castlemaine, not Castlemaine Prison.

Note that the column `local_date_time_full` is set to a POSIXct object in the local time of the **user**. For more details see “Appendix 1 - Output from `get_current_weather()`” in the **bomrang** vignette `vignette("bomrang", package = "bomrang")` for a complete list of fields and units.

Value

A `bomrang_tbl` object (extension of a [data.frame](#)) of requested BOM station’s current and prior 72hr data. For full details of fields and units returned, see Appendix 1 in the **bomrang** vignette, use `vignette("bomrang", package = "bomrang")` to view.

Author(s)

Hugh Parsonage, <hugh.parsonage@gmail.com>

References

Weather data observations are retrieved from: Australian Bureau of Meteorology (BOM) Weather Data Services, Observations - individual stations: <http://www.bom.gov.au/catalogue/data-feeds.shtml>

Station location and other metadata are sourced from the Australian Bureau of Meteorology (BOM) webpage, Bureau of Meteorology Site Numbers: <http://www.bom.gov.au/climate/cdo/about/site-num.shtml>

See Also

[get_historical_weather](#)

Examples

```
# warning
Melbourne_weather <- get_current_weather("Melbourne")

# no warning
Melbourne_weather <- get_current_weather("Melbourne (Olympic Park)")

# Get weather by latitude and longitude:
```

```
get_current_weather(latlon = c(-34, 151))
```

```
get_historical_weather
```

Obtain historical BOM data

Description

Retrieves daily observations for a given station.

Usage

```
get_historical_weather(
  stationid = NULL,
  latlon = NULL,
  radius = NULL,
  type = c("rain", "min", "max", "solar")
)
```

```
get_historical(
  stationid = NULL,
  latlon = NULL,
  radius = NULL,
  type = c("rain", "min", "max", "solar")
)
```

Arguments

stationid	BOM station 'ID'. See Details.
latlon	Length-2 numeric vector of Latitude/Longitude. See Details.
radius	Numeric value, distance (km) from <i>latlon</i> , must be numeric.
type	Measurement type for daily, either daily "rain", "min" (temp), "max" (temp), or "solar" (exposure). Partial matching is performed. If not specified returns the first matching type in the order listed. Ignored if hourly or minute are selected for <i>tscale</i> .

Value

A `bomrang_tbl` object (extension of a `data.frame`) of historical observations for the selected station/product type, with some subset of the following columns:

Product_code:	BOM internal code.
Station_number:	BOM station ID.
Year:	Year of observation (YYYY).
Month:	Month of observation (1-12).
Day:	Day of observation (1-31).

Min_temperature:	Minimum daily recorded temperature (degrees C).
Max_temperature:	Maximum daily recorded temperature (degrees C).
Accum_days_min:	Accumulated number of days of minimum temperature.
Accum_days_max:	Accumulated number of days of maximum temperature.
Rainfall:	Daily recorded rainfall in mm.
Period:	Period over which rainfall was measured.
Solar_exposure:	Daily global solar exposure in MJ/m ² .
Quality:	Y, N, or missing. Data which have not yet completed the routine quality control process are marked accordingly.

The following attributes are set on the data, and these are used to generate the header:

site:	BOM station ID.
name:	BOM station name.
lat:	Latitude in decimal degrees.
lon:	Longitude in decimal degrees.
start:	Date observations start.
end:	Date observations end.
years:	Available number of years data.
percent:	Percent complete.
AWS:	Automated weather station?
type:	Measurement types available for the station.

Caution

Temperature data prior to 1910 should be used with extreme caution as many stations prior to that date were exposed in non-standard shelters. Some of which give readings which are several degrees warmer or cooler than those measured according to post-1910 standards.

Daily maximum temperatures usually occur in the afternoon and daily minimum temperatures overnight or near dawn. Occasionally, however, the lowest temperature in the 24 hours to prior to 9 AM can occur around 9 AM the previous day if the night was particularly warm.

Either stationid or latlon must be provided, but if both are, then stationid will be used as it is more reliable.

In some cases data is available back to the 1800s, so tens-of-thousands of daily records will be returned. Other stations will be newer and will return fewer observations.

dplyr Compatibility

The bomrang_tbl class is compatible with [dplyr](#) as long as the bomrang package is on the search path. Common functions ([filter](#), [select](#), [arrange](#), [mutate](#), [rename](#), [arrange](#), [slice](#), [group_by](#)) are provided which mask the **dplyr** versions (but use those internally, maintaining attributes).

Note

Methods `get_historical_weather` and `get_historical` are equivalent. No preference is given to the use of either.

Author(s)

Jonathan Carroll, <rpkg@jcarroll.com.au>

See Also

[get_current_weather](#)

Examples

```
get_historical_weather(stationid = "023000",
                       type = "max") ## ~48,000+ daily records
get_historical_weather(latlon = c(-35.2809, 149.1300),
                       type = "min") ## 3,500+ daily records
```

```
get_historical(stationid = "023000",
               type = "max") ## ~48,000+ daily records
get_historical(latlon = c(-35.2809, 149.1300),
               type = "min") ## 3,500+ daily records
```

get_precis_forecast *Get BOM daily précis forecast for select towns from BOM*

Description

Fetch the BOM daily précis forecast and return a data frame of the seven-day town forecasts for a specified state or territory.

Usage

```
get_precis_forecast(state = "AUS")
```

Arguments

state	Australian state or territory as full name or postal code. Fuzzy string matching via agrep is done. Defaults to "AUS" returning all state bulletins, see Details for more.
-------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Details

Allowed state and territory postal codes, only one state per request or all using AUS.

ACT Australian Capital Territory (will return NSW)

NSW New South Wales

NT Northern Territory

QLD Queensland

SA South Australia

TAS Tasmania

VIC Victoria

WA Western Australia

AUS Australia, returns forecast for all states, NT and ACT

Value

A [data.table](#) of an Australia BOM précis seven day forecasts for BOM selected towns. For full details of fields and units returned see Appendix 2 in the **bomrang** vignette, use `vignette("bomrang", package = "bomrang")` to view.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com> and Keith Pembleton, <keith.pembleton@usq.edu.au> and Paul Melloy, <paul@melloy.com.au>

References

Forecast data come from Australian Bureau of Meteorology (BOM) Weather Data Services
<http://www.bom.gov.au/catalogue/data-feeds.shtml>

Location data and other metadata for towns come from the BOM anonymous FTP server with spatial data

<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/>, specifically the DBF file portion of a shapefile,

<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/IDM00013.dbf>

See Also

[parse_precis_forecast](#)

Examples

```
# get the short forecast for Queensland
BOM_forecast <- get_precis_forecast(state = "QLD")
BOM_forecast
```

get_radar_imagery *Get BOM radar imagery*

Description

Fetch BOM radar imagery from <ftp://ftp.bom.gov.au/anon/gen/radar/> and return a [SpatRaster](#) layer object. Files available are the most recent radar snapshot which are updated approximately every 6 to 10 minutes. Suggested to check file availability first by using [get_available_radar](#).

Usage

```
get_radar_imagery(product_id, path = NULL, download_only = FALSE)
```

Arguments

product_id	Character. BOM product ID to download and import as a magick object. Value is required.
path	Character. A character string with the name where the downloaded file is saved. If not provided, the default value NULL is used which saves the file in an R session temp directory.
download_only	Logical. Whether the radar image is loaded into the environment as a magick object or just downloaded.

Details

Valid BOM RADAR Product IDs for radar imagery can be obtained from [get_available_radar](#).

Value

A **magick** object of the most recent RADAR image snapshot published by the BOM. If `download_only = TRUE` there will be a NULL return value with the download path printed in the console as a message.

Author(s)

Dean Marchiori, <deanmarchiori@gmail.com>

References

Australian Bureau of Meteorology (BOM) radar images
<http://www.bom.gov.au/australia/radar/>

See Also

[get_available_radar](#)

Examples

```
# Fetch most recent radar image for Wollongong 256km radar
imagery <- get_radar_imagery(product_id = "IDR032")
imagery
```

get_satellite_imagery *Get BOM Satellite GeoTIFF Imagery*

Description

Fetch BOM satellite GeoTIFF imagery from <ftp://ftp.bom.gov.au/anon/gen/gms/> and return a raster [SpatRaster](#) object of 'GeoTIFF' files. Files are available at ten minute update frequency with a 24 hour delete time. Suggested to check file availability first by using [get_available_imagery](#).

Usage

```
get_satellite_imagery(product_id, scans = 1, cache = FALSE)
```

Arguments

product_id	Character. BOM product ID to download in 'GeoTIFF' format and import as a SpatRaster object. A vector of values from get_available_imagery may be used here. Value is required.
scans	Numeric. Number of scans to download, starting with most recent and progressing backwards, <i>e.g.</i> , 1 - the most recent single scan available, 6 - the most recent hour available, 12 - the most recent 2 hours available, etc. Negating will return the oldest files first. Defaults to 1. Value is optional.
cache	Logical. Store image files locally for later use? If FALSE, the downloaded files are removed when R session is closed. To take advantage of cached files in future sessions, use cache = TRUE. Defaults to FALSE. Value is optional.

Details

Valid BOM satellite Product IDs for use with *product_id* include:

- IDE00420** AHI cloud cover only 2km FD GEOS GIS
- IDE00421** AHI IR (Ch13) greyscale 2km FD GEOS GIS
- IDE00422** AHI VIS (Ch3) greyscale 2km FD GEOS GIS
- IDE00423** AHI IR (Ch13) Zehr 2km FD GEOS GIS
- IDE00425** AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km FD GEOS GIS
- IDE00426** AHI VIS (true colour) / IR (Ch13 greyscale) composite 2km FD GEOS GIS
- IDE00427** AHI WV (Ch8) 2km FD GEOS GIS

IDE00430 AHI cloud cover only 2km AUS equirect. GIS
IDE00431 AHI IR (Ch13) greyscale 2km AUS equirect. GIS
IDE00432 AHI VIS (Ch3) greyscale 2km AUS equirect. GIS
IDE00433 AHI IR (Ch13) Zehr 2km AUS equirect. GIS
IDE00435 AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km AUS equirect. GIS
IDE00436 AHI VIS (true colour) / IR (Ch13 greyscale) composite 2km AUS equirect. GIS
IDE00437 AHI WV (Ch8) 2km AUS equirect. GIS
IDE00439 AHI VIS (Ch3) greyscale 0.5km AUS equirect. GIS

We cache using **hoardr**, find your cache folder by executing `manage_cache$cache_path_get`.

Value

A `SpatRaster` object of GeoTIFF images with layers named by BOM Product ID, timestamp and band.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

References

Australian Bureau of Meteorology (BOM) high-definition satellite images
<http://www.bom.gov.au/australia/satellite/index.shtml>

See Also

[get_available_imagery](#) [manage_cache](#)

Examples

```
# Fetch AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km FD
# GEOS GIS \code{SpatRaster} object for most recent single scan available

imagery <- get_satellite_imagery(product_id = "IDE00425", scans = 1)

# Get a list of available image files and use that to specify files for
# download, downloading the two most recent files available

avail <- get_available_imagery(product_id = "IDE00425")
imagery <- get_satellite_imagery(product_id = avail, scans = 2)
```

get_weather_bulletin *Get BOM 0900 or 1500 weather bulletin*

Description

Fetch the daily BOM 0900 or 1500 weather bulletins and return a data frame for a specified state or territory.

Usage

```
get_weather_bulletin(state = "qld", morning = TRUE)
```

Arguments

state	Australian state or territory as full name or postal code. Fuzzy string matching via agrep is done.
morning	If TRUE, return the 9am bulletin for the nominated state; otherwise return the 3pm bulletin.

Details

Allowed state and territory postal codes:

ACT Australian Capital Territory (will return NSW)

NSW New South Wales

NT Northern Territory

QLD Queensland

SA South Australia

TAS Tasmania

VIC Victoria

WA Western Australia

It is not possible to return weather bulletins for the entire country in a single call. Rainfall figures for the 9am bulletin are generally for the preceding 24 hours, while those for the 3pm bulletin are for the preceding 6 hours since 9am. Note that values are manually entered into the bulletins and sometimes contain typographical errors which may lead to warnings about "NAs introduced by coercion".

Value

Data frame as a [data.table](#) object of Australian 9am or 3pm weather observations for a state. For full details of fields and units returned see Appendix 4, "Appendix 4 - Output from get_weather_bulletin()" in the **bomrang** vignette, use `vignette("bomrang", package = "bomrang")` to view.

Author(s)

Mark Padgham, <mark.padgham@email.com>

References

Daily observation data come from Australian Bureau of Meteorology (BOM) website. The 3pm bulletin for Queensland is, for example,

http://www.bom.gov.au/qld/observations/3pm_bulletin.shtml

Examples

```
qld_weather <- get_weather_bulletin(state = "QLD", morning = FALSE)
qld_weather
```

manage_cache	<i>Manage locally cached bomrang files</i>
--------------	--------------------------------------------

Description

Manage cached **bomrang** satellite imagery files with **hoardr**.

Details

The default cache directory is `file.path(rappdirs::user_cache_dir(), "R/bomrang")`, but you can set your own path using `manage_cache$cache_path_set()`

`manage_cache$cache_delete` only accepts one file name, while `manage_cache$cache_delete_all` does not accept any names, but deletes all files. For deleting many specific files, use `manage_cache$cache_delete` in an `lapply` type call.

Useful user functions

- `manage_cache$cache_path_get()` - get cache path
- `manage_cache$cache_path_set()` - set cache path
- `manage_cache$list()` - returns a character vector of full path file names
- `manage_cache$files()` - returns file objects with metadata
- `manage_cache$details()` - returns files with details
- `manage_cache$delete()` - delete specific files
- `manage_cache$delete_all()` - delete all files, returns nothing

Examples

```
## Not run:

# list files in cache

imagery <- get_satellite_imagery(product_id = "IDE00425",
                                scans = 1,
                                cache = TRUE)

manage_cache$list()

# delete certain database files
manage_cache$delete("file path")
manage_cache$list()

# delete all files in cache
manage_cache$delete_all()
manage_cache$list()

# set a different cache path from the default
manage_cache$cache_path_set("~/tmp")

## End(Not run)
```

parse_ag_bulletin *Parse local BOM agriculture bulletin XML file(s) for select stations*

Description

Parse local BOM agriculture bulletin XML file(s) and return a data frame for a specified state or territory or all Australia.

Usage

```
parse_ag_bulletin(state, filepath)
```

Arguments

state	Required value of an Australian state or territory as full name or postal code. Fuzzy string matching via agrep is done.
filepath	A string providing the directory location of the précis file(s) to parse. See Details for more.

Details

Allowed state and territory postal codes, only one state per request or all using AUS.

ACT Australian Capital Territory (will return NSW)

NSW New South Wales
NT Northern Territory
QLD Queensland
SA South Australia
TAS Tasmania
VIC Victoria
WA Western Australia
AUS Australia, returns forecast for all states, NT and ACT

The *filepath* argument will only accept a directory where files are located for parsing. DO NOT supply the full path including the file name. This function will only parse the requested state or all of Australia in the same fashion as `get_precis_forecast`, provided that the files are all present in the directory.

Value

A [data.table](#) of Australia BOM agricultural bulletin information. For full details of fields and units returned see Appendix 3 in the **bomrang** vignette, use `vignette("bomrang", package = "bomrang")` to view.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com> and Paul Melloy <paul@melloy.com.au>

References

Agricultural observations are retrieved from the Australian Bureau of Meteorology (BOM) Weather Data Services Agriculture Bulletins,

<http://www.bom.gov.au/catalogue/observations/about-agricultural.shtml>

and

Australian Bureau of Meteorology (BOM) Weather Data Services Observation of Rainfall,

<http://www.bom.gov.au/climate/how/observations/rain-measure.shtml>

Station location and other metadata are sourced from the Australian Bureau of Meteorology (BOM) webpage, Bureau of Meteorology Site Numbers:

<http://www.bom.gov.au/climate/cdo/about/site-num.shtml>

See Also

[get_ag_bulletin](#)

Examples

```
# parse the ag bulletin for Queensland

# download to tempfile() using basename() to keep original name
download.file(url = "ftp://ftp.bom.gov.au/anon/gen/fwo/IDQ60604.xml",
```

```
destfile = file.path(tempdir(),
  basename("ftp://ftp.bom.gov.au/anon/gen/fwo/IDQ60604.xml")),
mode = "wb")

BOM_bulletin <- parse_ag_bulletin(state = "QLD",
  filepath = tempdir())

BOM_bulletin
```

parse_coastal_forecast

Parse local BOM coastal waters forecast XML files

Description

Parse local BOM daily coastal waters forecast XML file(s) and return a data frame for a specified state or territory or all Australia.

Usage

```
parse_coastal_forecast(state = "AUS", filepath)
```

Arguments

state	Required value of an Australian state or territory as full name or postal code. Fuzzy string matching via agrep is done.
filepath	A string providing the directory location of the coastal forecast file(s) to parse. See Details for more.

Details

Allowed state and territory postal codes, only one state per request or all using AUS.

ACT Australian Capital Territory (will return NSW)

NSW New South Wales

NT Northern Territory

QLD Queensland

SA South Australia

TAS Tasmania

VIC Victoria

WA Western Australia

AUS Australia, returns forecast for all states, NT and ACT

The *filepath* argument will only accept a directory where files are located for parsing. DO NOT supply the full path including the file name. This function will only parse the requested state or all of Australia in the same fashion as [get_coastal_forecast](#), provided that the files are all present in the directory.

Value

A [data.table](#) of an Australia BOM Coastal Waters Forecast. For full details of fields and units returned see Appendix 5 in the **bomrang** vignette, use `vignette("bomrang", package = "bomrang")` to view.

Author(s)

Dean Marchiori, <deanmarchiori@gmail.com> and Paul Melloy <paul@melloy.com.au>

References

Forecast data come from Australian Bureau of Meteorology (BOM) Weather Data Services <http://www.bom.gov.au/catalogue/data-feeds.shtml>

Location data and other metadata come from the BOM anonymous FTP server with spatial data <ftp://ftp.bom.gov.au/anon/home/adfd/spatial/>, specifically the DBF file portion of a shapefile, <ftp://ftp.bom.gov.au/anon/home/adfd/spatial/IDM00003.dbf>

See Also

[get_coastal_forecast](#)

Examples

```
# parse the coastal forecast for Queensland

#download to tempfile() using basename() to keep original name
download.file(url = "ftp://ftp.bom.gov.au/anon/gen/fwo/IDQ11290.xml",
             destfile = file.path(tempdir(),
                                 basename("ftp://ftp.bom.gov.au/anon/gen/fwo/IDQ11290.xml")),
             mode = "wb")

coastal_forecast <- parse_coastal_forecast(state = "QLD",
                                         filepath = tempdir())

coastal_forecast
```

`parse_precis_forecast` *Parse local BOM daily précis forecast XML file(s) for select towns*

Description

Parse local BOM daily précis forecast XML file(s) and return a data frame of the seven-day town forecasts for a specified state or territory or all Australia.

Usage

```
parse_precis_forecast(state, filepath)
```

Arguments

state	Required value of an Australian state or territory as full name or postal code. Fuzzy string matching via agrep is done.
filepath	A string providing the directory location of the précis file(s) to parse. See Details for more.

Details

Allowed state and territory postal codes, only one state per request or all using AUS.

ACT Australian Capital Territory (will return NSW)

NSW New South Wales

NT Northern Territory

QLD Queensland

SA South Australia

TAS Tasmania

VIC Victoria

WA Western Australia

AUS Australia, returns forecast for all states, NT and ACT

The *filepath* argument will only accept a directory where files are located for parsing. DO NOT supply the full path including the file name. This function will only parse the requested state or all of Australia in the same fashion as [get_precis_forecast](#), provided that the files are all present in the directory.

Value

A [data.table](#) of Australia BOM précis seven day forecasts for BOM selected towns. For full details of fields and units returned see Appendix 2 in the [bomrang](#) vignette, use `vignette("bomrang", package = "bomrang")` to view.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com> and Keith Pembleton, <keith.pembleton@usq.edu.au> and Paul Melloy, <paul@melloy.com.au>

References

Forecast data come from Australian Bureau of Meteorology (BOM) Weather Data Services
<http://www.bom.gov.au/catalogue/data-feeds.shtml>

Location data and other metadata for towns come from the BOM anonymous FTP server with spatial data
<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/>, specifically the DBF file portion of a shape-file,
<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/IDM00013.dbf>

See Also

[get_precis_forecast](#)

Examples

```
# parse the short forecast for Queensland

# download to tempfile() using basename() to keep original name
download.file(url = "ftp://ftp.bom.gov.au/anon/gen/fwo/IDQ11295.xml",
             destfile = file.path(tempdir(),
                                 basename("ftp://ftp.bom.gov.au/anon/gen/fwo/IDQ11295.xml")),
             mode = "wb")

BOM_forecast <- parse_precis_forecast(state = "QLD",
                                   filepath = tempdir())

BOM_forecast
```

```
sweep_for_forecast_towns
      Find nearest BOM forecast towns
```

Description

Find nearest BOM forecast towns

Usage

```
sweep_for_forecast_towns(latlon = c(-35.3, 149.2))
```

Arguments

`latlon` A length-2 numeric vector. By default, Canberra (approximately).

Value

A [data.table](#) of all forecast towns (in this package) sorted by distance from `latlon`, ascending.

Author(s)

Hugh Parsonage, <hugh.parsonage@gmail.com> and James Goldie, <me@rensa.co>

sweep_for_stations *Find nearest BOM weather stations*

Description

Find nearest BOM weather stations

Usage

```
sweep_for_stations(latlon = c(-35.3, 149.2))
```

Arguments

latlon A length-2 numeric vector. By default, Canberra (approximately).

Value

A [data.table](#) of all weather stations (in this package) sorted by distance from *latlon*, ascending.

Author(s)

Hugh Parsonage, <hugh.parsonage@gmail.com>

update_forecast_towns *Update internal database with latest BOM forecast towns*

Description

Download the latest select forecast towns from the BOM server and update internal database of précis forecast town names and AAC codes used by [get_precis_forecast](#). There is no need to use this unless you know that a forecast town exists in a more current version of the BOM précis forecast town name database that is not available in the database distributed with **bomrang**. In fact, for reproducibility purposes, users are discouraged from using this function.

Usage

```
update_forecast_towns()
```

Value

Updated database of BOM précis forecast towns

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

References

Data are sourced from: Australian Bureau of Meteorology (BOM) webpage, “Weather Data Services”, <http://www.bom.gov.au/catalogue/data-feeds.shtml>

Examples

```
## Not run:  
update_forecast_towns()  
  
## End(Not run)
```

update_station_locations

Update internal databases with latest BOM station metadata

Description

Download the latest station locations and metadata and update internal databases that support the use of [get_current_weather](#), [get_ag_bulletin](#) and [get_historical_weather](#). There is no need to use this unless you know that a station exists in BOM’s database that is not available in the databases distributed with **bomrang**. In fact, for reproducibility purposes, users are discouraged from using this function.

Usage

```
update_station_locations()
```

Details

If **ASGS.foyer** is installed locally, this function will automatically check and correct any invalid state values for stations located in Australia. If **ASGS.foyer** is not installed, the function will update the internal database without validating the state values for stations by reported longitude/latitude location.

Value

Updated internal databases of BOM station locations and JSON URLs

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

References

Station location and other metadata are sourced from the Australian Bureau of Meteorology (BOM) webpage, Bureau of Meteorology Site Numbers: <http://www.bom.gov.au/climate/cdo/about/site-num.shtml>

Examples

```
## Not run:  
update_station_locations()  
  
## End(Not run)
```

Index

agrep, [3](#), [7](#), [8](#), [12](#), [17](#), [19](#), [21](#), [23](#)
arrange, [11](#)

data.frame, [9](#), [10](#)
data.table, [3](#), [7](#), [13](#), [17](#), [20](#), [22–25](#)
dplyr, [11](#)

filter, [11](#)

get_ag_bulletin, [3](#), [20](#), [26](#)
get_available_imagery, [4](#), [15](#), [16](#)
get_available_radar, [6](#), [14](#)
get_coastal_forecast, [7](#), [21](#), [22](#)
get_current_weather, [8](#), [12](#), [26](#)
get_historical
 (get_historical_weather), [10](#)
get_historical_weather, [9](#), [10](#), [26](#)
get_precis_forecast, [12](#), [20](#), [23–25](#)
get_radar_imagery, [14](#)
get_satellite_imagery, [4](#), [15](#)
get_weather_bulletin, [17](#)
group_by, [11](#)

lapply, [18](#)

manage_cache, [16](#), [18](#)
mutate, [11](#)

parse_ag_bulletin, [4](#), [19](#)
parse_coastal_forecast, [8](#), [21](#)
parse_precis_forecast, [13](#), [22](#)

rename, [11](#)

select, [11](#)
slice, [11](#)
SpatRaster, [14](#), [15](#)
sweep_for_forecast_towns, [24](#)
sweep_for_stations, [8](#), [25](#)

update_forecast_towns, [25](#)
update_station_locations, [26](#)