
espatools Documentation

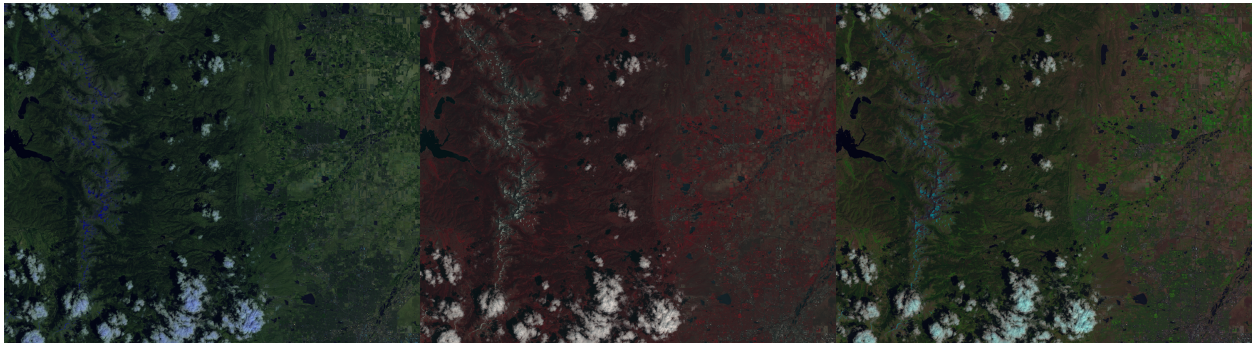
Release v0.0.2

Bane Sullivan

Dec 05, 2020

Contents

1	Connections	3
2	Getting Started	5
2.1	Usage	5
2.2	Example False Color	5
	Python Module Index	17
	Index	19



CHAPTER 1

Connections

- The package heavily uses [properties](#) for the creation of strongly typed objects in a consistent, declarative way.
- This package implements a way to convert these datasets to a [PyVista](#) dataset (`vtkImageData`).
- [PVGeo](#) has implemented an interface for `espatools` to read Landsat imagery via XML metadata files. Check out PVGeo's [Landsat Reader](#) for more details.

CHAPTER 2

Getting Started

`espatools` is available from [PyPI](#)

```
$ pip install espatools
```

2.1 Usage

We think *espatools* is easy to use; give it a try and let us know what you think as this is just the alpha-release!

1. First, checkout [this Jupyter Notebook](#) for a demonstration of some simple plotting after reading Landsat imagery in a Python environment.
2. And take a look at the `.to_pyvista()` method on `RasterSet` objects to have a 3D dataset of the imagery in PyVista/VTK
3. Then take a look at the [Landsat Reader](#) in [PVGeo's](#) documentation where `espatools` has an interface for direct use in ParaView.

2.2 Example False Color

```
import espatools
import matplotlib.pyplot as plt

# Create the reader to manage I/O
reader = espatools.RasterSetReader(filename='metadata.xml')

# Perform the read and yield a raster set
raster = reader.read()

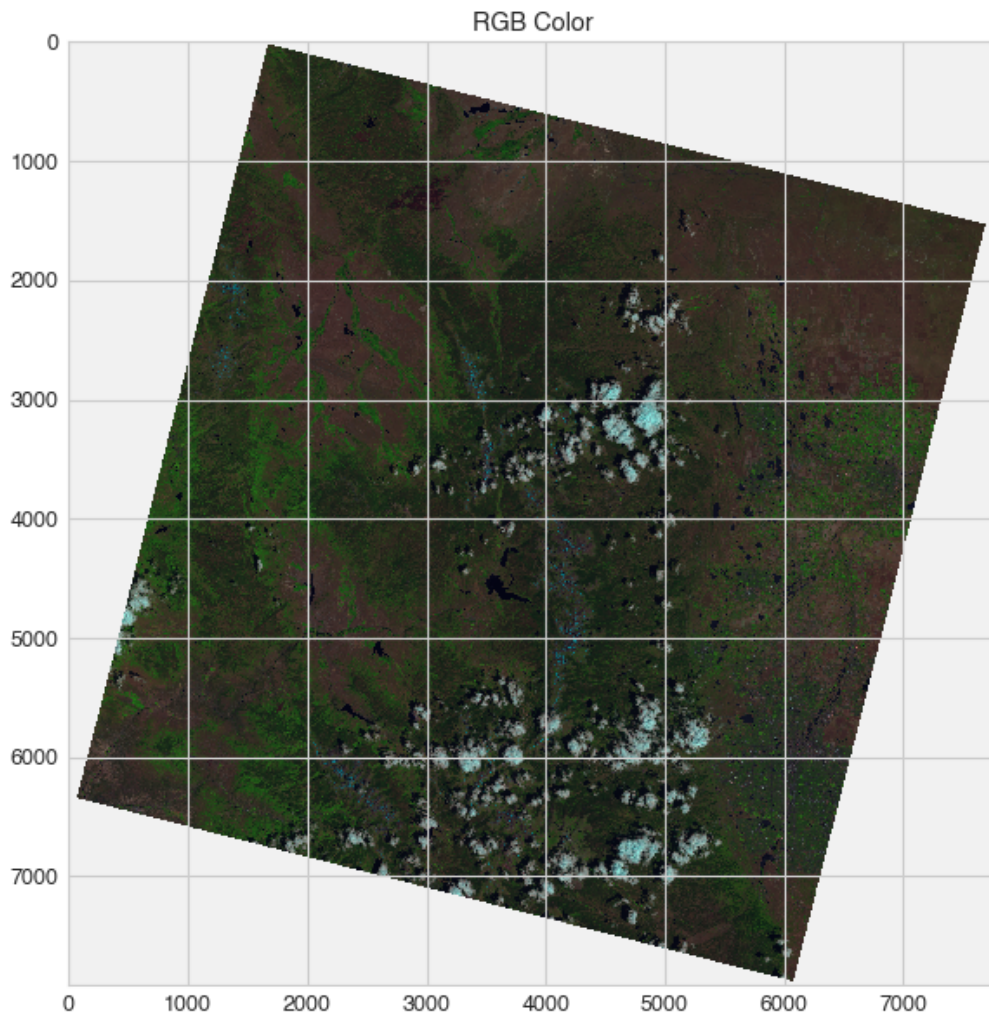
# Get an RGB color scheme
color = raster.get_rgb('false_a')
```

(continues on next page)

(continued from previous page)

```
# Now plot the false color image  
plt.imshow(color)
```

The results of the above code yield the following false color image:



You can also view the dataset in 3D using [PyVista](#):

```
mesh = raster.to_pyvista()  
mesh.plot(scalars='false_a', rgb=True, cpos='xy')
```

2.2.1 About ESPA Tools

- Author: Bane Sullivan
- License: BSD-3-Clause
- Copyright: 2018, Bane Sullivan
- Version: 0.1.1

espatools: An open-source Python package for simple loading of Landsat imagery as NumPy arrays.

meta

This module holds classes that contain the metadata information for a `raster` data object.

BoundingBoxCoordinates

class espatools.meta.BoundingBoxCoordinates (**kwargs)

Bases: `properties.base.base.HasProperties`

Required Properties:

- **east** (`Float`): East line, a float
- **north** (`Float`): North line, a float
- **south** (`Float`): South line, a float
- **west** (`Float`): West line, a float

east

east (`Float`) – East line, a float

north

north (`Float`) – North line, a float

south

south (`Float`) – South line, a float

west

west (`Float`) – West line, a float

Corner

class espatools.meta.Corner (**kwargs)

Bases: `properties.base.base.HasProperties`

Required Properties:

- **latitude** (`Float`): The latitude, a float in range [-90.0, 90.0]
- **location** (`String`): The location, a unicode string
- **longitude** (`Float`): The longitude, a float in range [-180.0, 180.0]

latitude

latitude (`Float`) – The latitude, a float in range [-90.0, 90.0]

location

location (`String`) – The location, a unicode string

longitude

longitude (`Float`) – The longitude, a float in range [-180.0, 180.0]

CornerPoint

class espatools.meta.**CornerPoint** (***kwargs*)
 Bases: `properties.base.base.HasProperties`

Required Properties:

- **location** (`String`): The location, a unicode string
- **x** (`Float`): The X value, a float
- **y** (`Float`): The Y value, a float

location

location (`String`) – The location, a unicode string

x

x (`Float`) – The X value, a float

y

y (`Float`) – The Y value, a float

Lum

class espatools.meta.**Lum** (***kwargs*)
 Bases: `properties.base.base.HasProperties`

Required Properties:

- **bias** (`Float`): The bias, a float
- **gain** (`Float`): The gain, a float

bias

bias (`Float`) – The bias, a float

gain

gain (`Float`) – The gain, a float

PixelSize

class espatools.meta.**PixelSize** (***kwargs*)
 Bases: `properties.base.base.HasProperties`

Required Properties:

- **units** (`String`): The pixel size units, a unicode string
- **x** (`Float`): The X size of the pixel, a float
- **y** (`Float`): The Y size of the pixel, a float

units

units (`String`) – The pixel size units, a unicode string

x

x (`Float`) – The X size of the pixel, a float

y**y** (`Float`) – The Y size of the pixel, a float

Projection

```
class espatools.meta.Projection (**kwargs)
    Bases: properties.base.base.HasProperties
```

Required Properties:

- **corner_point** (a list of `CornerPoint`): The corner points, a list (each item is an instance of `CornerPoint`)
- **datum** (`String`): The projection datum, a unicode string
- **grid_origin** (`String`): The grid origin, a unicode string
- **projection** (`String`): The coordinate projection, a unicode string
- **units** (`String`): The projection units, a unicode string

Optional Properties:

- **albers_proj_params** (`Dictionary`): The Albers projection parameters, a dictionary
- **ps_proj_params** (`Dictionary`): The PS projection parameters, a dictionary
- **sin_proj_params** (`Dictionary`): The Sin projection parameters, a dictionary
- **utm_proj_params** (`Dictionary`): The UTM projection parameters, a dictionary

albers_proj_params

albers_proj_params (`Dictionary`) – The Albers projection parameters, a dictionary

corner_point

corner_point (a list of `CornerPoint`) – The corner points, a list (each item is an instance of `CornerPoint`)

datum

datum (`String`) – The projection datum, a unicode string

grid_origin

grid_origin (`String`) – The grid origin, a unicode string

projection

projection (`String`) – The coordinate projection, a unicode string

ps_proj_params

ps_proj_params (`Dictionary`) – The PS projection parameters, a dictionary

sin_proj_params

sin_proj_params (`Dictionary`) – The Sin projection parameters, a dictionary

units

units (`String`) – The projection units, a unicode string

utm_proj_params

utm_proj_params (`Dictionary`) – The UTM projection parameters, a dictionary

RasterMetaData

```
class espatools.meta.RasterMetaData (**kwargs)
    Bases: properties.base.base.HasProperties
```

An object to contain all the information for a single swath.

Required Properties:

- **bounding_coordinates** (*BoundingCoordinates*): The bounding coordinates, an instance of *BoundingCoordinates*
- **corner** (a list of *Corner*): The corner points, a list (each item is an instance of *Corner*)
- **data_provider** (*String*): The data provider, a unicode string
- **instrument** (*String*): The instrument on the satellite, a unicode string
- **orientation_angle** (*Float*): The orientation angle, a float in range [-360.0, 360.0]
- **projection_information** (*Projection*): The projection, an instance of *Projection*
- **satellite** (*String*): The satellite from which data was aquired, a unicode string

Optional Properties:

- **acquisition_date** (*String*): The date of acquisition, a unicode string
- **earth_sun_distance** (*Float*): The earth-sun distance, a float
- **level1_production_date** (*String*): Production date, a unicode string
- **lpgs_metadata_file** (*String*): metadata file, a unicode string
- **product_id** (*String*): Data product ID, a unicode string
- **scene_center_time** (*String*): Center time, a unicode string
- **solar_angles** (*SolarAngle*): The solar angles, an instance of *SolarAngle*
- **wrs** (*WRS*): WRS, an instance of *WRS*

acquisition_date

acquisition_date (*String*) – The date of acquisition, a unicode string

bounding_coordinates

bounding_coordinates (*BoundingCoordinates*) – The bounding coordinates, an instance of *BoundingCoordinates*

corner

corner (a list of *Corner*) – The corner points, a list (each item is an instance of *Corner*)

data_provider

data_provider (*String*) – The data provider, a unicode string

earth_sun_distance

earth_sun_distance (*Float*) – The earth-sun distance, a float

instrument

instrument (*String*) – The instrument on the satellite, a unicode string

level1_production_date

level1_production_date (*String*) – Production date, a unicode string

lpgs_metadata_file

lpgs_metadata_file (*String*) – metadata file, a unicode string

orientation_angle

orientation_angle (*Float*) – The orientation angle, a float in range [-360.0, 360.0]

product_id

product_id (*String*) – Data product ID, a unicode string

projection_information
projection_information (*Projection*) – The projection, an instance of Projection

satellite
satellite (*String*) – The satellite from which data was aquired, a unicode string

scene_center_time
scene_center_time (*String*) – Center time, a unicode string

solar_angles
solar_angles (*SolarAngle*) – The solar angles, an instance of SolarAngle

wrs
wrs (*WRS*) – WRS, an instance of WRS

SolarAngle

class espatoools.meta.**SolarAngle** (***kwargs*)
 Bases: properties.base.base.HasProperties

Required Properties:

- **azimuth** (*Float*): The azimuth, a float
- **units** (*String*): The units, a unicode string
- **zenith** (*Float*): The zenith, a float

azimuth
azimuth (*Float*) – The azimuth, a float

units
units (*String*) – The units, a unicode string

zenith
zenith (*Float*) – The zenith, a float

ThermalConst

class espatoools.meta.**ThermalConst** (***kwargs*)
 Bases: properties.base.base.HasProperties

Required Properties:

- **k1** (*Float*): K1, a float
- **k2** (*Float*): K2, a float

k1
k1 (*Float*) – K1, a float

k2
k2 (*Float*) – K2, a float

ValidRange

class espatoools.meta.**ValidRange** (***kwargs*)
 Bases: properties.base.base.HasProperties

Required Properties:

- **max** (`Float`): Maximum of valid range, a float
- **min** (`Float`): Minimum of valid range, a float

max

max (`Float`) – Maximum of valid range, a float

min

min (`Float`) – Minimum of valid range, a float

WRS

```
class espatools.meta.WRS (**kwargs)
    Bases: properties.base.base.HasProperties
```

Required Properties:

- **path** (`Integer`): The WRS path, an integer
- **row** (`Integer`): The WRS row, an integer
- **system** (`Integer`): The system type, an integer

path

path (`Integer`) – The WRS path, an integer

row

row (`Integer`) – The WRS row, an integer

system

system (`Integer`) – The system type, an integer

raster

Band

```
class espatools.raster.Band (**kwargs)
    Bases: properties.base.base.HasProperties
```

Contains raster metadata and data for a single band.

Required Properties:

- **data_type** (`String`): Band data type, a unicode string
- **file_name** (`String`): Original file name, a unicode string
- **fill_value** (`Integer`): fill value, an integer, Default: -9999
- **long_name** (`String`): Long display name, a unicode string
- **name** (`String`): Name of the band, a unicode string
- **nlines** (`Integer`): number of lines, an integer
- **nsamps** (`Integer`): number of samples, an integer
- **pixel_size** (`PixelSize`): The pixel size, an instance of `PixelSize`
- **product** (`String`): Data product, a unicode string
- **short_name** (`String`): Short name, a unicode string

Optional Properties:

- **add_offset** (*Float*): Add offset, a float
- **app_version** (*String*): app version, a unicode string
- **bitmap_description** (*Dictionary*): band bitmap description (not always present), a dictionary (keys: a unicode string; values: a unicode string)
- **category** (*String*): Band category, a unicode string
- **data_units** (*String*): Data units, a unicode string
- **percent_coverage** (*Float*): percent coverage, a float
- **production_date** (*String*): production date, a unicode string
- **qa_description** (*String*): QA description, a unicode string
- **radiance** (*Lum*): The radiance, an instance of Lum
- **reflectance** (*Lum*): The reflectance, an instance of Lum
- **resample_method** (*String*): resample method, a unicode string
- **saturate_value** (*Integer*): Saturate value, an integer
- **scale_factor** (*Float*): Scaling factor, a float
- **source** (*String*): Band source, a unicode string
- **thermal_const** (*ThermalConst*): The thermal const, an instance of ThermalConst
- **valid_range** (*ValidRange*): The valid data range, an instance of ValidRange

add_offset

add_offset (*Float*) – Add offset, a float

app_version

app_version (*String*) – app version, a unicode string

bitmap_description

bitmap_description (*Dictionary*) – band bitmap description (not always present), a dictionary (keys – a unicode string; values: a unicode string)

category

category (*String*) – Band category, a unicode string

data = None**data_type**

data_type (*String*) – Band data type, a unicode string

data_units

data_units (*String*) – Data units, a unicode string

file_name

file_name (*String*) – Original file name, a unicode string

fill_value

fill_value (*Integer*) – fill value, an integer, Default – -9999

long_name

long_name (*String*) – Long display name, a unicode string

name

name (*String*) – Name of the band, a unicode string

nlines
nlines (*Integer*) – number of lines, an integer

nsamps
nsamps (*Integer*) – number of samples, an integer

percent_coverage
percent_coverage (*Float*) – percent coverage, a float

pixel_size
pixel_size (*PixelSize*) – The pixel size, an instance of PixelSize

product
product (*String*) – Data product, a unicode string

production_date
production_date (*String*) – production date, a unicode string

qa_description
qa_description (*String*) – QA description, a unicode string

radiance
radiance (*Lum*) – The radiance, an instance of Lum

reflectance
reflectance (*Lum*) – The reflectance, an instance of Lum

resample_method
resample_method (*String*) – resample method, a unicode string

saturate_value
saturate_value (*Integer*) – Saturate value, an integer

scale_factor
scale_factor (*Float*) – Scaling factor, a float

short_name
short_name (*String*) – Short name, a unicode string

source
source (*String*) – Band source, a unicode string

thermal_const
thermal_const (*ThermalConst*) – The thermal const, an instance of ThermalConst

valid_range
valid_range (*ValidRange*) – The valid data range, an instance of ValidRange

ColorSchemes

class espatools.raster.ColorSchemes

Bases: *object*

A class to hold various RGB color schemes for reference. These color schemes are defined on the [USGS website](#).

```
LOOKUP_FALSE_COLOR_A = {'LANDSAT_4': ['sr_band5', 'sr_band4', 'sr_band3'], 'LANDSAT_5': ['sr_band5', 'sr_band4', 'sr_band3']}
LOOKUP_FALSE_COLOR_B = {'LANDSAT_4': ['sr_band7', 'sr_band5', 'sr_band3'], 'LANDSAT_5': ['sr_band7', 'sr_band5', 'sr_band3']}
LOOKUP_FALSE_COLOR_C = {'LANDSAT_4': ['sr_band7', 'sr_band4', 'sr_band2'], 'LANDSAT_5': ['sr_band7', 'sr_band4', 'sr_band2']}
LOOKUP_INFRARED = {'LANDSAT_4': ['sr_band4', 'sr_band3', 'sr_band2'], 'LANDSAT_5': ['sr_band4', 'sr_band3', 'sr_band2']}
```

```
LOOKUP_TRUE_COLOR = {'LANDSAT_4': ['sr_band3', 'sr_band2', 'sr_band1'], 'LANDSAT_5':
```

RasterSet

```
class espatools.raster.RasterSet (**kwargs)
    Bases: properties.base.base.HasProperties
```

The main class to hold a set of raster data. This contains all of the bands for a given set of rasters. This is generated by the RasterSetReader.

Required Properties:

- **bands** (*Dictionary*): A dictionary of bands for the swath, a dictionary (keys: a unicode string; values: an instance of Band)
- **global_metadata** (*RasterMetaData*): Raster metadata, an instance of RasterMetaData
- **nlines** (*Integer*): The number of lines, an integer
- **nsamps** (*Integer*): The number of samples, an integer
- **pixel_size** (*PixelSize*): The pixel size, an instance of PixelSize

Optional Properties:

- **version** (*String*): version, a unicode string

```
GetRGB (*args, **kwargs)
```

```
RGB_SCHEMES = {'false_a': {'LANDSAT_8': ['sr_band6', 'sr_band5', 'sr_band4'], 'LANDSAT_5':
```

bands

bands (*Dictionary*) – A dictionary of bands for the swath, a dictionary (keys – a unicode string; values: an instance of Band)

```
get_rgb (scheme='infrared', names=None)
```

Get an RGB color scheme based on predefined presets or specify your own band names to use. A given set of names always overrides a scheme.

Note: Available schemes are defined in RGB_SCHEMES and include:

- true
 - infrared
 - false_a
 - false_b
 - false_c
-

global_metadata

global_metadata (*RasterMetaData*) – Raster metadata, an instance of RasterMetaData

nlines

nlines (*Integer*) – The number of lines, an integer

nsamps

nsamps (*Integer*) – The number of samples, an integer

pixel_size

pixel_size (*PixelSize*) – The pixel size, an instance of PixelSize

to_pyvista (*z=0.0*)

Create a `pyvista.UniformGrid` of this raster. Use the `z` argument to control the dataset's Z spatial reference.

validate ()

version

version (*String*) – version, a unicode string

read

This module holds the file I/O methods for rasters and bands.

RasterSetReader

class `espatools.read.RasterSetReader` (***kwargs*)

Bases: `object`

Read a series of raster files via their XML metadata file in ESPA schema

Read (**args, **kwargs*)

SetFileName (*filename*)

static clean_dict (*d*)

generate_band (*band, meta_only=False, cast=False*)

Genreate a Band object given band metadata

Parameters *band* (*dict*) – dictionary containing metadata for a given band

Returns the loaded Band onject

Return type *Band*

read (*meta_only=False, allowed=None, cast=False*)

Read the ESPA XML metadata file

static read_tif (*tifFile*)

Reads a tif file to a 2D NumPy array

set_file_name (*filename*)

set_properties

`espatools.read.set_properties` (*has_props_cls, input_dict, include_immutable=True*)

A helper method to set an `HasProperties` object's properties from a dictionary

e

`espatools.meta`, [7](#)
`espatools.raster`, [12](#)
`espatools.read`, [16](#)

A

acquisition_date (espatools.meta.RasterMetaData attribute), 10
add_offset (espatools.raster.Band attribute), 13
albers_proj_params (espatools.meta.Projection attribute), 9
app_version (espatools.raster.Band attribute), 13
azimuth (espatools.meta.SolarAngle attribute), 11

B

Band (class in espatools.raster), 12
bands (espatools.raster.RasterSet attribute), 15
bias (espatools.meta.Lum attribute), 8
bitmap_description (espatools.raster.Band attribute), 13
bounding_coordinates (espatools.meta.RasterMetaData attribute), 10
BoundingCoordinates (class in espatools.meta), 7

C

category (espatools.raster.Band attribute), 13
clean_dict() (espatools.read.RasterSetReader static method), 16
ColorSchemes (class in espatools.raster), 14
Corner (class in espatools.meta), 7
corner (espatools.meta.RasterMetaData attribute), 10
corner_point (espatools.meta.Projection attribute), 9
CornerPoint (class in espatools.meta), 8

D

data (espatools.raster.Band attribute), 13
data_provider (espatools.meta.RasterMetaData attribute), 10
data_type (espatools.raster.Band attribute), 13
data_units (espatools.raster.Band attribute), 13
datum (espatools.meta.Projection attribute), 9

E

earth_sun_distance (espatools.meta.RasterMetaData attribute), 10

east (espatools.meta.BoundingCoordinates attribute), 7
espatools.meta (module), 7
espatools.raster (module), 12
espatools.read (module), 16

F

file_name (espatools.raster.Band attribute), 13
fill_value (espatools.raster.Band attribute), 13

G

gain (espatools.meta.Lum attribute), 8
generate_band() (espatools.read.RasterSetReader method), 16
get_rgb() (espatools.raster.RasterSet method), 15
GetRGB() (espatools.raster.RasterSet method), 15
global_metadata (espatools.raster.RasterSet attribute), 15
grid_origin (espatools.meta.Projection attribute), 9

I

instrument (espatools.meta.RasterMetaData attribute), 10

K

k1 (espatools.meta.ThermalConst attribute), 11
k2 (espatools.meta.ThermalConst attribute), 11

L

latitude (espatools.meta.Corner attribute), 7
level1_production_date (espatools.meta.RasterMetaData attribute), 10
location (espatools.meta.Corner attribute), 7
location (espatools.meta.CornerPoint attribute), 8
long_name (espatools.raster.Band attribute), 13
longitude (espatools.meta.Corner attribute), 8
LOOKUP_FALSE_COLOR_A (espatools.raster.ColorSchemes attribute), 14
LOOKUP_FALSE_COLOR_B (espatools.raster.ColorSchemes attribute), 14
LOOKUP_FALSE_COLOR_C (espatools.raster.ColorSchemes attribute), 14

LOOKUP_INFRARED (espatools.raster.ColorSchemes attribute), 14
 LOOKUP_TRUE_COLOR (espatools.raster.ColorSchemes attribute), 14
 lpgs_metadata_file (espatools.meta.RasterMetaData attribute), 10
 Lum (class in espatools.meta), 8

M

max (espatools.meta.ValidRange attribute), 12
 min (espatools.meta.ValidRange attribute), 12

N

name (espatools.raster.Band attribute), 13
 nlines (espatools.raster.Band attribute), 13
 nlines (espatools.raster.RasterSet attribute), 15
 north (espatools.meta.BoundingCoordinates attribute), 7
 nsamps (espatools.raster.Band attribute), 14
 nsamps (espatools.raster.RasterSet attribute), 15

O

orientation_angle (espatools.meta.RasterMetaData attribute), 10

P

path (espatools.meta.WRS attribute), 12
 percent_coverage (espatools.raster.Band attribute), 14
 pixel_size (espatools.raster.Band attribute), 14
 pixel_size (espatools.raster.RasterSet attribute), 15
 PixelSize (class in espatools.meta), 8
 product (espatools.raster.Band attribute), 14
 product_id (espatools.meta.RasterMetaData attribute), 10
 production_date (espatools.raster.Band attribute), 14
 Projection (class in espatools.meta), 9
 projection (espatools.meta.Projection attribute), 9
 projection_information (espatools.meta.RasterMetaData attribute), 10
 ps_proj_params (espatools.meta.Projection attribute), 9

Q

qa_description (espatools.raster.Band attribute), 14

R

radiance (espatools.raster.Band attribute), 14
 RasterMetaData (class in espatools.meta), 9
 RasterSet (class in espatools.raster), 15
 RasterSetReader (class in espatools.read), 16
 Read() (espatools.read.RasterSetReader method), 16
 read() (espatools.read.RasterSetReader method), 16
 read_tif() (espatools.read.RasterSetReader static method), 16
 reflectance (espatools.raster.Band attribute), 14
 resample_method (espatools.raster.Band attribute), 14

RGB_SCHEMES (espatools.raster.RasterSet attribute), 15
 row (espatools.meta.WRS attribute), 12

S

satellite (espatools.meta.RasterMetaData attribute), 11
 saturate_value (espatools.raster.Band attribute), 14
 scale_factor (espatools.raster.Band attribute), 14
 scene_center_time (espatools.meta.RasterMetaData attribute), 11
 set_file_name() (espatools.read.RasterSetReader method), 16
 set_properties() (in module espatools.read), 16
 SetFileName() (espatools.read.RasterSetReader method), 16
 short_name (espatools.raster.Band attribute), 14
 sin_proj_params (espatools.meta.Projection attribute), 9
 solar_angles (espatools.meta.RasterMetaData attribute), 11
 SolarAngle (class in espatools.meta), 11
 source (espatools.raster.Band attribute), 14
 south (espatools.meta.BoundingCoordinates attribute), 7
 system (espatools.meta.WRS attribute), 12

T

thermal_const (espatools.raster.Band attribute), 14
 ThermalConst (class in espatools.meta), 11
 to_pyvista() (espatools.raster.RasterSet method), 15

U

units (espatools.meta.PixelSize attribute), 8
 units (espatools.meta.Projection attribute), 9
 units (espatools.meta.SolarAngle attribute), 11
 utm_proj_params (espatools.meta.Projection attribute), 9

V

valid_range (espatools.raster.Band attribute), 14
 validate() (espatools.raster.RasterSet method), 16
 ValidRange (class in espatools.meta), 11
 version (espatools.raster.RasterSet attribute), 16

W

west (espatools.meta.BoundingCoordinates attribute), 7
 WRS (class in espatools.meta), 12
 wrs (espatools.meta.RasterMetaData attribute), 11

X

x (espatools.meta.CornerPoint attribute), 8
 x (espatools.meta.PixelSize attribute), 8

Y

y (espatools.meta.CornerPoint attribute), 8
 y (espatools.meta.PixelSize attribute), 8

Z

zenith (espatools.meta.SolarAngle attribute), [11](#)