

# Package ‘getlandsat’

August 29, 2016

**Type** Package

**Title** Get Landsat 8 Data from Amazon Public Data Sets

**Description** Get Landsat 8 Data from Amazon Web Services ('AWS') public data sets (<<https://aws.amazon.com/public-data-sets/landsat/>>). Includes functions for listing images and fetching them, and handles caching to prevent unnecessary additional requests.

**Version** 0.1.0

**License** MIT + file LICENSE

**URL** <https://github.com/ropenscilabs/getlandsat>

**BugReports** <https://github.com/ropenscilabs/getlandsat/issues>

**Imports** methods, readr (>= 1.0.0), httr (>= 1.2.0), xml2 (>= 1.0.0), data.table, tibble, rappdirs

**Suggests** roxygen2 (>= 5.0.1), testthat, knitr, covr, raster

**VignetteBuilder** knitr

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Scott Chamberlain [aut, cre]

**Maintainer** Scott Chamberlain <[myrmecocystus@gmail.com](mailto:myrmecocystus@gmail.com)>

**Repository** CRAN

**Date/Publication** 2016-08-16 19:14:01

## R topics documented:

|                    |   |
|--------------------|---|
| getlandsat-package | 2 |
| lsat_cache         | 2 |
| lsat_image         | 3 |
| lsat_list          | 4 |
| lsat_scenes        | 5 |
| lsat_scene_files   | 6 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>8</b> |
|--------------|----------|

---

getlandsat-package      *getlandsat - get Landsat 8 data from AWS public data sets*

---

### Description

**getlandsat** provides access to Landsat <https://landsat.usgs.gov> 8 metadata and images hosted on AWS S3 at <https://aws.amazon.com/public-data-sets/landsat>. The package only fetches data. It does not attempt to aid users in downstream usage.

### Examples

```
## Not run:
## List scenes
(res <- lsat_scenes(n_max = 10))

## List scene files
lsat_scene_files(x = res$download_ur1[1])

## Get an image
### Returns path to the image
lsat_image(x = "LC80101172015002LGN00_B5.TIF")

## Visualize
if (requireNamespace("raster")) {
  library("raster")
  x <- lsat_cache_details()[[1]]
  img <- raster(x$file)
  plot(img)
}

## End(Not run)
```

---

lsat\_cache      *Manage cached files*

---

### Description

Manage cached files

### Usage

```
lsat_cache_list()

lsat_cache_delete(files, force = TRUE)

lsat_cache_delete_all(force = TRUE)

lsat_cache_details(files = NULL)
```

## Arguments

files (character) one or more complete file names  
force (logical) Should files be force deleted? Default: TRUE

## Details

cache\_delete only accepts 1 file name, while cache\_delete\_all doesn't accept any names, but deletes all files. For deleting many specific files, use cache\_delete in a `lapply` type call

We cache using `user_cache_dir`, find your cache folder by executing `rappdirs::user_cache_dir("landsat-pds")`

## Functions

- `lsat_cache_list()` returns a character vector of full path file names
- `lsat_cache_delete()` deletes one or more files, returns nothing
- `lsat_cache_delete_all()` delete all files, returns nothing
- `lsat_cache_details()` prints file name and file size for each file, supply with one or more files, or no files (and get details for all available)

## Examples

```
## Not run:  
# list files in cache  
lsat_cache_list()  
  
# List info for single files  
lsat_cache_details(files = lsat_cache_list()[1])  
lsat_cache_details(files = lsat_cache_list()[2])  
  
# List info for all files  
lsat_cache_details()  
  
# delete files by name in cache  
# lsat_cache_delete(files = lsat_cache_list()[1])  
  
# delete all files in cache  
# lsat_cache_delete_all()  
  
## End(Not run)
```

---

lsat\_image

*GET Landsat image(s)*

---

## Description

GET Landsat image(s)

**Usage**

```
lsat_image(x, overwrite = FALSE, ...)
```

**Arguments**

x (character) A file name for a geotif file, will be more general soon.  
 overwrite (logical) Will only overwrite existing path if TRUE  
 ... Curl args passed on to [GET](#)

**Value**

Path to the file, whether found in cache or new file requested.

**See Also**

[lsat\\_cache](#)

**Examples**

```
## Not run:
# pass an image name
(res <- lsat_list(max = 40))
tifs <- grep("\\.TIF$", res$Key, value = TRUE)
lsat_image(tifs[5])
lsat_image(tifs[6])
lsat_image(tifs[9])

# caching
## requesting an image you already have will return path if found
lsat_image(tifs[5])

## End(Not run)
```

---

lsat\_list

*List Landsat images*

---

**Description**

List Landsat images

**Usage**

```
lsat_list(max = NULL, marker = NULL, prefix = NULL, delimiter = NULL,
  ...)
```

**Arguments**

|           |   |
|-----------|---|
| max       | (integer) number indicating the maximum number of keys to return (max 1000, default 1000).  |
| marker    | (character) string that pecifies the key to start with when listing objects in a AWS bucket. Amazon S3 returns object keys in alphabetical order, starting with key after the marker in order |
| prefix    | (character) string that limits the response to keys that begin with the specified prefix  |
| delimiter | (character) string used to group keys. Read the AWS doc for more detail.  |
| ...       | curl args passed on to <a href="#">GET</a>  |

**Details**

This is an alternative to using [lsat\\_scenes](#). This function uses the AWS S3 API, while [lsat\\_scenes](#) downloads the up to date compressed csv file.

**Examples**

```
## Not run:
lsat_list(max = 10)

# paging, start a specific key string
res <- lsat_list(max = 10)
lsat_list(marker = res$Key[10], max = 10)

# curl options
library("httr")
lsat_list(max = 3, config = verbose())

## End(Not run)
```

---

lsat\_scenes

*List Landsat scenes*


---

**Description**

List Landsat scenes

**Usage**

```
lsat_scenes(...)
```

**Arguments**

... Further args passed on to [read\\_csv](#)

## Details

We use `read_csv` to read the scene list file from [http://landsat-pds.s3.amazonaws.com/scene\\_list.gz](http://landsat-pds.s3.amazonaws.com/scene_list.gz). See the help file for `read_csv` for what parameter you can pass to modify it's behavior.

This is an alternative to using `lsat_list`. This function downloads the up to date compressed csv file, while `lsat_list` uses the AWS S3 API.

## See Also

[lsat\\_scene\\_files](#)

## Examples

```
## Not run:
res <- lsat_scenes()
head(res)

# read only N rows
lsat_scenes(n_max = 10)

## End(Not run)
```

---

|                  |                                       |
|------------------|---------------------------------------|
| lsat_scene_files | <i>List files for a Landsat scene</i> |
|------------------|---------------------------------------|

---

## Description

List files for a Landsat scene

## Usage

```
lsat_scene_files(x, ...)
```

## Arguments

x (character) A URL to a scene html file  
... Curl options passed on to [GET](#)

## Details

This function fetches files available in a scene, while `lsat_scenes` lists the scenes, but not their files

## Value

A data.frame with two columns:

- file - file name
- size - file size

**See Also**

[lsat\\_scenes](#)

**Examples**

```
## Not run:  
res <- lsat_scenes(n_max = 10)  
lsat_scene_files(x = res$download_url[1])  
lsat_scene_files(x = res$download_url[2])  
  
## End(Not run)
```

# Index

## \*Topic **package**

getlandsat-package, 2

GET, 4–6

getlandsat (getlandsat-package), 2

getlandsat-package, 2

lapply, 3

lsat\_cache, 2, 4

lsat\_cache\_delete (lsat\_cache), 2

lsat\_cache\_delete\_all (lsat\_cache), 2

lsat\_cache\_details (lsat\_cache), 2

lsat\_cache\_list (lsat\_cache), 2

lsat\_image, 3

lsat\_list, 4, 6

lsat\_scene\_files, 6, 6

lsat\_scenes, 5, 5, 6, 7

read\_csv, 5, 6

user\_cache\_dir, 3