Scikit Data Access

Generated by Doxygen 1.8.13

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 skdaccess Namespace Reference

Namespaces

- astro
- framework
- geo
- planetary
- solar
- utilities

5.2 skdaccess.astro Namespace Reference

Namespaces

- kepler
- voyager

5.3 skdaccess.astro.kepler Namespace Reference

Namespaces

- data_fetcher

5.4 skdaccess.astro.kepler.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for Kepler light curve data.

5.5 skdaccess.astro.voyager Namespace Reference

Namespaces

- data_fetcher

5.6 skdaccess.astro.voyager.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for Mahali temperature data.

5.7 skdaccess.framework Namespace Reference

Namespaces

- data_class
- param_class

5.8 skdaccess.framework.data_class Namespace Reference

Classes

- class DataFetcherBase

  Base class for all data fetchers.
- class DataFetcherCache

  Data fetcher base class for downloading data and caching results on hard disk.
- class DataFetcherLocal

  Data fetcher base class for use when storing data locally.
- class DataFetcherStorage

  Data fetcher base class for use when entire data set is downloaded.
- class DataFetcherStream

  Data fetcher base class for downloading data into memory.
- class DataWrapperBase

  Base class for wrapping data for use in DiscoveryPipeline.
- class ImageWrapper

  Wrapper for image data.
- class SeriesDictionaryWrapper

  Data wrapper for series data using a dictionary of data frames.
- class SeriesWrapper

  Data wrapper for series data using a data panel.
- class TableWrapper

  Data wrapper for table data using an ordered dictionary.
- class XArrayWrapper

  Wrapper for xarrays.

5.9 skdaccess.framework.param_class Namespace Reference

Classes

- class AutoList

  Specifies a list for returning selections of lists, as opposed to a single element.
- class AutoListCycle

  An Autolist that cycles through different lists.
- class AutoListPermute

  A perturber that permutes a list.
- class AutoListRemove

  Removes a different single element from the initial list at each perturb call.
- class AutoListSubset

  An AutoList perturber that creates random subsets of a list.
- class AutoParam

  Defines a tunable parameter class inherited by specific subclasses.
- class AutoParamList

  A tunable parameter with a specified list of choices that can be randomly selected via perturb.
- class AutoParamListCycle

  Cycles through a list of paramters.
- class AutoParamMinMax

  A tunable parameter with min and max ranges, perturbs to a random value in range.

5.10 skdaccess.geo Namespace Reference

Namespaces

- era_interim
- gldas
- grace
- groundwater
- imsdnhs
- magnetometer
- mahali
- modis
- ngl_gps
- pbo
- sentinel_1
- srtm
- uavsar
- wyoming_sounding

5.11 skdaccess.geo.era_interim Namespace Reference

Namespaces

- cache

5.12 skdaccess.geo.era_interim.cache Namespace Reference

Namespaces

- data_fetcher

5.13 skdaccess.geo.era_interim.cache.data_fetcher Namespace Reference

Classes

- class DataFetcher

  DataFetcher for retrieving ERA-I data.

5.14 skdaccess.geo.gldas Namespace Reference

Namespaces

- data_fetcher

5.15 skdaccess.geo.gldas.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for GLDAS data.

5.16 skdaccess.geo.grace Namespace Reference

Namespaces

- data_fetcher
- mascon

5.17 skdaccess.geo.grace.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for GRACE data.

5.18 skdaccess.geo.grace.mascon Namespace Reference

Namespaces

- cache

5.19 skdaccess.geo.grace.mascon.cache Namespace Reference

Namespaces

- data_fetcher

5.20 skdaccess.geo.grace.mascon.cache.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for GRACE mascon data.

5.21 skdaccess.geo.groundwater Namespace Reference

Namespaces

- data_fetcher

5.22 skdaccess.geo.groundwater.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Generates Data Wrappers of groundwater measurements taken in the US.

5.23 skdaccess.geo.imsdnhs Namespace Reference

Namespaces

- data_fetcher

5.24 skdaccess.geo.imsdnhs.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Fetches data for the Interactive Multisensor Snow and Ice Mapping System Daily Northern Hemisphere Snow and Ice Analysis.

5.25 skdaccess.geo.magnetometer Namespace Reference

Namespaces

- data_fetcher

5.26 skdaccess.geo.magnetometer.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data fetcher for USGS geomagnetic observatories.

5.27 skdaccess.geo.mahali Namespace Reference

Namespaces

- rinex
- tec
- temperature

5.28 skdaccess.geo.mahali.rinex Namespace Reference

Namespaces

- data_fetcher
- data_wrapper

5.29 skdaccess.geo.mahali.rinex.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for Mahali Data.

5.30 skdaccess.geo.mahali.rinex.data_wrapper Namespace Reference

Classes

- class DataWrapper

  Data wrapper for Mahali data.

5.31 skdaccess.geo.mahali.tec Namespace Reference

Namespaces

- data_fetcher

5.32 skdaccess.geo.mahali.tec.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for Mahali Data.

5.33 skdaccess.geo.mahali.temperature Namespace Reference

Namespaces

- data_fetcher

5.34 skdaccess.geo.mahali.temperature.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for Mahali temperature data.

5.35 skdaccess.geo.modis Namespace Reference

Namespaces

- cache
- stream

5.36 skdaccess.geo.modis.cache Namespace Reference

Namespaces

- cloud_mask
- cloud_opacity
- data_fetcher
- reflectance

5.37 skdaccess.geo.modis.cache.cloud_mask Namespace Reference

Namespaces

- data_fetcher

5.38 skdaccess.geo.modis.cache.cloud_mask.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for MODIS Cloud Mask.

5.39 skdaccess.geo.modis.cache.cloud_opacity Namespace Reference

Namespaces

- data_fetcher

5.40 skdaccess.geo.modis.cache.cloud_opacity.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for MODIS Cloud Opacity.

5.41 skdaccess.geo.modis.cache.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for MODIS data.

5.42 skdaccess.geo.modis.cache.reflectance Namespace Reference

Namespaces

- data_fetcher

5.43 skdaccess.geo.modis.cache.reflectance.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data fetcher for the modis surface reflectance product ('09', 1 km resolution)

5.44 skdaccess.geo.modis.stream Namespace Reference

Namespaces

- cloud_mask
- cloud_opacity
- data_fetcher
- reflectance

5.45 skdaccess.geo.modis.stream.cloud_mask Namespace Reference

Namespaces

- data_fetcher

5.46 skdaccess.geo.modis.stream.cloud_mask.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for MODIS Cloud Mask.

5.47 skdaccess.geo.modis.stream.cloud_opacity Namespace Reference

Namespaces

- data_fetcher

5.48 skdaccess.geo.modis.stream.cloud_opacity.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for MODIS Cloud Opacity.

5.49 skdaccess.geo.modis.stream.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for MODIS data.

5.50 skdaccess.geo.modis.stream.reflectance Namespace Reference

Namespaces

- data_fetcher

5.51 skdaccess.geo.modis.stream.reflectance.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data fetcher for the modis surface reflectance product ('09', 1 km resolution)

5.52 skdaccess.geo.ngl_gps Namespace Reference

Namespaces

- data_fetcher

5.53 skdaccess.geo.ngl_gps.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data fetcher for GPS data from Neveda Geodetic Laboratory.

5.54 skdaccess.geo.pbo Namespace Reference

Namespaces

- data_fetcher

5.55 skdaccess.geo.pbo.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data fetcher for PBO GPS data.

5.56 skdaccess.geo.sentinel_1 Namespace Reference

Namespaces

- cache

5.57 skdaccess.geo.sentinel_1.cache Namespace Reference

Namespaces

- data_fetcher

5.58 skdaccess.geo.sentinel_1.cache.data_fetcher Namespace Reference

Classes

- class DataFetcher

  DataFetcher for retrieving Sentinel SLC data.

5.59 skdaccess.geo.srtm Namespace Reference

Namespaces

- cache

5.60 skdaccess.geo.srtm.cache Namespace Reference

Namespaces

- data_fetcher

5.61 skdaccess.geo.srtm.cache.data_fetcher Namespace Reference

Classes

- class DataFetcher

  DataFetcher for retrieving data from the Shuttle Radar Topography Mission.

5.62 skdaccess.geo.uavsar Namespace Reference

Namespaces

- cache

5.63 skdaccess.geo.uavsar.cache Namespace Reference

Namespaces

- data_fetcher

5.64 skdaccess.geo.uavsar.cache.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for UAVSAR data.

5.65 skdaccess.geo.wyoming_sounding Namespace Reference

Namespaces

- cache
- stream

5.66 skdaccess.geo.wyoming_sounding.cache Namespace Reference

Namespaces

- data_fetcher

5.67 skdaccess.geo.wyoming_sounding.cache.data_fetcher Namespace Reference

Classes

- class DataFetcher

  DataFetcher for retrieving Wyoming Sounding data.

5.68 skdaccess.geo.wyoming_sounding.stream Namespace Reference

Namespaces

- data_fetcher

5.69 skdaccess.geo.wyoming_sounding.stream.data_fetcher Namespace Reference

Classes

- class DataFetcher

  DataFetcher for retrieving Wyoming Sounding data.

5.70 skdaccess.planetary Namespace Reference

Namespaces

- ode

5.71 skdaccess.planetary.ode Namespace Reference

Namespaces

- cache

5.72 skdaccess.planetary.ode.cache Namespace Reference

Namespaces

- data_fetcher

5.73 skdaccess.planetary.ode.cache.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher from the Orbital Data Explorer (ODE)

5.74 skdaccess.solar Namespace Reference

Namespaces

- sdo

5.75 skdaccess.solar.sdo Namespace Reference

Namespaces

- data_fetcher

5.76 skdaccess.solar.sdo.data_fetcher Namespace Reference

Classes

- class DataFetcher

  Data Fetcher for Mahali temperature data.

5.77 skdaccess.utilities Namespace Reference

Namespaces

- file_browser
- grace_util
- gw_util
- image_util
- kepler_util
- mahali_util
- modis_util
- ode_util
- pbo_util
- sentinel_1_util
- sounding_util
- srtm_util
- support
- uavsar_util

5.78 skdaccess.utilities.file_browser Namespace Reference

Classes

- class FileBrowser

5.79 skdaccess.utilities.grace_util Namespace Reference

Functions

- def averageDates (dates, round_nearest_day=False)

  Compute the average of a pandas series of timestamps.
- def dateMismatch (dates, days=10)

  Check if dates are not within a certain number of days of each other.
- def computeEWD (grace_data, scale_factor, round_nearest_day=False)

  Compute scale corrected equivalent water depth.
- def readTellusData (filename, lat_lon_list, lat_name, lon_name, data_name, data_label=None, time_name=None, lat_bounds_name=None, lon_bounds_name=None, uncertainty_name=None, lat_bounds=None, lon_bounds=None)

  This function reads in netcdf data provided by GRACE Tellus.
- def getStartEndDate (in_data)

5.79.1 Function Documentation

5.79.1.1 averageDates()

```
def skdaccess.utilities.grace_util.averageDates (
              dates,
              round_nearest_day = False )
```

Compute the average of a pandas series of timestamps.

Parameters

dates	Pandas series of pandas datetime objects
round_nearest_day	Round to the nearest day

Returns

Average of dates

5.79.1.2 computeEWD()

```
def skdaccess.utilities.grace_util.computeEWD (
              grace_data,
              scale_factor,
              round_nearest_day = False )
```

Compute scale corrected equivalent water depth.

Equivalent water depth by averaging results from GFZ, CSR, and JPL, and then applying the scale factor

Parameters

grace_data	Data frame containing grace data
scale_factor	Scale factor to apply
round_nearest_day	Round dates to nearest day

Returns

Equivalent water depth determined by applying the scale factor to the average GFZ, JPL and CSR.

5.79.1.3 dateMismatch()

```
def skdaccess.utilities.grace_util.dateMismatch (
              dates,
              days = 10 )
```

Check if dates are not within a certain number of days of each other.

Parameters

dates	Iterable container of pandas timestamps
days	Number of days

Returns

true if they are not with 10 days, false otherwise

5.79.1.4 getStartEndDate()

```
def skdaccess.utilities.grace_util.getStartEndDate (
              in_data )
```

5.79.1.5 readTellusData()

```
def skdaccess.utilities.grace_util.readTellusData (
              filename,
              lat_lon_list,
              lat_name,
              lon_name,
              data_name,
              data_label = None,
              time_name = None,
              lat_bounds_name = None,
              lon_bounds_name = None,
              uncertainty_name = None,
              lat_bounds = None,
              lon_bounds = None )
```

This function reads in netcdf data provided by GRACE Tellus.

Parameters

filename	Name of file to read in
lat_lon_list	List of latitude, longitude tuples that are to be read
data_label	Label for data
lat_name	Name of latitude data
lon_name	Name of longitude data
data_name	Name of data product
time_name	Name of time data
lat_bounds_name	Name of latitude boundaries
lon_bounds_name	Name of longitude boundaries
uncertainty_name	Name of uncertainty in data set
lat_bounds	Latitude bounds
lon_bounds	Longitude bounds

Returns

dictionary containing data and dictionary containing latitude and longitude

5.80 skdaccess.utilities.gw_util Namespace Reference

Functions

- def combine_water_heights (in_data)

  Combine median and average water heights.

5.80.1 Function Documentation

5.80.1.1 combine_water_heights()

```
def skdaccess.utilities.gw_util.combine_water_heights (
              in_data )
```

Combine median and average water heights.

Create a column of water heights in input data frame using Median Water Depth by default, but fills in missing data using average values

Parameters

in_data	Input water heights data

5.81 skdaccess.utilities.image_util Namespace Reference

Classes

- class AffineGlobalCoords

  Convert between projected and pixel coordinates using an affine transformation.
- class LinearGeolocation

  This class provides functions to convert between pixel and geodetic coordinates.
- class SplineLatLon

  Holds a 2d spline for interpolating lat/lon grid.

Functions

- def SplineGeolocation (object)

  This class holds splines to convert between 2d cartesian and geodetic coordinates.
- def getExtentsFromCentersPlateCarree (westmost_pixel_lon, eastmost_pixel_lon, southmost_pixel_lat, northmost_pixel_lat, lon_grid_spacing, lat_grid_spacing)
- def convertBinCentersToEdges (bin_centers, dtype=None)

  Calculate edges of a set of bins from their centers.
- def getGeoTransform (extents, x_size, y_size, y_flipped=True)

  Get 6 geotransform coefficients from the extents of an image and its shape.

Variables

- x_offset
- y_offset
- lat_spline
- lon_spline
- x_spline
- y_spline

5.81.1 Function Documentation

5.81.1.1 convertBinCentersToEdges()

```
def skdaccess.utilities.image_util.convertBinCentersToEdges (
              bin_centers,
              dtype = None )
```

Calculate edges of a set of bins from their centers.

Parameters

bin_centers	Array of bin centers
dtype	Data type of array used to store bin edges

Returns

bin_edges

5.81.1.2 getExtentsFromCentersPlateCarree()

```
def skdaccess.utilities.image_util.getExtentsFromCentersPlateCarree (
              westmost_pixel_lon,
              eastmost_pixel_lon,
              southmost_pixel_lat,
              northmost_pixel_lat,
              lon_grid_spacing,
              lat_grid_spacing )
```

5.81.1.3 getGeoTransform()

```
def skdaccess.utilities.image_util.getGeoTransform (
              extents,
              x_size,
              y_size,
              y_flipped = True )
```

Get 6 geotransform coefficients from the extents of an image and its shape.

Assumes origin is in the upper left and the x pixel coordinate does not depend on y projected coordinate, and the y pixl coordinate doesn't depend on the x projected coordinate

Parameters

extents	Image extents (x_min, x_max, y_min, y_max)
x_size	Number of x pixels
y_size	Number of y pixels
y_flipped	The y pixel coordinates are flipped relative to the projected coordinates

Returns

list containing the 6 affine transformation coordinates

5.81.1.4 SplineGeolocation()

```
def skdaccess.utilities.image_util.SplineGeolocation (
              object )
```

This class holds splines to convert between 2d cartesian and geodetic coordinates.

5.81.2 Variable Documentation

5.81.2.1 lat_spline

```
skdaccess.utilities.image_util.lat_spline
```

5.81.2.2 lon_spline

```
skdaccess.utilities.image_util.lon_spline
```

5.81.2.3 x_offset

```
skdaccess.utilities.image_util.x_offset
```

5.81.2.4 x_spline

```
skdaccess.utilities.image_util.x_spline
```

5.81.2.5 y_offset

```
skdaccess.utilities.image_util.y_offset
```

5.81.2.6 y_spline

```
skdaccess.utilities.image_util.y_spline
```

5.82 skdaccess.utilities.kepler_util Namespace Reference

Functions

- def normalize (in_data, column='PDCSAP_FLUX', group_column='QUARTER')

  This function normalizes PDCSAP_FLUX data by quarter by dividing the flux by the median for the quarter.

5.82.1 Function Documentation

5.82.1.1 normalize()

```
def skdaccess.utilities.kepler_util.normalize (
              in_data,
              column = 'PDCSAP_FLUX',
              group_column = 'QUARTER' )
```

This function normalizes PDCSAP_FLUX data by quarter by dividing the flux by the median for the quarter.

Parameters

in_data	Data to be normalized
column	Name of column to be normalized
group_column	Name of column used to group data

5.83 skdaccess.utilities.mahali_util Namespace Reference

Functions

- def convert_date (in_date)

  Converts input string to pandas date time, ignores other types of objects.
- def parseIonoFile (in_file, compression='infer')

5.83.1 Function Documentation

5.83.1.1 convert_date()

```
def skdaccess.utilities.mahali_util.convert_date (
              in_date )
```

Converts input string to pandas date time, ignores other types of objects.

Parameters

in_date	Input date

return pandas data time object

5.83.1.2 parseIonoFile()

```
def skdaccess.utilities.mahali_util.parseIonoFile (
              in_file,
              compression = 'infer' )
```

5.84 skdaccess.utilities.modis_util Namespace Reference

Classes

- class LatLon

  Calculates Lat/Lon position from y,x pixel coordinate.

Functions

- def getImageType (in_data)

  Determine what type of modis data is being processed.
- def calibrateModis (data, metadata)

  This function calibrates input modis data.
- def rescale (in_array, max_val=0.9, min_val=-0.01)

  This function rescales an image to fall between 0 and 1.
- def checkBit (data, bit)

  Get the bit value from a bit flag.
- def createGrid (data, y_start, y_end, x_start, x_end, y_grid, x_grid, dtype, grid_fill=np.nan)

  Subsets image data into a smaller image.
- def getFileIDs (modis_identifier, start_date, end_date, lat, lon, daynightboth)

  Retrieve file IDs for images matching search parameters.
- def getFileURLs (file_ids)

  Retrieve the ftp location for a list of file IDs.
- def getModisData (dataset, variable_name)

  Loads modis data.
- def readMODISData (modis_list, variables, grid, grid_fill, use_long_name, platform, product_id)

  Retrieve a list of modis data.

5.84.1 Function Documentation

5.84.1.1 calibrateModis()

```
def skdaccess.utilities.modis_util.calibrateModis (
              data,
              metadata )
```

This function calibrates input modis data.

Parameters

data	Input modis data
metadata	Metadata associated with modis input data

Returns

calibrated modis data

5.84.1.2 checkBit()

```
def skdaccess.utilities.modis_util.checkBit (
              data,
              bit )
```

Get the bit value from a bit flag.

Parameters

data	Integer bit flag
bit	Which bit to select (start indexing at 0)

Returns

value of chosen bit in bit flag

5.84.1.3 createGrid()

```
def skdaccess.utilities.modis_util.createGrid (
              data,
              y_start,
              y_end,
              x_start,
```

```
                x_end,
                y_grid,
                x_grid,
                dtype,
                grid_fill = np.nan )
```

Subsets image data into a smaller image.

Takes care to make sure the resulting subsection has the expected size by filling in missing data

Parameters

data	Input data
y_start	Starting pixel for y
y_end	Ending pixel for y
x_start	Starting pixel x
x_end	Ending pixel for x
y_grid	Grid size for y
x_grid	Grid size for x
dtype	The dtype of the new grid data
grid↩ _fill	Fill value to use when there is no data

Returns

image subsection, fraction of valid data

5.84.1.4 getFileIDs()

```
def skdaccess.utilities.modis_util.getFileIDs (
                modis_identifier,
                start_date,
                end_date,
                lat,
                lon,
                daynightboth )
```

Retrieve file IDs for images matching search parameters.

Parameters

modis_identifier	Product identifier (e.g. MOD09)
start_date	Starting date
end_date	Ending date
lat	Latitude
lon	Longitude
daynightboth	Get daytime images ('D'), nightime images ('N') or both ('B')

Returns

list of file IDs

5.84.1.5 getFileURLs()

```
def skdaccess.utilities.modis_util.getFileURLs (
              file_ids )
```

Retrieve the ftp location for a list of file IDs.

Parameters

file_ids	List of file IDs

Returns

List of ftp locations

5.84.1.6 getImageType()

```
def skdaccess.utilities.modis_util.getImageType (
              in_data )
```

Determine what type of modis data is being processed.

There are 3 array shapes we deal with:

```
mode 1 -> (y, x, z)
mode 2 -> (y, x)
mode 3 -> (z, y ,x)
```

where z axis represents different data products and y and x correspond to the y and x image coordinates from the modis instrument

Parameters

in_data	Input modis data

Returns

type of modis data

5.84.1.7 getModisData()

```
def skdaccess.utilities.modis_util.getModisData (
              dataset,
              variable_name )
```

Loads modis data.

Parameters

dataset	netCDF4 dataset
variable_name	Name of variable to extract from dataset

Returns

(modis_data, metadata)

5.84.1.8 readMODISData()

```
def skdaccess.utilities.modis_util.readMODISData (
              modis_list,
              variables,
              grid,
              grid_fill,
              use_long_name,
              platform,
              product_id )
```

Retrieve a list of modis data.

Parameters

modis_list	List of MODIS data to load
variables	List of variables in the MODIS data to load
grid	Further divide each image into a multiple grids of size (y,x)
grid_fill	Fill value to use when creating gridded data
use_long_name	Use long names for metadata instead of variable name
platform	Which satellite to use, either MOD or MYD.
product_id	Product string (e.g. '06_L2')

5.84.1.9 rescale()

```
def skdaccess.utilities.modis_util.rescale (
            in_array,
            max_val = 0.9,
            min_val = -0.01 )
```

This function rescales an image to fall between 0 and 1.

Parameters

in_array	Data to be rescaled
max_val	Values greater than or equal to max_val will become 1
min_val	Values less than or equal to min_val will become 0

Returns

scaled data

5.85 skdaccess.utilities.ode_util Namespace Reference

Functions

- def query_yes_no (question, default="yes")
- def get_query_url (target, mission, instrument, product_type, western_lon, eastern_lon, min_lat, max_lat, min↩_ob_time, max_ob_time, product_id, query_type, output, results, number_product_limit, result_offset_number)
- def get_files_urls (query_url, file_name=' ∗', print_info=False)
- def query_files_urls (target, mission, instrument, product_type, western_lon, eastern_lon, min_lat, max_lat, min↩_ob_time, max_ob_time, product_id, file_name, number_product_limit, result_offset_number)

  Retrieve the URL locations based on a query using ODE REST interface.
- def correct_CRISM_label (label_file_location)
- def correct_file_name_case_in_label (label_file_location, other_file_locations)
- def correct_label_file (label_file_location, other_file_locations=[ ])

  Correct a label file if GDAL cannot open the corresponding data file.
- def get_raster_array (gdal_raster, remove_ndv=True)

  Get a NumPy array from a raster opened with GDAL.
- def get_raster_extent (gdal_raster)

  Get the extent of a raster opened with GDAL.

5.85.1 Function Documentation

5.85.1.1 correct_CRISM_label()

```
def skdaccess.utilities.ode_util.correct_CRISM_label (
              label_file_location )
```

5.85.1.2 correct_file_name_case_in_label()

```
def skdaccess.utilities.ode_util.correct_file_name_case_in_label (
              label_file_location,
              other_file_locations )
```

5.85.1.3 correct_label_file()

```
def skdaccess.utilities.ode_util.correct_label_file (
              label_file_location,
              other_file_locations = [] )
```

Correct a label file if GDAL cannot open the corresponding data file.

Parameters

label_file_location	Local address of the current label
other_file_locations	Other files that were downloaded with the label file

Returns

Local address of the new label

5.85.1.4 get_files_urls()

```
def skdaccess.utilities.ode_util.get_files_urls (
              query_url,
              file_name = '*',
              print_info = False )
```

5.85.1.5 get_query_url()

```
def skdaccess.utilities.ode_util.get_query_url (
              target,
              mission,
              instrument,
              product_type,
              western_lon,
              eastern_lon,
              min_lat,
              max_lat,
              min_ob_time,
              max_ob_time,
              product_id,
              query_type,
              output,
              results,
              number_product_limit,
              result_offset_number )
```

5.85.1.6 get_raster_array()

```
def skdaccess.utilities.ode_util.get_raster_array (
              gdal_raster,
              remove_ndv = True )
```

Get a NumPy array from a raster opened with GDAL.

Parameters

gdal_raster	A raster opened with GDAL
remove_ndv	Replace the no-data value as mentionned in the label by np.nan

Returns

The array

5.85.1.7 get_raster_extent()

```
def skdaccess.utilities.ode_util.get_raster_extent (
              gdal_raster )
```

Get the extent of a raster opened with GDAL.

Parameters

gdal_raster	A raster opened with GDAL

Returns

The raster extent

5.85.1.8 query_files_urls()

```
def skdaccess.utilities.ode_util.query_files_urls (
              target,
              mission,
              instrument,
              product_type,
              western_lon,
              eastern_lon,
              min_lat,
              max_lat,
              min_ob_time,
              max_ob_time,
              product_id,
              file_name,
              number_product_limit,
              result_offset_number )
```

Retrieve the URL locations based on a query using ODE REST interface.

Parameters

target	Aimed planetary body, i.e., Mars, Mercury, Moon, Phobos, or Venus
mission	Aimed mission, e.g., MGS or MRO
instrument	Aimed instrument from the mission, e.g., HIRISE or CRISM
product_type	Type of product to look for, e.g., DTM or RDRV11
western_lon	Western longitude to look for the data, from 0 to 360
eastern_lon	Eastern longitude to look for the data, from 0 to 360
min_lat	Minimal latitude to look for the data, from -90 to 90
max_lat	Maximal latitude to look for the data, from -90 to 90
min_ob_time	Minimal observation time in (even partial) UTC format, e.g., '2017-03-01'
max_ob_time	Maximal observation time in (even partial) UTC format, e.g., '2017-03-01'
product_id	PDS Product Id to look for, with wildcards ($*$) allowed
file_name	File name to look for, with wildcards ($*$) allowed
number_product_limit	Maximal number of products to return (100 at most)
result_offset_number	Offset the return products, to go beyond the limit of 100 returned products

Returns

List of URL locations

5.85.1.9 query_yes_no()

```
def skdaccess.utilities.ode_util.query_yes_no (
            question,
            default = "yes" )
```

5.86 skdaccess.utilities.pbo_util Namespace Reference

Functions

- def getStationCoords (pbo_info, station_list)

  Get the station coordinates for a list of stations.
- def getLatLonRange (pbo_info, station_list)

  Retrive the range of latitude and longitude occupied by a set of stations.
- def getROIstations (geo_point, radiusParam, data, header)

  This function returns the 4ID station codes for the stations in a region.
- def stab_sys (data_iterator, metadata, stab_min_NE=.0005, stab_min_U=.005, sigsc=2, errProp=1)

  Stabilize GPS data to a region.
- def propagateErrors (R, sc, stationCovs)

  Propagate GPS errors.
- def nostab_sys (allH, allD, timerng, indx=1, mdyratio=.7, use_progress_bar=True, index_date_only=False)

  Do not apply stabilization and simply returns stations after checking for sufficient amount of data.
- def removeAntennaOffset (antenna_offsets, data, window_start=pd.to_timedelta('4D'), window_end=pd.to_timedelta('4D'), min_diff=0.005, debug=False)

  Remove offsets caused by changes in antennas.

5.86.1 Function Documentation

5.86.1.1 getLatLonRange()

```
def skdaccess.utilities.pbo_util.getLatLonRange (
            pbo_info,
            station_list )
```

Retrive the range of latitude and longitude occupied by a set of stations.

Parameters

pbo_info	PBO Metadata
station_list	List of stations

Returns

list containg two tuples, lat_range and lon_range

5.86.1.2 getROIstations()

```
def skdaccess.utilities.pbo_util.getROIstations (
              geo_point,
              radiusParam,
              data,
              header )
```

This function returns the 4ID station codes for the stations in a region.

The region of interest is defined by the geographic coordinate and a window size

Parameters

geo_point	The geographic (lat,lon) coordinate of interest
radiusParam	An overloaded radius of interest [km] or latitude and longitude window [deg] around the geo_point
data	Stabilized (or unstabilized) data generated from the data fetcher or out of stab_sys
header	Header dictionary with stations metadata keyed by their 4ID code. This is output with the data.

Returns

station_list, list of site 4ID codes in the specified geographic region

5.86.1.3 getStationCoords()

```
def skdaccess.utilities.pbo_util.getStationCoords (
              pbo_info,
              station_list )
```

Get the station coordinates for a list of stations.

Parameters

pbo_info	PBO Metadata
station_list	List of stations

Returns

list of tuples containing lat, lon coordinates of stations

5.86.1.4 nostab_sys()

```
def skdaccess.utilities.pbo_util.nostab_sys (
              allH,
              allD,
              timerng,
              indx = 1,
              mdyratio = .7,
              use_progress_bar = True,
              index_date_only = False )
```

Do not apply stabilization and simply returns stations after checking for sufficient amount of data.

Parameters

allH	a dictionary of all of the headers of all sites loaded from the data directory
allD	a dictionary of all of the panda format data of all of the corresponding sites
timerng	an array with two string elements, describing the starting and ending dates
indx	a list of site 4ID's indicating stations in the relevant geographic location, or 1 for all sites
mdyratio	optional parameter for the minimum required ratio of data to determine if a sitef is kept for further analysis
use_progress_bar	Display a progress bar
index_date_only	When creating an index for the data, use date (not the time) only

Returns

smSet, a reduced size dictionary of the data (in meters) for the sites in the specified geographic region and smHdr, a reduced size dictionary of the headers for the sites in the region

5.86.1.5 propagateErrors()

```
def skdaccess.utilities.pbo_util.propagateErrors (
              R,
```

```
                sc,
                stationCovs )
```

Propagate GPS errors.

By writing out the R∗E∗R.T equations... to calculate the new covariance matrix without needing to form the matrix first as an intermediate step. Modifies covariance matrix in place

Parameters

R	Rotation matrix
sc	Scaling value
stationCovs	Station Covariances

5.86.1.6 removeAntennaOffset()

```
def skdaccess.utilities.pbo_util.removeAntennaOffset (
                antenna_offsets,
                data,
                window_start = pd.to_timedelta('4D'),
                window_end = pd.to_timedelta('4D'),
                min_diff = 0.005,
                debug = False )
```

Remove offsets caused by changes in antennas.

Parameters

antenna_offsets	Pandas series of dates describing when the antenna changes were made
data	Input GPS data
window_start	Starting time before and after event to use for calculating offset
window_end	Ending time before and after event to use before calculating offset
min_diff	Minimum difference before and after offset to for applying correction
debug	Enable debug output

Returns

GPS data with the offsets removed

5.86.1.7 stab_sys()

```
def skdaccess.utilities.pbo_util.stab_sys (
                data_iterator,
```

```
                metadata,
                stab_min_NE = .0005,
                stab_min_U = .005,
                sigsc = 2,
                errProp = 1 )
```

Stabilize GPS data to a region.

The stab_sys function is a Python implemention of the Helmhert 7-parameter transformation, used to correct for common mode error. This builds on Prof Herring's stab_sys function in his tscon Fortran code. It uses a SVD approach to estimating the rotation matrix gathered from 'Computing Helmert Transformations' by G.A. Watson as well as its references. Note that units should be in meters, that is in the format from the level 2 processed UNAVCO pos files

Parameters

data_iterator	Expects an iterator that returns label, pandas dataframe
metadata	Metadata that contains 'refXYZ' and 'refNEU'
stab_min_NE	Optional minimum horizontal covariance parameter
stab_min_U	Optional minimum vertical covariance parameter
sigsc	Optional scaling factor for determining cutoff bounds for non stable sites
errProp	Propagate errors through the transformation

Returns

smSet, a reduced size dictionary of the data (in mm) for the sites in the specified geographic region, smHdr, a reduced size dictionary of the headers for the sites in the region

5.87 skdaccess.utilities.sentinel_1_util Namespace Reference

Functions

- def parseSatelliteData (in_satellite_file)

  Parse Sentinel satelllite data.

5.87.1 Function Documentation

5.87.1.1 parseSatelliteData()

```
def skdaccess.utilities.sentinel_1_util.parseSatelliteData (
                in_satellite_file )
```

Parse Sentinel satelllite data.

Parameters

in_satellite_file	Satellite orbit filename

Returns

DataFrame of orbit information

5.88 skdaccess.utilities.sounding_util Namespace Reference

Classes

- class SoundingParser

  This class parses Wyoming Sounding data.

Functions

- def generateQueries (station_number, year_list, month_list, day_start, day_end, start_hour, end_hour)

  Generate url queries for sounding data.

5.88.1 Function Documentation

5.88.1.1 generateQueries()

```
def skdaccess.utilities.sounding_util.generateQueries (
              station_number,
              year_list,
              month_list,
              day_start,
              day_end,
              start_hour,
              end_hour )
```

Generate url queries for sounding data.

Parameters

station_number	Input station number
year_list	Input years as a list
month_list	Input month as a list
day_start	Starting day
day_end	Ending day
start_hour	Starting hour
end_hour	Ending hour

Returns

list of urls containing requested data

5.89 skdaccess.utilities.srtm_util Namespace Reference

Functions

- def merge_srtm_tiles (srtm_tiles, lon_min, lon_max, lat_min, lat_max)
- def getSRTMLatLon (lat_min, lat_max, lon_min, lon_max)

  Retrieve parameters that encompass area when creating SRTM data fetcher.
- def getSRTMData (srtmdw, lat_start, lat_end, lon_start, lon_end)

  Select SRTM data in a latitude/longitude box.

5.89.1 Function Documentation

5.89.1.1 getSRTMData()

```
def skdaccess.utilities.srtm_util.getSRTMData (
              srtmdw,
              lat_start,
              lat_end,
              lon_start,
              lon_end )
```

Select SRTM data in a latitude/longitude box.

Parameters

srtmdw	SRTM data wrapper
lat_start	Starting latiude
lat_end	Ending latiude
lon_start	Starting longitude
lon_end	Ending longitude
flip_y	Flip the y axis so that increasing y pixels are increasing in latitude

Returns

Tuple containing the cut data, new extents, and a affine geotransform coefficients

5.89.1.2 getSRTMLatLon()

```
def skdaccess.utilities.srtm_util.getSRTMLatLon (
              lat_min,
              lat_max,
              lon_min,
              lon_max )
```

Retrieve parameters that encompass area when creating SRTM data fetcher.

Parameters

lat_min	Minimum latitude
lat_max	Maximum latitude
lon_min	Minimum longitude
lon_max	Maximum longitude

Returns

(starting_latitude, ending_latitude, starting_longitude, ending_longitude)

5.89.1.3 merge_srtm_tiles()

```
def skdaccess.utilities.srtm_util.merge_srtm_tiles (
              srtm_tiles,
              lon_min,
              lon_max,
              lat_min,
              lat_max )
```

5.90 skdaccess.utilities.support Namespace Reference

Functions

- def retrieveCommonDatesHDF (support_data_filename, key_list, in_date_list)

  Get a list of all dates that have data available.
- def progress_bar (in_iterable, total=None, enabled=True)

  Progess bar using tqdm.
- def convertToStr (in_value, zfill=0)

5.90.1 Function Documentation

5.90.1.1 convertToStr()

```
def skdaccess.utilities.support.convertToStr (
              in_value,
              zfill = 0 )
```

5.90.1.2 progress_bar()

```
def skdaccess.utilities.support.progress_bar (
              in_iterable,
              total = None,
              enabled = True )
```

Progess bar using tqdm.

Parameters

in_iterable	Input iterable
total	Total number of elements
enabled	Enable progress bar

5.90.1.3 retrieveCommonDatesHDF()

```
def skdaccess.utilities.support.retrieveCommonDatesHDF (
              support_data_filename,
              key_list,
              in_date_list )
```

Get a list of all dates that have data available.

Parameters

support_data_filename	Filename of support data
key_list	List of keys in HDF file
in_date_list	Input date list to check

Returns

dictionary of dates with data

5.91 skdaccess.utilities.uavsar_util Namespace Reference

Functions

- def readUAVSARMetadata (in_file)

  Parse UAVSAR metadata.

5.91.1 Function Documentation

5.91.1.1 readUAVSARMetadata()

```
def skdaccess.utilities.uavsar_util.readUAVSARMetadata (
              in_file )
```

Parse UAVSAR metadata.

Parameters

in_file	String of Metadata filename or file object (file should end in .ann)

Returns

OrderedDict of metadata

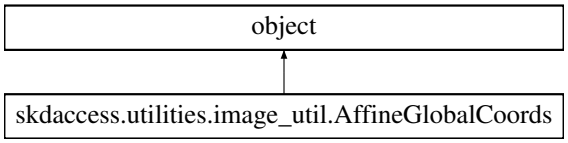
Chapter 6

Class Documentation

6.1 skdaccess.utilities.image_util.AffineGlobalCoords Class Reference

Convert between projected and pixel coordinates using an affine transformation.

Inheritance diagram for skdaccess.utilities.image_util.AffineGlobalCoords:



Public Member Functions

- def __init__ (self, aff_coeffs, center_pixels=False)

  Initialize Global Coords Object.
- def getProjectedYX (self, y_array, x_array)

  Convert pixel coordinates to projected coordinates.
- def getPixelYX (self, y_proj, x_proj)

  Convert from projected coordinates to pixel coordinates.

6.1.1 Detailed Description

Convert between projected and pixel coordinates using an affine transformation.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def skdaccess.utilities.image_util.AffineGlobalCoords.__init__ (
              self,
              aff_coeffs,
              center_pixels = False )
```

Initialize Global Coords Object.

Parameters

aff_coeffs	Affine coefficients
center_pixels	Apply offsets so that integer values refer to the center of the pixel and not the edge

6.1.3 Member Function Documentation

6.1.3.1 getPixelYX()

```
def skdaccess.utilities.image_util.AffineGlobalCoords.getPixelYX (
              self,
              y_proj,
              x_proj )
```

Convert from projected coordinates to pixel coordinates.

Parameters

y_proj	Input projected y coordinates
x_proj	Input projected x coordinates

Returns

y pixel coordinates, x pixel coordinates

6.1.3.2 getProjectedYX()

```
def skdaccess.utilities.image_util.AffineGlobalCoords.getProjectedYX (
              self,
              y_array,
              x_array )
```

Convert pixel coordinates to projected coordinates.

Parameters

y_array	Input y pixel coordinates
x_array	Input x pixel coordinates

Returns

projected y coordinates, projected x coordinates

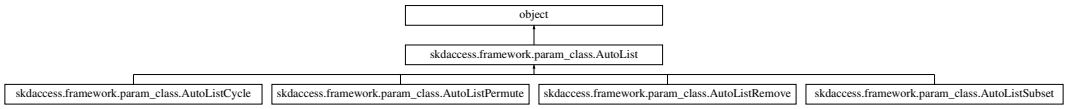
The documentation for this class was generated from the following file:

- utilities/image_util.py

6.2 skdaccess.framework.param_class.AutoList Class Reference

Specifies a list for returning selections of lists, as opposed to a single element.

Inheritance diagram for skdaccess.framework.param_class.AutoList:



Public Member Functions

- def __init__ (self, val_list)

  Construct a AutoList object.
- def val (self)

  Retrieves current list of parameters.
- def perturb (self)

  This class doesn't change the list when being perturbed.
- def reset (self)

  Reset current list to initial list.
- def getAllOptions (self)

  Get all possible options.
- def __str__ (self)

  String representation of class.
- def __len__ (self)

  Retrieves the length of parameters contained in the list.
- def __getitem__ (self, ii)

  Retrieves item from list.
- def __setitem__ (self, ii, val)

  Set a value in the list.
- def __call__ (self)

  Retrieve current list.

Public Attributes

- val_init
- val_list

6.2.1 Detailed Description

Specifies a list for returning selections of lists, as opposed to a single element.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 __init__()

```
def skdaccess.framework.param_class.AutoList.__init__ (
              self,
              val_list )
```

Construct a AutoList object.

Parameters

val_list	List of parameters

6.2.3 Member Function Documentation

6.2.3.1 __call__()

```
def skdaccess.framework.param_class.AutoList.__call__ (
              self )
```

Retrieve current list.

Returns

Current list

6.2.3.2 __getitem__()

```
def skdaccess.framework.param_class.AutoList.__getitem__ (
              self,
              ii )
```

Retrieves item from list.

Parameters

ii	Index of item to be retrieved

Returns

Item at index ii

6.2.3.3 __len__()

```
def skdaccess.framework.param_class.AutoList.__len__ (
              self )
```

Retrieves the length of parameters contained in the list.

Returns

Number of elements in the list

6.2.3.4 __setitem__()

```
def skdaccess.framework.param_class.AutoList.__setitem__ (
              self,
              ii,
              val )
```

Set a value in the list.

Parameters

ii	Index of list to be set
val	Input value

6.2.3.5 __str__()

```
def skdaccess.framework.param_class.AutoList.__str__ (
              self )
```

String representation of class.

Returns

String containing all parmaters in list

6.2.3.6 getAllOptions()

```
def skdaccess.framework.param_class.AutoList.getAllOptions (
              self )
```

Get all possible options.

Returns

List that contains every option that could possibly be selected

6.2.3.7 perturb()

```
def skdaccess.framework.param_class.AutoList.perturb (
              self )
```

This class doesn't change the list when being perturbed.

6.2.3.8 reset()

```
def skdaccess.framework.param_class.AutoList.reset (
              self )
```

Reset current list to initial list.

6.2.3.9 val()

```
def skdaccess.framework.param_class.AutoList.val (
              self )
```

Retrieves current list of parameters.

Returns

List of current parameters

6.2.4 Member Data Documentation

6.2.4.1 val_init

```
skdaccess.framework.param_class.AutoList.val_init
```

6.2.4.2 val_list

```
skdaccess.framework.param_class.AutoList.val_list
```

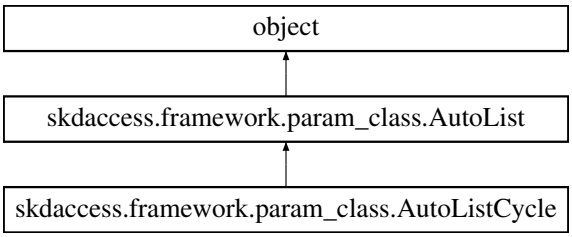
The documentation for this class was generated from the following file:

- framework/param_class.py

6.3 skdaccess.framework.param_class.AutoListCycle Class Reference

An Autolist that cycles through different lists.

Inheritance diagram for skdaccess.framework.param_class.AutoListCycle:



Public Member Functions

- def __init__ (self, list_val_list)

  Construct a AutoList_Cycle object.
- def perturb (self)

  Select next list from list of lists.
- def reset (self)

  Resets to the first list in the list of lists.
- def getAllOptions (self)

  Get elements that could possibly be called.
- def val (self)

  Retrieves current list of parameters.
- def __str__ (self)

  String representation of class.
- def __len__ (self)

  Retrieves the length of parameters contained in the list.
- def __getitem__ (self, ii)

  Retrieves item from list.
- def __setitem__ (self, ii, val)

  Set a value in the list.
- def __call__ (self)

  Retrieve current list.

Public Attributes

- list_val_list
- val_list
- index
- val_init

6.3.1 Detailed Description

An Autolist that cycles through different lists.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 __init__()

```
def skdaccess.framework.param_class.AutoListCycle.__init__ (
            self,
            list_val_list )
```

Construct a AutoList_Cycle object.

Parameters

list_val_list	List of different lists to cycle through

6.3.3 Member Function Documentation

6.3.3.1 __call__()

```
def skdaccess.framework.param_class.AutoList.__call__ (
              self )  [inherited]
```

Retrieve current list.

Returns

Current list

6.3.3.2 __getitem__()

```
def skdaccess.framework.param_class.AutoList.__getitem__ (
              self,
              ii )  [inherited]
```

Retrieves item from list.

Parameters

ii	Index of item to be retrieved

Returns

Item at index ii

6.3.3.3 __len__()

```
def skdaccess.framework.param_class.AutoList.__len__ (
              self )  [inherited]
```

Retrieves the length of parameters contained in the list.

Returns

Number of elements in the list

6.3.3.4 __setitem__()

```
def skdaccess.framework.param_class.AutoList.__setitem__ (
              self,
              ii,
              val )  [inherited]
```

Set a value in the list.

Parameters

ii	Index of list to be set
val	Input value

6.3.3.5 __str__()

```
def skdaccess.framework.param_class.AutoList.__str__ (
              self )  [inherited]
```

String representation of class.

Returns

String containing all parmaters in list

6.3.3.6 getAllOptions()

```
def skdaccess.framework.param_class.AutoListCycle.getAllOptions (
              self )
```

Get elements that could possibly be called.

Returns

List of all possible elements

6.3.3.7 perturb()

```
def skdaccess.framework.param_class.AutoListCycle.perturb (
              self )
```

Select next list from list of lists.

6.3.3.8 reset()

```
def skdaccess.framework.param_class.AutoListCycle.reset (
              self )
```

Resets to the first list in the list of lists.

6.3.3.9 val()

```
def skdaccess.framework.param_class.AutoList.val (
              self )  [inherited]
```

Retrieves current list of parameters.

Returns

List of current parameters

6.3.4 Member Data Documentation

6.3.4.1 index

```
skdaccess.framework.param_class.AutoListCycle.index
```

6.3.4.2 list_val_list

```
skdaccess.framework.param_class.AutoListCycle.list_val_list
```

6.3.4.3 val_init

```
skdaccess.framework.param_class.AutoList.val_init  [inherited]
```

6.3.4.4 val_list

```
skdaccess.framework.param_class.AutoListCycle.val_list
```

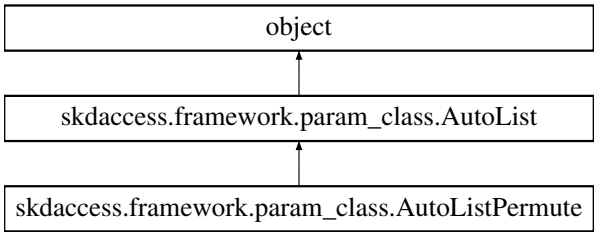
The documentation for this class was generated from the following file:

- framework/param_class.py

6.4 skdaccess.framework.param_class.AutoListPermute Class Reference

A perturber that permutes a list.

Inheritance diagram for skdaccess.framework.param_class.AutoListPermute:



Public Member Functions

- def perturb (self)

  Randomly permutes the initial list.
- def val (self)

  Retrieves current list of parameters.
- def reset (self)

  Reset current list to initial list.
- def getAllOptions (self)

  Get all possible options.
- def __str__ (self)

  String representation of class.
- def __len__ (self)

  Retrieves the length of parameters contained in the list.
- def __getitem__ (self, ii)

  Retrieves item from list.
- def __setitem__ (self, ii, val)

  Set a value in the list.
- def __call__ (self)

  Retrieve current list.

Public Attributes

- val_init
- val_list

6.4.1 Detailed Description

A perturber that permutes a list.

6.4.2 Member Function Documentation

6.4.2.1 __call__()

```
def skdaccess.framework.param_class.AutoList.__call__ (
              self )  [inherited]
```

Retrieve current list.

Returns

Current list

6.4.2.2 __getitem__()

```
def skdaccess.framework.param_class.AutoList.__getitem__ (
              self,
              ii )  [inherited]
```

Retrieves item from list.

Parameters

ii	Index of item to be retrieved

Returns

Item at index ii

6.4.2.3 __len__()

```
def skdaccess.framework.param_class.AutoList.__len__ (
              self )  [inherited]
```

Retrieves the length of parameters contained in the list.

Returns

Number of elements in the list

6.4.2.4 __setitem__()

```
def skdaccess.framework.param_class.AutoList.__setitem__ (
              self,
              ii,
              val )  [inherited]
```

Set a value in the list.

Parameters

ii	Index of list to be set
val	Input value

6.4.2.5 __str__()

```
def skdaccess.framework.param_class.AutoList.__str__ (
              self )  [inherited]
```

String representation of class.

Returns

String containing all parmaters in list

6.4.2.6 getAllOptions()

```
def skdaccess.framework.param_class.AutoList.getAllOptions (
              self )  [inherited]
```

Get all possible options.

Returns

List that contains every option that could possibly be selected

6.4.2.7 perturb()

```
def skdaccess.framework.param_class.AutoListPermute.perturb (
              self )
```

Randomly permutes the initial list.

6.4.2.8 reset()

```
def skdaccess.framework.param_class.AutoList.reset (
              self )  [inherited]
```

Reset current list to initial list.

6.4.2.9 val()

```
def skdaccess.framework.param_class.AutoList.val (
              self )  [inherited]
```

Retrieves current list of parameters.

Returns

List of current parameters

6.4.3 Member Data Documentation

6.4.3.1 val_init

```
skdaccess.framework.param_class.AutoList.val_init  [inherited]
```

6.4.3.2 val_list

```
skdaccess.framework.param_class.AutoList.val_list  [inherited]
```

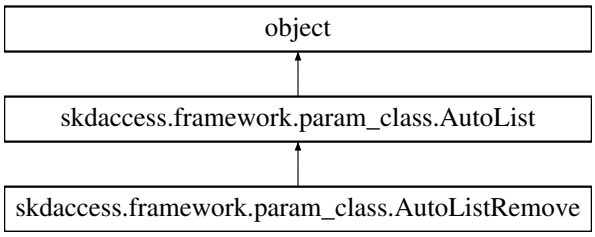
The documentation for this class was generated from the following file:

- framework/param_class.py

6.5 skdaccess.framework.param_class.AutoListRemove Class Reference

Removes a different single element from the initial list at each perturb call.

Inheritance diagram for skdaccess.framework.param_class.AutoListRemove:



Public Member Functions

- def __init__ (self, val_list)

  Construct a AutoList_Cycle object.
- def perturb (self)

  Systematically change which item is absent from the list.
- def reset (self)

  Reset the list to its initial value.
- def val (self)

  Retrieves current list of parameters.
- def getAllOptions (self)

  Get all possible options.
- def __str__ (self)

  String representation of class.
- def __len__ (self)

  Retrieves the length of parameters contained in the list.
- def __getitem__ (self, ii)

  Retrieves item from list.
- def __setitem__ (self, ii, val)

  Set a value in the list.
- def __call__ (self)

  Retrieve current list.

Public Attributes

- n
- val_list
- val_init

6.5.1 Detailed Description

Removes a different single element from the initial list at each perturb call.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 __init__()

```
def skdaccess.framework.param_class.AutoListRemove.__init__ (
              self,
              val_list )
```

Construct a AutoList_Cycle object.

Parameters

val_list	Initial list of parameters.

6.5.3 Member Function Documentation

6.5.3.1 __call__()

```
def skdaccess.framework.param_class.AutoList.__call__ (
              self )  [inherited]
```

Retrieve current list.

Returns

Current list

6.5.3.2 __getitem__()

```
def skdaccess.framework.param_class.AutoList.__getitem__ (
              self,
              ii )  [inherited]
```

Retrieves item from list.

Parameters

ii	Index of item to be retrieved

Returns

Item at index ii

6.5.3.3 __len__()

```
def skdaccess.framework.param_class.AutoList.__len__ (
              self )  [inherited]
```

Retrieves the length of parameters contained in the list.

Returns

Number of elements in the list

6.5.3.4 __setitem__()

```
def skdaccess.framework.param_class.AutoList.__setitem__ (
              self,
              ii,
              val )  [inherited]
```

Set a value in the list.

Parameters

ii	Index of list to be set
val	Input value

6.5.3.5 __str__()

```
def skdaccess.framework.param_class.AutoList.__str__ (
              self )  [inherited]
```

String representation of class.

Returns

String containing all parmaters in list

6.5.3.6 getAllOptions()

```
def skdaccess.framework.param_class.AutoList.getAllOptions (
              self )  [inherited]
```

Get all possible options.

Returns

List that contains every option that could possibly be selected

6.5.3.7 perturb()

```
def skdaccess.framework.param_class.AutoListRemove.perturb (
              self )
```

Systematically change which item is absent from the list.

6.5.3.8 reset()

```
def skdaccess.framework.param_class.AutoListRemove.reset (
              self )
```

Reset the list to its initial value.

6.5.3.9 val()

```
def skdaccess.framework.param_class.AutoList.val (
              self )  [inherited]
```

Retrieves current list of parameters.

Returns

List of current parameters

6.5.4 Member Data Documentation

6.5.4.1 n

```
skdaccess.framework.param_class.AutoListRemove.n
```

6.5.4.2 val_init

```
skdaccess.framework.param_class.AutoList.val_init  [inherited]
```

6.5.4.3 val_list

```
skdaccess.framework.param_class.AutoListRemove.val_list
```

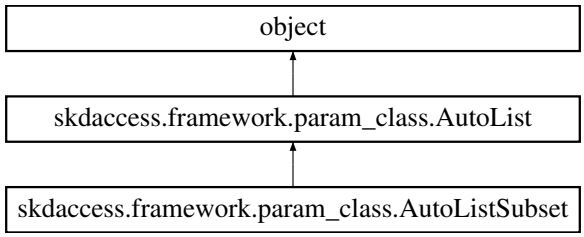
The documentation for this class was generated from the following file:

- framework/param_class.py

6.6 skdaccess.framework.param_class.AutoListSubset Class Reference

An AutoList perturber that creates random subsets of a list.

Inheritance diagram for skdaccess.framework.param_class.AutoListSubset:



Public Member Functions

- def perturb (self)

  Peturb the list by selecting a random subset of the initial list.
- def val (self)

  Retrieves current list of parameters.
- def reset (self)

  Reset current list to initial list.
- def getAllOptions (self)

  Get all possible options.
- def __str__ (self)

  String representation of class.
- def __len__ (self)

  Retrieves the length of parameters contained in the list.
- def __getitem__ (self, ii)

  Retrieves item from list.
- def __setitem__ (self, ii, val)

  Set a value in the list.
- def __call__ (self)

  Retrieve current list.

Public Attributes

- val_list
- val_init

6.6.1 Detailed Description

An AutoList perturber that creates random subsets of a list.

List can be empty

6.6.2 Member Function Documentation

6.6.2.1 __call__()

```
def skdaccess.framework.param_class.AutoList.__call__ (
              self )  [inherited]
```

Retrieve current list.

Returns

Current list

6.6.2.2 __getitem__()

```
def skdaccess.framework.param_class.AutoList.__getitem__ (
              self,
              ii )  [inherited]
```

Retrieves item from list.

Parameters

ii	Index of item to be retrieved

Returns

Item at index ii

6.6.2.3 __len__()

```
def skdaccess.framework.param_class.AutoList.__len__ (
              self )  [inherited]
```

Retrieves the length of parameters contained in the list.

Returns

Number of elements in the list

6.6.2.4 __setitem__()

```
def skdaccess.framework.param_class.AutoList.__setitem__ (
              self,
              ii,
              val )  [inherited]
```

Set a value in the list.

Parameters

ii	Index of list to be set
val	Input value

6.6.2.5 __str__()

```
def skdaccess.framework.param_class.AutoList.__str__ (
              self )  [inherited]
```

String representation of class.

Returns

String containing all parmaters in list

6.6.2.6 getAllOptions()

```
def skdaccess.framework.param_class.AutoList.getAllOptions (
              self )  [inherited]
```

Get all possible options.

Returns

List that contains every option that could possibly be selected

6.6.2.7 perturb()

```
def skdaccess.framework.param_class.AutoListSubset.perturb (
              self )
```

Peturb the list by selecting a random subset of the initial list.

6.6.2.8 reset()

```
def skdaccess.framework.param_class.AutoList.reset (
              self )  [inherited]
```

Reset current list to initial list.

6.6.2.9 val()

```
def skdaccess.framework.param_class.AutoList.val (
              self )  [inherited]
```

Retrieves current list of parameters.

Returns

List of current parameters

6.6.3 Member Data Documentation

6.6.3.1 val_init

```
skdaccess.framework.param_class.AutoList.val_init  [inherited]
```

6.6.3.2 val_list

```
skdaccess.framework.param_class.AutoListSubset.val_list
```

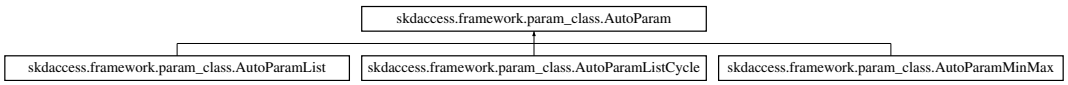
The documentation for this class was generated from the following file:

- framework/param_class.py

6.7 skdaccess.framework.param_class.AutoParam Class Reference

Defines a tunable parameter class inherited by specific subclasses.

Inheritance diagram for skdaccess.framework.param_class.AutoParam:



Public Member Functions

- def __init__ (self, val_init)

  Initialize an AutoParam object.
- def perturb (self)

  Perturb paramter.
- def reset (self)

  Reset value to initial value.
- def __str__ (self)

  String representation of class.
- def __call__ (self)

  Retrieves current value of the parameter.

Public Attributes

- val
- val_init

6.7.1 Detailed Description

Defines a tunable parameter class inherited by specific subclasses.

AutoParam class and subclass work on a single value. functions perturb value and reset to initial value

6.7.2 Constructor & Destructor Documentation

6.7.2.1 __init__()

```
def skdaccess.framework.param_class.AutoParam.__init__ (
            self,
            val_init )
```

Initialize an AutoParam object.

Parameters

val_init	Value for parameter

6.7.3 Member Function Documentation

6.7.3.1 __call__()

```
def skdaccess.framework.param_class.AutoParam.__call__ (
              self )
```

Retrieves current value of the parameter.

Returns

Current value of the parameter

6.7.3.2 __str__()

```
def skdaccess.framework.param_class.AutoParam.__str__ (
              self )
```

String representation of class.

Returns

String of current value

6.7.3.3 perturb()

```
def skdaccess.framework.param_class.AutoParam.perturb (
              self )
```

Perturb paramter.

This class doesn't change the value.

6.7.3.4 reset()

```
def skdaccess.framework.param_class.AutoParam.reset (
              self )
```

Reset value to initial value.

6.7.4 Member Data Documentation

6.7.4.1 val

```
skdaccess.framework.param_class.AutoParam.val
```

6.7.4.2 val_init

```
skdaccess.framework.param_class.AutoParam.val_init
```

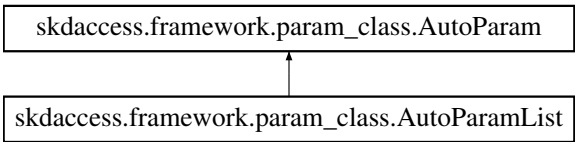
The documentation for this class was generated from the following file:

- framework/param_class.py

6.8 skdaccess.framework.param_class.AutoParamList Class Reference

A tunable parameter with a specified list of choices that can be randomly selected via perturb.

Inheritance diagram for skdaccess.framework.param_class.AutoParamList:



Public Member Functions

- def __init__ (self, val_init, val_list)

  Construct an AutoParamList object.
- def perturb (self)

  Randomly select a value from val_list.
- def reset (self)

  Reset the list to the default value.
- def __str__ (self)

  String representation of class.
- def __call__ (self)

  Retrieves current value of the parameter.

Public Attributes

- val
- val_init
- val_list

6.8.1 Detailed Description

A tunable parameter with a specified list of choices that can be randomly selected via perturb.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 __init__()

```
def skdaccess.framework.param_class.AutoParamList.__init__ (
              self,
              val_init,
              val_list )
```

Construct an AutoParamList object.

Parameters

val_init	initial value for the parameter
val_list	List of possible variants for the parameter

6.8.3 Member Function Documentation

6.8.3.1 __call__()

```
def skdaccess.framework.param_class.AutoParam.__call__ (
              self )  [inherited]
```

Retrieves current value of the parameter.

Returns

Current value of the parameter

6.8.3.2 __str__()

```
def skdaccess.framework.param_class.AutoParam.__str__ (
              self )  [inherited]
```

String representation of class.

Returns

String of current value

6.8.3.3 perturb()

```
def skdaccess.framework.param_class.AutoParamList.perturb (
              self )
```

Randomly select a value from val_list.

6.8.3.4 reset()

```
def skdaccess.framework.param_class.AutoParamList.reset (
              self )
```

Reset the list to the default value.

6.8.4 Member Data Documentation

6.8.4.1 val

```
skdaccess.framework.param_class.AutoParamList.val
```

6.8.4.2 val_init

```
skdaccess.framework.param_class.AutoParamList.val_init
```

6.8.4.3 val_list

`skdaccess.framework.param_class.AutoParamList.val_list`

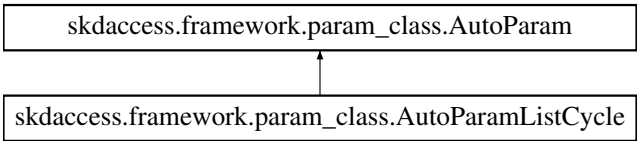
The documentation for this class was generated from the following file:

- framework/param_class.py

6.9 skdaccess.framework.param_class.AutoParamListCycle Class Reference

Cycles through a list of paramters.

Inheritance diagram for skdaccess.framework.param_class.AutoParamListCycle:



Public Member Functions

- def __init__ (self, val_list)

  Construct an AutoParamListCycle.
- def perturb (self)

  Select the next value from the list of parameters.
- def reset (self)

  Reset the list to the default values.
- def __str__ (self)

  String representation of class.
- def __call__ (self)

  Retrieves current value of the parameter.

Public Attributes

- val
- val_list
- current_index
- val_init

6.9.1 Detailed Description

Cycles through a list of paramters.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 __init__()

```
def skdaccess.framework.param_class.AutoParamListCycle.__init__ (
              self,
              val_list )
```

Construct an AutoParamListCycle.

Parameters

val_list	List of possible variants for the parameter

6.9.3 Member Function Documentation

6.9.3.1 __call__()

```
def skdaccess.framework.param_class.AutoParam.__call__ (
              self )  [inherited]
```

Retrieves current value of the parameter.

Returns

Current value of the parameter

6.9.3.2 __str__()

```
def skdaccess.framework.param_class.AutoParam.__str__ (
              self )  [inherited]
```

String representation of class.

Returns

String of current value

6.9.3.3 perturb()

```
def skdaccess.framework.param_class.AutoParamListCycle.perturb (
              self )
```

Select the next value from the list of parameters.

6.9.3.4 reset()

```
def skdaccess.framework.param_class.AutoParamListCycle.reset (
              self )
```

Reset the list to the default values.

6.9.4 Member Data Documentation

6.9.4.1 current_index

```
skdaccess.framework.param_class.AutoParamListCycle.current_index
```

6.9.4.2 val

```
skdaccess.framework.param_class.AutoParamListCycle.val
```

6.9.4.3 val_init

```
skdaccess.framework.param_class.AutoParam.val_init  [inherited]
```

6.9.4.4 val_list

```
skdaccess.framework.param_class.AutoParamListCycle.val_list
```

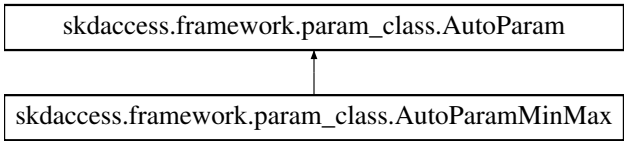
The documentation for this class was generated from the following file:

- framework/param_class.py

6.10 skdaccess.framework.param_class.AutoParamMinMax Class Reference

A tunable parameter with min and max ranges, perturbs to a random value in range.

Inheritance diagram for skdaccess.framework.param_class.AutoParamMinMax:



Public Member Functions

- def __init__ (self, val_init, val_min, val_max, decimals=0, extreme=0)

  Construct AutoParamMinMax object.
- def perturb (self)

  Peturb the paramter by choosing a random value between val_min and val_max.
- def reset (self)

  Reset to initial value.
- def __str__ (self)

  String representation of class.
- def __call__ (self)

  Retrieves current value of the parameter.

Public Attributes

- val
- val_init
- val_min
- val_max
- n
- n_max
- decimals

6.10.1 Detailed Description

A tunable parameter with min and max ranges, perturbs to a random value in range.

It can optionally choose either the min or the max after n perturbs

6.10.2 Constructor & Destructor Documentation

6.10.2.1 __init__()

```
def skdaccess.framework.param_class.AutoParamMinMax.__init__ (
              self,
              val_init,
              val_min,
              val_max,
              decimals = 0,
              extreme = 0 )
```

Construct AutoParamMinMax object.

Parameters

val_init	Initial value for parameter
val_min	Minimum value for param
val_max	Maximum value for parameter
decimals	Number of decimals to include in the random number
extreme	Either the maximum or minimum is chosen every extreme number of iterations. Using a value of one will be an extreme value every time. Using a value of zero will always choose a random value.

6.10.3 Member Function Documentation

6.10.3.1 __call__()

```
def skdaccess.framework.param_class.AutoParam.__call__ (
              self )  [inherited]
```

Retrieves current value of the parameter.

Returns

Current value of the parameter

6.10.3.2 __str__()

```
def skdaccess.framework.param_class.AutoParam.__str__ (
              self )  [inherited]
```

String representation of class.

Returns

String of current value

6.10.3.3 perturb()

```
def skdaccess.framework.param_class.AutoParamMinMax.perturb (
              self )
```

Peturb the paramter by choosing a random value between val_min and val_max.

Will choose a random number with precision specified by decimals. Will optionally pick the min or the max value after a specified number of perturb calls

6.10.3.4 reset()

```
def skdaccess.framework.param_class.AutoParamMinMax.reset (
              self )
```

Reset to initial value.

6.10.4 Member Data Documentation

6.10.4.1 decimals

```
skdaccess.framework.param_class.AutoParamMinMax.decimals
```

6.10.4.2 n

```
skdaccess.framework.param_class.AutoParamMinMax.n
```

6.10.4.3 n_max

```
skdaccess.framework.param_class.AutoParamMinMax.n_max
```

6.10.4.4 val

```
skdaccess.framework.param_class.AutoParamMinMax.val
```

6.10.4.5 val_init

```
skdaccess.framework.param_class.AutoParamMinMax.val_init
```

6.10.4.6 val_max

```
skdaccess.framework.param_class.AutoParamMinMax.val_max
```

6.10.4.7 val_min

```
skdaccess.framework.param_class.AutoParamMinMax.val_min
```

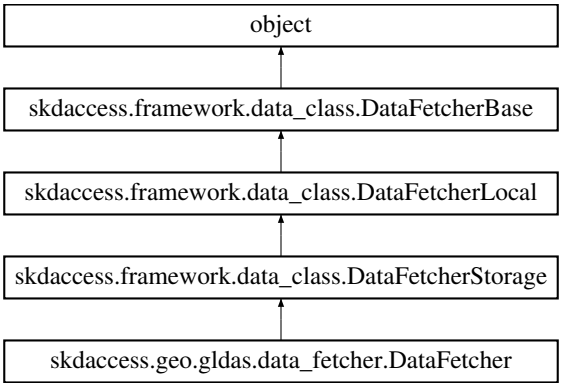
The documentation for this class was generated from the following file:

- framework/param_class.py

6.11 skdaccess.geo.gldas.DataFetcher Class Reference

Data Fetcher for GLDAS data.

Inheritance diagram for skdaccess.geo.gldas.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_date=None, end_date=None, resample=False)

  Construct a GLDAS Data Fetcher.
- def output (self)

  Create data wrapper of GLDAS data for specified geopoint.
- def downloadFullDataset (cls, out_file=None, use_file=None)

  Download GLDAS data.
- def __str__ (self)

  String representation of data fetcher.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- start_date
- end_date
- resample
- ap_paramList
- verbose

6.11.1 Detailed Description

Data Fetcher for GLDAS data.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 __init__()

```
def skdaccess.geo.gldas.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_date = None,
              end_date = None,
              resample = False )
```

Construct a GLDAS Data Fetcher.

Parameters

ap_paramList[geo_point]	Autolist of Geographic location tuples
start_date	Beginning date
end_date	Ending date
resample	Resample the data to daily resolution, leaving NaN's in days without data (Default True)

6.11.3 Member Function Documentation

6.11.3.1 __str__()

```
def skdaccess.geo.gldas.DataFetcher.__str__ (
              self )
```

String representation of data fetcher.

Returns

String listing the name and geopoint of data fetcher

6.11.3.2 downloadFullDataset()

```
def skdaccess.geo.gldas.DataFetcher.downloadFullDataset (
              cls,
              out_file = None,
              use_file = None )
```

Download GLDAS data.

Parameters

out_file	Output filename for parsed data
use_file	Directory of downloaded data. If None, data will be downloaded.

Returns

Absolute path of parsed data

6.11.3.3 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.11.3.4 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.11.3.5 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.11.3.6 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStorage.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.11.3.7 output()

```
def skdaccess.geo.gldas.DataFetcher.output (
              self )
```

Create data wrapper of GLDAS data for specified geopoint.

Returns

GLDAS Data Wrapper

6.11.3.8 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.11.3.9 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.11.3.10 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.11.3.11 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.11.3.12 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.11.4 Member Data Documentation

6.11.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.11.4.2 end_date

```
skdaccess.geo.gldas.DataFetcher.end_date
```

6.11.4.3 resample

```
skdaccess.geo.gldas.DataFetcher.resample
```

6.11.4.4 start_date

```
skdaccess.geo.gldas.DataFetcher.start_date
```

6.11.4.5 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

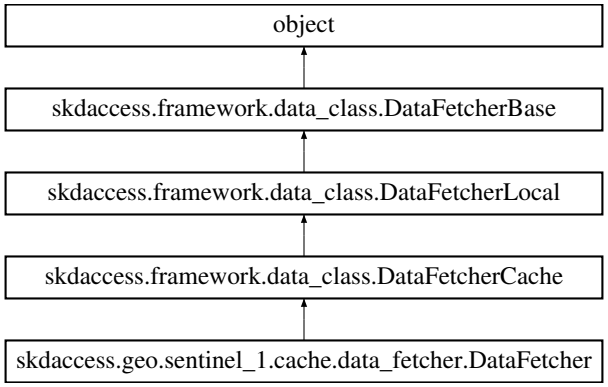
The documentation for this class was generated from the following file:

- geo/gldas/data_fetcher.py

6.12 skdaccess.geo.sentinel_1.cache.DataFetcher Class Reference

DataFetcher for retrieving Sentinel SLC data.

Inheritance diagram for skdaccess.geo.sentinel_1.cache.DataFetcher:



Public Member Functions

- def __init__ (self, url_list, satellite_url_list, username, password, swath, polarization='VV', local_paths=False, verbose=True)

  Initialize Sentinel Data Fetcher.
- def output (self)

  Generate data wrapper.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- url_list
- satellite_url_list
- swath
- username
- password
- polarization
- local_paths
- ap_paramList
- verbose

6.12.1 Detailed Description

DataFetcher for retrieving Sentinel SLC data.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 __init__()

```
def skdaccess.geo.sentinel_1.cache.DataFetcher.__init__ (
              self,
              url_list,
              satellite_url_list,
              username,
              password,
              swath,
              polarization = 'VV',
              local_paths = False,
              verbose = True )
```

Initialize Sentinel Data Fetcher.

Parameters

url_list	List of urls of SLC data
satellite_url_list	List of satellite urls
username	Username for downloading data
password	Password for downloading data
swath	Swath number (1, 2, or 3)
polarization	Polarization of data to retrieve
local_paths	locations are local paths, not urls
verbose	Print additional information

6.12.3 Member Function Documentation

6.12.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.12.3.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.12.3.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.12.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.12.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.12.3.6 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.12.3.7 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.12.3.8 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.12.3.9 output()

```
def skdaccess.geo.sentinel_1.cache.DataFetcher.output (
              self )
```

Generate data wrapper.

Returns

Sentinel SLC data in a data wrapper

6.12.3.10 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.12.3.11 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.12.3.12 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.12.3.13 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.12.3.14 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.12.4 Member Data Documentation

6.12.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.12.4.2 local_paths

```
skdaccess.geo.sentinel_1.cache.DataFetcher.local_paths
```

6.12.4.3 password

```
skdaccess.geo.sentinel_1.cache.DataFetcher.password
```

6.12.4.4 polarization

```
skdaccess.geo.sentinel_1.cache.DataFetcher.polarization
```

6.12.4.5 satellite_url_list

```
skdaccess.geo.sentinel_1.cache.DataFetcher.satellite_url_list
```

6.12.4.6 swath

```
skdaccess.geo.sentinel_1.cache.DataFetcher.swath
```

6.12.4.7 url_list

```
skdaccess.geo.sentinel_1.cache.DataFetcher.url_list
```

6.12.4.8 username

```
skdaccess.geo.sentinel_1.cache.DataFetcher.username
```

6.12.4.9 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

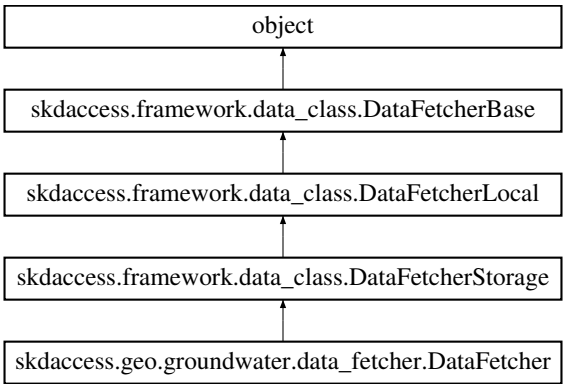
The documentation for this class was generated from the following file:

- geo/sentinel_1/cache/data_fetcher.py

6.13 skdaccess.geo.groundwater.DataFetcher Class Reference

Generates Data Wrappers of groundwater measurements taken in the US.

Inheritance diagram for skdaccess.geo.groundwater.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList=[ ], start_date=None, end_date=None, cutoff=0.75)

  Construct a Groundwater Data Fetcher.
- def output (self)

  Fetch Groundwater Data Wrapper.
- def __str__ (self)

  String representation of data fetcher.
- def getStationMetadata ()

  Retrieve metadata on groundwater wells.
- def downloadFullDataset (cls, out_file='gw.h5', use_file=None)

  Download and parse US groundwater data provided by USGS.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- start_date
- end_date
- ap_paramList
- cutoff
- verbose

6.13.1 Detailed Description

Generates Data Wrappers of groundwater measurements taken in the US.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 __init__()

```
def skdaccess.geo.groundwater.DataFetcher.__init__ (
              self,
              ap_paramList = [],
              start_date = None,
              end_date = None,
              cutoff = 0.75 )
```

Construct a Groundwater Data Fetcher.

Parameters

ap_paramList[LowerLat]	Autoparam Lower latitude
ap_paramList[UpperLat]	Autoparam Upper latitude
ap_paramList[LeftLon]	Autoparam Left longitude
ap_paramList[RightLon]	Autoparam Right longitude
start_date	Starting date (defualt: None)
end_date	Ending date (default: None)
cutoff	Required amount of data for each station

6.13.3 Member Function Documentation

6.13.3.1 __str__()

```
def skdaccess.geo.groundwater.DataFetcher.__str__ (
              self )
```

String representation of data fetcher.

Returns

string describing data fetcher

6.13.3.2 downloadFullDataset()

```
def skdaccess.geo.groundwater.DataFetcher.downloadFullDataset (
              cls,
              out_file = 'gw.h5',
              use_file = None )
```

Download and parse US groundwater data provided by USGS.

Parameters

out_file	Output filename for parsed data
use_file	Specify the directory where the data is. If None, the function will download the data

Returns

Absolute path of parsed data

6.13.3.3 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.13.3.4 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.13.3.5 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.13.3.6 getStationMetadata()

```
def skdaccess.geo.groundwater.DataFetcher.getStationMetadata ( )
```

Retrieve metadata on groundwater wells.

Returns

pandas dataframe with groundwater well information

6.13.3.7 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStorage.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.13.3.8 output()

```
def skdaccess.geo.groundwater.DataFetcher.output (
              self )
```

Fetch Groundwater Data Wrapper.

Returns

Groundwater Data Wrapper

6.13.3.9 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.13.3.10 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.13.3.11 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.13.3.12 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.13.3.13 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.13.4 Member Data Documentation

6.13.4.1 ap_paramList

```
skdaccess.geo.groundwater.DataFetcher.ap_paramList
```

6.13.4.2 cutoff

```
skdaccess.geo.groundwater.DataFetcher.cutoff
```

6.13.4.3 end_date

```
skdaccess.geo.groundwater.DataFetcher.end_date
```

6.13.4.4 start_date

```
skdaccess.geo.groundwater.DataFetcher.start_date
```

6.13.4.5 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

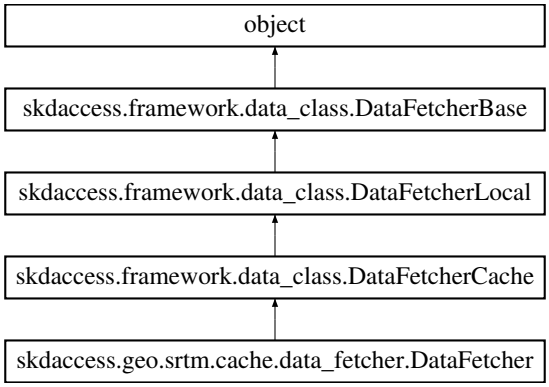
The documentation for this class was generated from the following file:

- geo/groundwater/data_fetcher.py

6.14 skdaccess.geo.srtm.cache.DataFetcher Class Reference

DataFetcher for retrieving data from the Shuttle Radar Topography Mission.

Inheritance diagram for skdaccess.geo.srtm.cache.DataFetcher:



Public Member Functions

- def __init__ (self, lat_tile_start, lat_tile_end, lon_tile_start, lon_tile_end, username, password, arcsecond_sampling=1, mask_water=True, store_geolocation_grids=False)

  Initialize Data Fetcher.
- def output (self)

  Generate SRTM data wrapper.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- lat_tile_start
- lat_tile_end
- lon_tile_start
- lon_tile_end
- username
- password
- arcsecond_sampling

  Determine the longitude and latitude of the lowerleft corner of the input filename.
- mask_water
- store_geolocation_grids
- ap_paramList
- verbose

6.14.1 Detailed Description

DataFetcher for retrieving data from the Shuttle Radar Topography Mission.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 __init__()

```
def skdaccess.geo.srtm.cache.DataFetcher.__init__ (
              self,
              lat_tile_start,
              lat_tile_end,
              lon_tile_start,
              lon_tile_end,
              username,
              password,
              arcsecond_sampling = 1,
              mask_water = True,
              store_geolocation_grids = False )
```

Initialize Data Fetcher.

Parameters

lat_tile_start	Latitude of the southwest corner of the starting tile
lat_tile_end	Latitude of the southwset corner of the last tile
lon_tile_start	Longitude of the southwest corner of the starting tile
lon_tile_end	Longitude of the southwest corner of the last tile
username	NASA Earth Data username
password	NASA Earth Data Password
arcsecond_sampling	Sample spacing of the SRTM data, either 1 arc- second or 3 arc-seconds
mask_water	True if the water bodies should be masked, false otherwise
store_geolocation_grids	Store grids of latitude and longitude in the metadata

6.14.3 Member Function Documentation

6.14.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.14.3.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.14.3.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.14.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.14.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.14.3.6 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.14.3.7 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.14.3.8 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.14.3.9 output()

```
def skdaccess.geo.srtm.cache.DataFetcher.output (
              self )
```

Generate SRTM data wrapper.

Returns

SRTM Image Wrapper

6.14.3.10 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.14.3.11 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.14.3.12 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.14.3.13 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.14.3.14 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.14.4 Member Data Documentation

6.14.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.14.4.2 arcsecond_sampling

```
skdaccess.geo.srtm.cache.DataFetcher.arcsecond_sampling
```

Determine the longitude and latitude of the lowerleft corner of the input filename.

Parameters

in_filename	Input SRTM filename

Returns

Latitude of southwest corner, Longitude of southwest corner

6.14.4.3 lat_tile_end

```
skdaccess.geo.srtm.cache.DataFetcher.lat_tile_end
```

6.14.4.4 lat_tile_start

```
skdaccess.geo.srtm.cache.DataFetcher.lat_tile_start
```

6.14.4.5 lon_tile_end

```
skdaccess.geo.srtm.cache.DataFetcher.lon_tile_end
```

6.14.4.6 lon_tile_start

```
skdaccess.geo.srtm.cache.DataFetcher.lon_tile_start
```

6.14.4.7 mask_water

```
skdaccess.geo.srtm.cache.DataFetcher.mask_water
```

6.14.4.8 password

```
skdaccess.geo.srtm.cache.DataFetcher.password
```

6.14.4.9 store_geolocation_grids

```
skdaccess.geo.srtm.cache.DataFetcher.store_geolocation_grids
```

6.14.4.10 username

```
skdaccess.geo.srtm.cache.DataFetcher.username
```

6.14.4.11 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

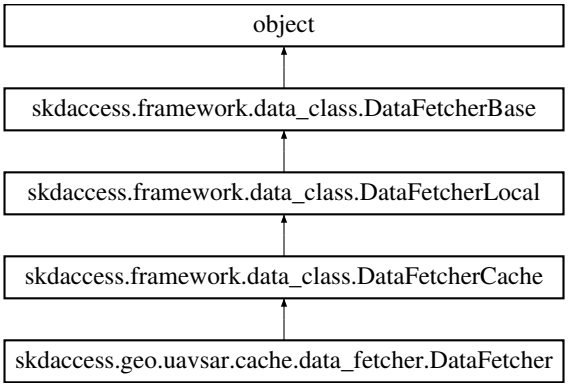
The documentation for this class was generated from the following file:

- geo/srtm/cache/data_fetcher.py

6.15 skdaccess.geo.uavsar.cache.DataFetcher Class Reference

Data Fetcher for UAVSAR data.

Inheritance diagram for skdaccess.geo.uavsar.cache.DataFetcher:



Public Member Functions

- def __init__ (self, slc_url_list, metadata_url_list, llh_url, memmap)

  Initialize UAVSAR data fetcher.
- def output (self)

  Output data as a data wrapper.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

Returns whether or not this data fetcher is multirun enabled.

- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- slc_url_list
- metadata_url_list
- llh_url
- memmap
- ap_paramList
- verbose

6.15.1 Detailed Description

Data Fetcher for UAVSAR data.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 __init__()

```
def skdaccess.geo.uavsar.cache.DataFetcher.__init__ (
              self,
              slc_url_list,
              metadata_url_list,
              llh_url,
              memmap )
```

Initialize UAVSAR data fetcher.

Parameters

slc_url_list	List of slc urls
metadata_url_list	List of metadata urls
llh_url	Latitude Longitude Height url
memmap	Open files using a memory map

6.15.3 Member Function Documentation

6.15.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.15.3.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.15.3.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.15.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.15.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.15.3.6 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.15.3.7 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.15.3.8 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.15.3.9 output()

```
def skdaccess.geo.uavsar.cache.DataFetcher.output (
              self )
```

Output data as a data wrapper.

Returns

Imagewrapper of data

6.15.3.10 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.15.3.11 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.15.3.12 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.15.3.13 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.15.3.14 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.15.4 Member Data Documentation

6.15.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.15.4.2 llh_url

```
skdaccess.geo.uavsar.cache.DataFetcher.llh_url
```

6.15.4.3 memmap

```
skdaccess.geo.uavsar.cache.DataFetcher.memmap
```

6.15.4.4 metadata_url_list

```
skdaccess.geo.uavsar.cache.DataFetcher.metadata_url_list
```

6.15.4.5 slc_url_list

```
skdaccess.geo.uavsar.cache.DataFetcher.slc_url_list
```

6.15.4.6 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

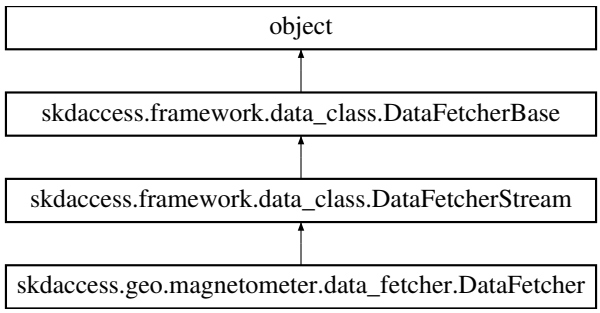
The documentation for this class was generated from the following file:

- geo/uavsar/cache/data_fetcher.py

6.16 skdaccess.geo.magnetometer.DataFetcher Class Reference

Data fetcher for USGS geomagnetic observatories.

Inheritance diagram for skdaccess.geo.magnetometer.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_time, end_time, interval='minute', channels=('X', 'Y', 'Z', 'F'), data_type='variation')

  Geomagnetism Data fetcher constructor.
- def output (self)

  Generate data wrapper for USGS geomagnetic data.
- def getDataMetadata ()

  Get data metadata.
- def retrieveOnlineData (self, data_specification)

  Method for downloading data into memory.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- start_time
- end_time
- interval
- channels
- data_type
- ap_paramList
- verbose

6.16.1 Detailed Description

Data fetcher for USGS geomagnetic observatories.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 __init__()

```
def skdaccess.geo.magnetometer.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_time,
              end_time,
              interval = 'minute',
              channels = ('X','Y','Z','F'),
              data_type = 'variation' )
```

Geomagnetism Data fetcher constructor.

Parameters

ap_paramList[AutoList]	AutoList of Observatory names
start_time	Starting time
end_time	Ending time
interval	Time resolution
channels	Data channels
data_type	= Data type

6.16.3 Member Function Documentation

6.16.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.16.3.2 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.16.3.3 getDataMetadata()

```
def skdaccess.geo.magnetometer.DataFetcher.getDataMetadata ( )
```

Get data metadata.

Returns

Pandas dataframe containing station latitude and longitude coordinates

6.16.3.4 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.16.3.5 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStream.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.16.3.6 output()

```
def skdaccess.geo.magnetometer.DataFetcher.output (
              self )
```

Generate data wrapper for USGS geomagnetic data.

Returns

geomagnetic data wrapper

6.16.3.7 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.16.3.8 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.16.3.9 retrieveOnlineData()

```
def skdaccess.framework.data_class.DataFetcherStream.retrieveOnlineData (
              self,
              data_specification )  [inherited]
```

Method for downloading data into memory.

Parameters

data_specification	Url list of data to be retrieved

Returns

Retrieved data

6.16.3.10 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.16.3.11 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.16.4 Member Data Documentation

6.16.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.16.4.2 channels

```
skdaccess.geo.magnetometer.DataFetcher.channels
```

6.16.4.3 data_type

```
skdaccess.geo.magnetometer.DataFetcher.data_type
```

6.16.4.4 end_time

```
skdaccess.geo.magnetometer.DataFetcher.end_time
```

6.16.4.5 interval

```
skdaccess.geo.magnetometer.DataFetcher.interval
```

6.16.4.6 start_time

```
skdaccess.geo.magnetometer.DataFetcher.start_time
```

6.16.4.7 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

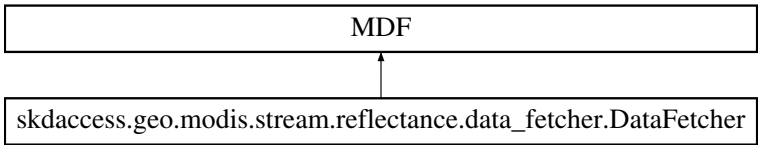
The documentation for this class was generated from the following file:

- geo/magnetometer/data_fetcher.py

6.17 skdaccess.geo.modis.stream.reflectance.DataFetcher Class Reference

Data fetcher for the modis surface reflectance product ('09', 1 km resolution)

Inheritance diagram for skdaccess.geo.modis.stream.reflectance.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_date, end_date, modis_platform='Terra', daynightboth='D', grid=None, bands=[1)

  Construct Data Fetcher for MODIS 1km surface reflectance.

6.17.1 Detailed Description

Data fetcher for the modis surface reflectance product ('09', 1 km resolution)

6.17.2 Constructor & Destructor Documentation

6.17.2.1 __init__()

```
def skdaccess.geo.modis.stream.reflectance.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_date,
              end_date,
              modis_platform = 'Terra',
              daynightboth = 'D',
              grid = None,
              bands = [1 )
```

Construct Data Fetcher for MODIS 1km surface reflectance.

Parameters

ap_paramList[lat]	Search latitude
ap_paramList[lon]	Search longitude
start_date	Starting date
end_date	Ending date
modis_platform	Paltform (Either "Terra" or "Aqua")
daynightboth	Use daytime data ('D'), nighttime data ('N') or both ('B')
grid	Further divide each image into a multiple grids of size (y,x)
bands	List of modis bands to retrieve

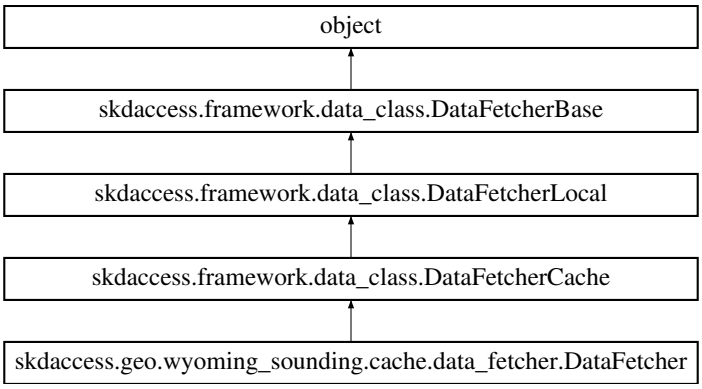
The documentation for this class was generated from the following file:

- geo/modis/stream/reflectance/data_fetcher.py

6.18 skdaccess.geo.wyoming_sounding.cache.DataFetcher Class Reference

DataFetcher for retrieving Wyoming Sounding data.

Inheritance diagram for skdaccess.geo.wyoming_sounding.cache.DataFetcher:



Public Member Functions

- def __init__ (self, station_number, year, month, day_start, day_end, start_hour=0, end_hour=12)

  Initialize Data Fetcher.
- def output (self)

  Generate data wrapper.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- station_number
- year_list
- month_list
- day_start
- day_end
- start_hour
- end_hour
- ap_paramList
- verbose

6.18.1 Detailed Description

DataFetcher for retrieving Wyoming Sounding data.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 __init__()

```
def skdaccess.geo.wyoming_sounding.cache.DataFetcher.__init__ (
              self,
              station_number,
              year,
              month,
              day_start,
              day_end,
              start_hour = 0,
              end_hour = 12 )
```

Initialize Data Fetcher.

Parameters

station_number	Station number
year	Input year
month	Input month (Integer for a single month, or a list of integers for multiple months)
day_start	First day of the month to include
day_end	Last day of the month to include
start_hour	Starting hour (may be either 0 or 12)
end_hour	Ending hour (may be either 0 or 12)

6.18.3 Member Function Documentation

6.18.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.18.3.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.18.3.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.18.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.18.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.18.3.6 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.18.3.7 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.18.3.8 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.18.3.9 output()

```
def skdaccess.geo.wyoming_sounding.cache.DataFetcher.output (
              self )
```

Generate data wrapper.

Returns

Wyoming sounding data in a data wrapper

6.18.3.10 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.18.3.11 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.18.3.12 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.18.3.13 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.18.3.14 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.18.4 Member Data Documentation

6.18.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.18.4.2 day_end

```
skdaccess.geo.wyoming_sounding.cache.DataFetcher.day_end
```

6.18.4.3 day_start

`skdaccess.geo.wyoming_sounding.cache.DataFetcher.day_start`

6.18.4.4 end_hour

`skdaccess.geo.wyoming_sounding.cache.DataFetcher.end_hour`

6.18.4.5 month_list

`skdaccess.geo.wyoming_sounding.cache.DataFetcher.month_list`

6.18.4.6 start_hour

`skdaccess.geo.wyoming_sounding.cache.DataFetcher.start_hour`

6.18.4.7 station_number

`skdaccess.geo.wyoming_sounding.cache.DataFetcher.station_number`

6.18.4.8 verbose

`skdaccess.framework.data_class.DataFetcherBase.verbose [inherited]`

6.18.4.9 year_list

`skdaccess.geo.wyoming_sounding.cache.DataFetcher.year_list`

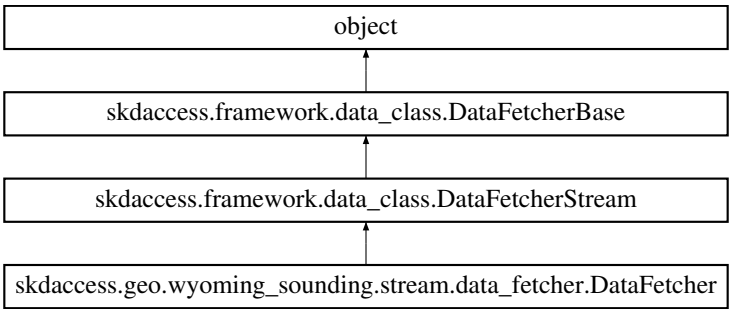
The documentation for this class was generated from the following file:

- geo/wyoming_sounding/cache/data_fetcher.py

6.19 skdaccess.geo.wyoming_sounding.stream.DataFetcher Class Reference

DataFetcher for retrieving Wyoming Sounding data.

Inheritance diagram for skdaccess.geo.wyoming_sounding.stream.DataFetcher:



Public Member Functions

- def __init__ (self, station_number, year, month, day_start, day_end, start_hour=0, end_hour=12)

  Initialize Data Fetcher.
- def output (self, shared_lock=None, shared_list=None)

  Generate data wrapper.
- def retrieveOnlineData (self, data_specification)

  Method for downloading data into memory.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def output (self)

  Output data wrapper.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- station_number
- year_list
- month_list
- day_start
- day_end
- start_hour
- end_hour
- ap_paramList
- verbose

6.19.1 Detailed Description

DataFetcher for retrieving Wyoming Sounding data.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 __init__()

```
def skdaccess.geo.wyoming_sounding.stream.DataFetcher.__init__ (
              self,
              station_number,
              year,
              month,
              day_start,
              day_end,
              start_hour = 0,
              end_hour = 12 )
```

Initialize Data Fetcher.

Parameters

station_number	Station number
year	Input year
month	Input month (Integer for a single month, or a list of integers for multiple months)
day_start	First day of the month to include
day_end	Last day of the month to include
start_hour	Starting hour (may be either 0 or 12)
end_hour	Ending hour (may be either 0 or 12)

6.19.3 Member Function Documentation

6.19.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.19.3.2 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.19.3.3 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.19.3.4 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStream.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.19.3.5 output() [1/2]

```
def skdaccess.geo.wyoming_sounding.stream.DataFetcher.output (
              self,
              shared_lock = None,
              shared_list = None )
```

Generate data wrapper.

Returns

Wyoming sounding data in a data wrapper

6.19.3.6 output() [2/2]

```
def skdaccess.framework.data_class.DataFetcherBase.output (
              self )  [inherited]
```

Output data wrapper.

Returns

Datawrapper

6.19.3.7 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.19.3.8 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.19.3.9 retrieveOnlineData()

```
def skdaccess.framework.data_class.DataFetcherStream.retrieveOnlineData (
              self,
              data_specification )  [inherited]
```

Method for downloading data into memory.

Parameters

data_specification	Url list of data to be retrieved

Returns

Retrieved data

6.19.3.10 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.19.3.11 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.19.4 Member Data Documentation

6.19.4.1 ap_paramList

`skdaccess.framework.data_class.DataFetcherBase.ap_paramList [inherited]`

6.19.4.2 day_end

`skdaccess.geo.wyoming_sounding.stream.DataFetcher.day_end`

6.19.4.3 day_start

`skdaccess.geo.wyoming_sounding.stream.DataFetcher.day_start`

6.19.4.4 end_hour

`skdaccess.geo.wyoming_sounding.stream.DataFetcher.end_hour`

6.19.4.5 month_list

`skdaccess.geo.wyoming_sounding.stream.DataFetcher.month_list`

6.19.4.6 start_hour

`skdaccess.geo.wyoming_sounding.stream.DataFetcher.start_hour`

6.19.4.7 station_number

`skdaccess.geo.wyoming_sounding.stream.DataFetcher.station_number`

6.19.4.8 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

6.19.4.9 year_list

```
skdaccess.geo.wyoming_sounding.stream.DataFetcher.year_list
```

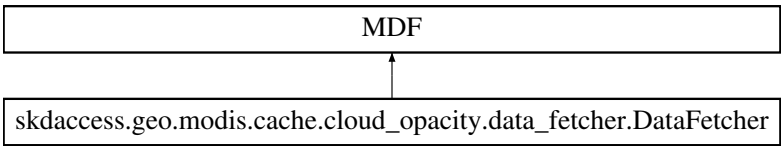
The documentation for this class was generated from the following file:

- geo/wyoming_sounding/stream/data_fetcher.py

6.20 skdaccess.geo.modis.cache.cloud_opacity.DataFetcher Class Reference

Data Fetcher for MODIS Cloud Opacity.

Inheritance diagram for skdaccess.geo.modis.cache.cloud_opacity.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_date, end_date, modis_platform='Terra', daynightboth='D', grid=None)
  Construct Data Fetcher object for MODIS cloud Opacity data.

6.20.1 Detailed Description

Data Fetcher for MODIS Cloud Opacity.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 __init__()

```
def skdaccess.geo.modis.cache.cloud_opacity.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_date,
              end_date,
              modis_platform = 'Terra',
              daynightboth = 'D',
              grid = None )
```

Construct Data Fetcher object for MODIS cloud Opacity data.

Parameters

ap_paramList[lat]	Search latitude
ap_paramList[lon]	Search longitude
start_date	Starting date
end_date	Ending date
modis_platform	Paltform (Either "Terra" or "Aqua")
daynightboth	Use daytime data ('D'), nighttime data ('N') or both ('B')
grid	Further divide each image into a multiple grids of size (y,x)

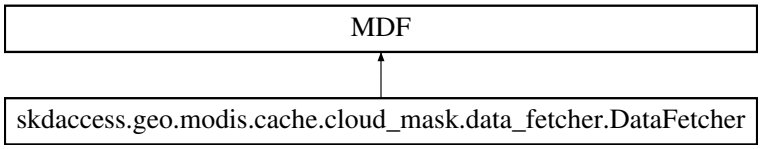
The documentation for this class was generated from the following file:

- geo/modis/cache/cloud_opacity/data_fetcher.py

6.21 skdaccess.geo.modis.cache.cloud_mask.DataFetcher Class Reference

Data Fetcher for MODIS Cloud Mask.

Inheritance diagram for skdaccess.geo.modis.cache.cloud_mask.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_date, end_date, modis_platform='Terra', daynightboth='D', grid=None)

  Construct Data Fetcher for MODIS cloud mask data.

6.21.1 Detailed Description

Data Fetcher for MODIS Cloud Mask.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 __init__()

```
def skdaccess.geo.modis.cache.cloud_mask.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_date,
              end_date,
              modis_platform = 'Terra',
              daynightboth = 'D',
              grid = None )
```

Construct Data Fetcher for MODIS cloud mask data.

Parameters

ap_paramList[lat]	Search latitude
ap_paramList[lon]	Search longitude
start_date	Starting date
end_date	Ending date
modis_platform	Paltform (Either "Terra" or "Aqua")
daynightboth	Use daytime data ('D'), nighttime data ('N') or both ('B')
grid	Further divide each image into a multiple grids of size (y,x)

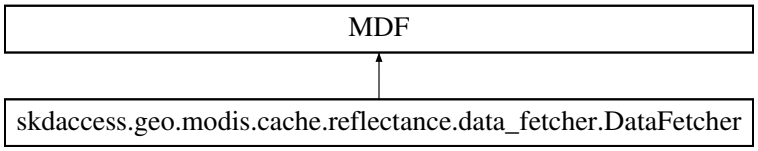
The documentation for this class was generated from the following file:

- geo/modis/cache/cloud_mask/data_fetcher.py

6.22 skdaccess.geo.modis.cache.reflectance.DataFetcher Class Reference

Data fetcher for the modis surface reflectance product ('09', 1 km resolution)

Inheritance diagram for skdaccess.geo.modis.cache.reflectance.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_date, end_date, modis_platform='Terra', daynightboth='D', grid=None, bands=[1])

  Construct Data Fetcher for MODIS 1km surface reflectance.

6.22.1 Detailed Description

Data fetcher for the modis surface reflectance product ('09', 1 km resolution)

6.22.2 Constructor & Destructor Documentation

6.22.2.1 __init__()

```
def skdaccess.geo.modis.cache.reflectance.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_date,
              end_date,
              modis_platform = 'Terra',
              daynightboth = 'D',
              grid = None,
              bands = [1 )
```

Construct Data Fetcher for MODIS 1km surface reflectance.

Parameters

ap_paramList[lat]	Search latitude
ap_paramList[lon]	Search longitude
start_date	Starting date
end_date	Ending date
modis_platform	Paltform (Either "Terra" or "Aqua")
daynightboth	Use daytime data ('D'), nighttime data ('N') or both ('B')
grid	Further divide each image into a multiple grids of size (y,x)
bands	List of modis bands to retrieve

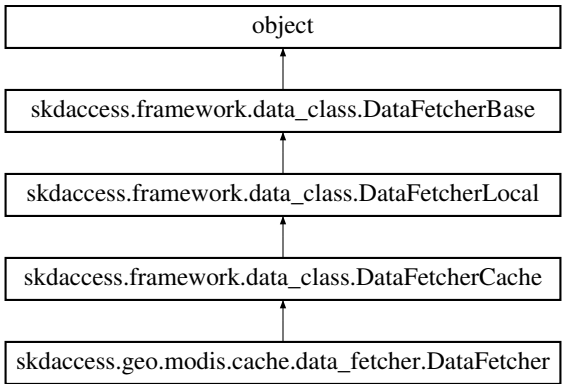
The documentation for this class was generated from the following file:

- geo/modis/cache/reflectance/data_fetcher.py

6.23 skdaccess.geo.modis.cache.DataFetcher Class Reference

Data Fetcher for MODIS data.

Inheritance diagram for skdaccess.geo.modis.cache.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, modis_platform, modis_id, variable_list, start_date, end_date, daynightboth='D', grid=None, grid_fill=np.nan, use_long_name=False)

  Construct Data Fetcher object.
- def find_data (self, fileid_list, file_object)

  Finds files previously downloaded files associated with fileids.
- def cacheData (self, data_specification)

  Download MODIS data.
- def output (self)

  Generate data wrapper.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.

- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- modis_id
- variable_list
- start_date
- end_date
- daynightboth
- grid
- grid_fill
- use_long_name
- modis_platform
- modis_identifier
- ap_paramList
- verbose

6.23.1 Detailed Description

Data Fetcher for MODIS data.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 __init__()

```
def skdaccess.geo.modis.cache.DataFetcher.__init__ (
              self,
              ap_paramList,
              modis_platform,
              modis_id,
              variable_list,
              start_date,
              end_date,
              daynightboth = 'D',
              grid = None,
              grid_fill = np.nan,
              use_long_name = False )
```

Construct Data Fetcher object.

Parameters

ap_paramList[lat]	Search latitude
ap_paramList[lon]	Search longitude
modis_platform	Platform (Either "Terra" or "Aqua")
modis_id	Product string (e.g. '06_L2')
variable_list	List of variables to fetch
start_date	Starting date
end_date	Ending date
daynightboth	Use daytime data ('D'), nighttime data ('N') or both ('B')
grid	Further divide each image into a multiple grids of size (y,x)
grid_fill	Fill value to use when creating gridded data
use_long_name	Use long names for metadata instead of variable name

6.23.3 Member Function Documentation

6.23.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.23.3.2 cacheData() [1/2]

```
def skdaccess.geo.modis.cache.DataFetcher.cacheData (
              self,
              data_specification )
```

Download MODIS data.

Parameters

data_specification	List of file IDs to cache

6.23.3.3 cacheData() [2/2]

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.23.3.4 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.23.3.5 find_data()

```
def skdaccess.geo.modis.cache.DataFetcher.find_data (
              self,
              fileid_list,
              file_object )
```

Finds files previously downloaded files associated with fileids.

Parameters

fileid_list	List of file id's
file_object	File object to read from

Returns

Pandas series of file locaitons indexed by file id

6.23.3.6 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.23.3.7 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.23.3.8 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.23.3.9 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.23.3.10 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.23.3.11 output()

```
def skdaccess.geo.modis.cache.DataFetcher.output (
              self )
```

Generate data wrapper.

Returns

data wrapper of MODIS data

6.23.3.12 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.23.3.13 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.23.3.14 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.23.3.15 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.23.3.16 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.23.4 Member Data Documentation

6.23.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.23.4.2 daynightboth

skdaccess.geo.modis.cache.DataFetcher.daynightboth

6.23.4.3 end_date

skdaccess.geo.modis.cache.DataFetcher.end_date

6.23.4.4 grid

skdaccess.geo.modis.cache.DataFetcher.grid

6.23.4.5 grid_fill

skdaccess.geo.modis.cache.DataFetcher.grid_fill

6.23.4.6 modis_id

skdaccess.geo.modis.cache.DataFetcher.modis_id

6.23.4.7 modis_identifier

skdaccess.geo.modis.cache.DataFetcher.modis_identifier

6.23.4.8 modis_platform

skdaccess.geo.modis.cache.DataFetcher.modis_platform

6.23.4.9 start_date

```
skdaccess.geo.modis.cache.DataFetcher.start_date
```

6.23.4.10 use_long_name

```
skdaccess.geo.modis.cache.DataFetcher.use_long_name
```

6.23.4.11 variable_list

```
skdaccess.geo.modis.cache.DataFetcher.variable_list
```

6.23.4.12 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

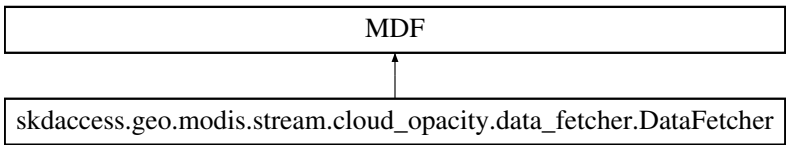
The documentation for this class was generated from the following file:

- geo/modis/cache/data_fetcher.py

6.24 skdaccess.geo.modis.stream.cloud_opacity.DataFetcher Class Reference

Data Fetcher for MODIS Cloud Opacity.

Inheritance diagram for skdaccess.geo.modis.stream.cloud_opacity.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_date, end_date, modis_platform='Terra', daynightboth='D', grid=None)
  Construct Data Fetcher object for MODIS cloud Opacity data.

6.24.1 Detailed Description

Data Fetcher for MODIS Cloud Opacity.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 __init__()

```
def skdaccess.geo.modis.stream.cloud_opacity.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_date,
              end_date,
              modis_platform = 'Terra',
              daynightboth = 'D',
              grid = None )
```

Construct Data Fetcher object for MODIS cloud Opacity data.

Parameters

ap_paramList[lat]	Search latitude
ap_paramList[lon]	Search longitude
start_date	Starting date
end_date	Ending date
modis_platform	Paltform (Either "Terra" or "Aqua")
daynightboth	Use daytime data ('D'), nighttime data ('N') or both ('B')
grid	Further divide each image into a multiple grids of size (y,x)

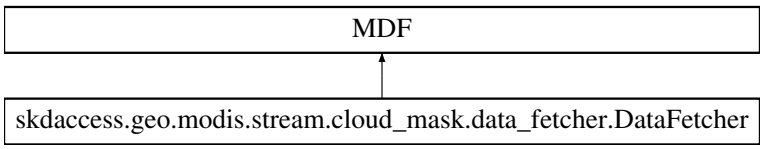
The documentation for this class was generated from the following file:

- geo/modis/stream/cloud_opacity/data_fetcher.py

6.25 skdaccess.geo.modis.stream.cloud_mask.DataFetcher Class Reference

Data Fetcher for MODIS Cloud Mask.

Inheritance diagram for skdaccess.geo.modis.stream.cloud_mask.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_date, end_date, modis_platform='Terra', daynightboth='D', grid=None)
  Construct Data Fetcher for MODIS cloud mask data.

6.25.1 Detailed Description

Data Fetcher for MODIS Cloud Mask.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 __init__()

```
def skdaccess.geo.modis.stream.cloud_mask.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_date,
              end_date,
              modis_platform = 'Terra',
              daynightboth = 'D',
              grid = None )
```

Construct Data Fetcher for MODIS cloud mask data.

Parameters

ap_paramList[lat]	Search latitude
ap_paramList[lon]	Search longitude
start_date	Starting date
end_date	Ending date
modis_platform	Paltform (Either "Terra" or "Aqua")
daynightboth	Use daytime data ('D'), nighttime data ('N') or both ('B')
grid	Further divide each image into a multiple grids of size (y,x)

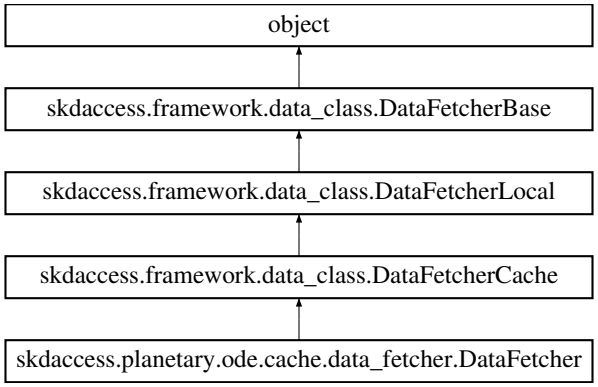
The documentation for this class was generated from the following file:

- geo/modis/stream/cloud_mask/data_fetcher.py

6.26 skdaccess.planetary.ode.cache.DataFetcher Class Reference

Data Fetcher from the Orbital Data Explorer (ODE)

Inheritance diagram for skdaccess.planetary.ode.cache.DataFetcher:



Public Member Functions

- def __init__ (self, target, mission, instrument, product_type, western_lon=None, eastern_lon=None, min_lat=None, max_lat=None, min_ob_time='', max_ob_time='', product_id='', file_name=' ∗', number_product_limit=10, result_offset_number=0, remove_ndv=True)
- def output (self)

  Generate data wrapper from ODE data.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- target
- mission
- instrument
- product_type
- western_lon
- eastern_lon
- min_lat
- max_lat
- min_ob_time
- max_ob_time
- product_id
- file_name
- number_product_limit
- result_offset_number
- remove_ndv
- ap_paramList
- verbose

6.26.1 Detailed Description

Data Fetcher from the Orbital Data Explorer (ODE)

6.26.2 Constructor & Destructor Documentation

6.26.2.1 __init__()

```
def skdaccess.planetary.ode.cache.DataFetcher.__init__ (
              self,
              target,
              mission,
              instrument,
              product_type,
              western_lon = None,
              eastern_lon = None,
              min_lat = None,
              max_lat = None,
              min_ob_time = '',
              max_ob_time = '',
              product_id = '',
              file_name = '*',
              number_product_limit = 10,
              result_offset_number = 0,
              remove_ndv = True )
```

6.26.3 Member Function Documentation

6.26.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.26.3.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.26.3.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.26.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.26.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.26.3.6 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.26.3.7 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.26.3.8 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.26.3.9 output()

```
def skdaccess.planetary.ode.cache.DataFetcher.output (
              self )
```

Generate data wrapper from ODE data.

6.26.3.10 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.26.3.11 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.26.3.12 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.26.3.13 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.26.3.14 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.26.4 Member Data Documentation

6.26.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.26.4.2 eastern_lon

```
skdaccess.planetary.ode.cache.DataFetcher.eastern_lon
```

6.26.4.3 file_name

```
skdaccess.planetary.ode.cache.DataFetcher.file_name
```

6.26.4.4 instrument

```
skdaccess.planetary.ode.cache.DataFetcher.instrument
```

6.26.4.5 max_lat

```
skdaccess.planetary.ode.cache.DataFetcher.max_lat
```

6.26.4.6 max_ob_time

```
skdaccess.planetary.ode.cache.DataFetcher.max_ob_time
```

6.26.4.7 min_lat

```
skdaccess.planetary.ode.cache.DataFetcher.min_lat
```

6.26.4.8 min_ob_time

```
skdaccess.planetary.ode.cache.DataFetcher.min_ob_time
```

6.26.4.9 mission

```
skdaccess.planetary.ode.cache.DataFetcher.mission
```

6.26.4.10 number_product_limit

```
skdaccess.planetary.ode.cache.DataFetcher.number_product_limit
```

6.26.4.11 product_id

```
skdaccess.planetary.ode.cache.DataFetcher.product_id
```

6.26.4.12 product_type

```
skdaccess.planetary.ode.cache.DataFetcher.product_type
```

6.26.4.13 remove_ndv

```
skdaccess.planetary.ode.cache.DataFetcher.remove_ndv
```

6.26.4.14 result_offset_number

```
skdaccess.planetary.ode.cache.DataFetcher.result_offset_number
```

6.26.4.15 target

```
skdaccess.planetary.ode.cache.DataFetcher.target
```

6.26.4.16 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

6.26.4.17 western_lon

`skdaccess.planetary.ode.cache.DataFetcher.western_lon`

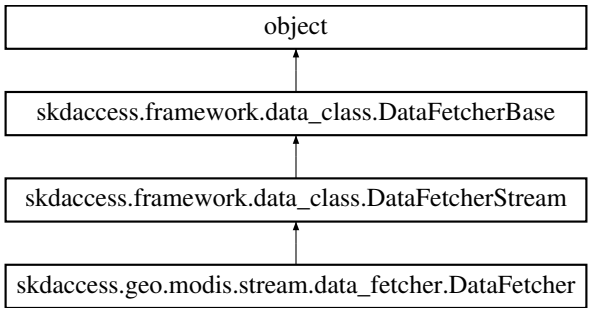
The documentation for this class was generated from the following file:

- planetary/ode/cache/data_fetcher.py

6.27 skdaccess.geo.modis.stream.DataFetcher Class Reference

Data Fetcher for MODIS data.

Inheritance diagram for skdaccess.geo.modis.stream.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, modis_platform, modis_id, variable_list, start_date, end_date, daynightboth='D', grid=None, grid_fill=np.nan, use_long_name=False)

  Construct Data Fetcher object.
- def output (self)

  Generate data wrapper.
- def retrieveOnlineData (self, data_specification)

  Method for downloading data into memory.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- modis_id
- variable_list
- start_date
- end_date
- daynightboth
- grid
- grid_fill
- use_long_name
- modis_platform
- modis_identifier
- ap_paramList
- verbose

6.27.1 Detailed Description

Data Fetcher for MODIS data.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 __init__()

```
def skdaccess.geo.modis.stream.DataFetcher.__init__ (
              self,
              ap_paramList,
              modis_platform,
              modis_id,
              variable_list,
              start_date,
              end_date,
              daynightboth = 'D',
              grid = None,
              grid_fill = np.nan,
              use_long_name = False )
```

Construct Data Fetcher object.

Parameters

ap_paramList[lat]	Search latitude
ap_paramList[lon]	Search longitude
modis_platform	Platform (Either "Terra" or "Aqua")
modis_id	Product string (e.g. '06_L2')
variable_list	List of variables to fetch

Parameters

start_date	Starting date
end_date	Ending date
daynightboth	Use daytime data ('D'), nighttime data ('N') or both ('B')
grid	Further divide each image into a multiple grids of size (y,x)
grid_fill	Fill value to use when creating gridded data
use_long_name	Use long names for metadata instead of variable name

6.27.3 Member Function Documentation

6.27.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.27.3.2 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.27.3.3 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.27.3.4 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStream.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.27.3.5 output()

```
def skdaccess.geo.modis.stream.DataFetcher.output (
              self )
```

Generate data wrapper.

Returns

data wrapper of MODIS data

6.27.3.6 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.27.3.7 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.27.3.8 retrieveOnlineData()

```
def skdaccess.framework.data_class.DataFetcherStream.retrieveOnlineData (
              self,
              data_specification )  [inherited]
```

Method for downloading data into memory.

Parameters

data_specification	Url list of data to be retrieved

Returns

Retrieved data

6.27.3.9 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.27.3.10 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.27.4 Member Data Documentation

6.27.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.27.4.2 daynightboth

```
skdaccess.geo.modis.stream.DataFetcher.daynightboth
```

6.27.4.3 end_date

```
skdaccess.geo.modis.stream.DataFetcher.end_date
```

6.27.4.4 grid

```
skdaccess.geo.modis.stream.DataFetcher.grid
```

6.27.4.5 grid_fill

```
skdaccess.geo.modis.stream.DataFetcher.grid_fill
```

6.27.4.6 modis_id

```
skdaccess.geo.modis.stream.DataFetcher.modis_id
```

6.27.4.7 modis_identifier

```
skdaccess.geo.modis.stream.DataFetcher.modis_identifier
```

6.27.4.8 modis_platform

```
skdaccess.geo.modis.stream.DataFetcher.modis_platform
```

6.27.4.9 start_date

```
skdaccess.geo.modis.stream.DataFetcher.start_date
```

6.27.4.10 use_long_name

```
skdaccess.geo.modis.stream.DataFetcher.use_long_name
```

6.27.4.11 variable_list

```
skdaccess.geo.modis.stream.DataFetcher.variable_list
```

6.27.4.12 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

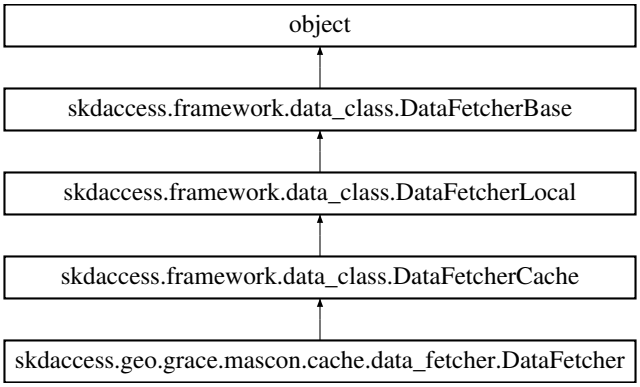
The documentation for this class was generated from the following file:

- geo/modis/stream/data_fetcher.py

6.28 skdaccess.geo.grace.mascon.cache.DataFetcher Class Reference

Data Fetcher for GRACE mascon data.

Inheritance diagram for skdaccess.geo.grace.mascon.cache.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_date=None, end_date=None)

  Construct a GRACE mascon Data Fetcher.
- def output (self)

  Create a datawrapper containing GRACE mascon data.
- def getMasconPlacement (self)

  Retrieve mascon placement data.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- start_date
- end_date
- mascon_url
- scale_factor_url
- mascon_placement_url
- ap_paramList
- verbose

6.28.1 Detailed Description

Data Fetcher for GRACE mascon data.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 __init__()

```
def skdaccess.geo.grace.mascon.cache.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_date = None,
              end_date = None )
```

Construct a GRACE mascon Data Fetcher.

Parameters

ap_paramList[geo_point]	AutoList of geographic location tuples (lat,lon)
start_date	Beginning date
end_date	Ending date

6.28.3 Member Function Documentation

6.28.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.28.3.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.28.3.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.28.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.28.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.28.3.6 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.28.3.7 getMasconPlacement()

```
def skdaccess.geo.grace.mascon.cache.DataFetcher.getMasconPlacement (
              self )
```

Retrieve mascon placement data.

Returns

Mascon data, Mascon metadata

6.28.3.8 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.28.3.9 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.28.3.10 output()

```
def skdaccess.geo.grace.mascon.cache.DataFetcher.output (
              self )
```

Create a datawrapper containing GRACE mascon data.

Returns

Table Datawrapper containing Mascon GRACE data

6.28.3.11 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.28.3.12 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.28.3.13 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.28.3.14 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.28.3.15 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.28.4 Member Data Documentation

6.28.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.28.4.2 end_date

```
skdaccess.geo.grace.mascon.cache.DataFetcher.end_date
```

6.28.4.3 mascon_placement_url

```
skdaccess.geo.grace.mascon.cache.DataFetcher.mascon_placement_url
```

6.28.4.4 mascon_url

```
skdaccess.geo.grace.mascon.cache.DataFetcher.mascon_url
```

6.28.4.5 scale_factor_url

```
skdaccess.geo.grace.mascon.cache.DataFetcher.scale_factor_url
```

6.28.4.6 start_date

```
skdaccess.geo.grace.mascon.cache.DataFetcher.start_date
```

6.28.4.7 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

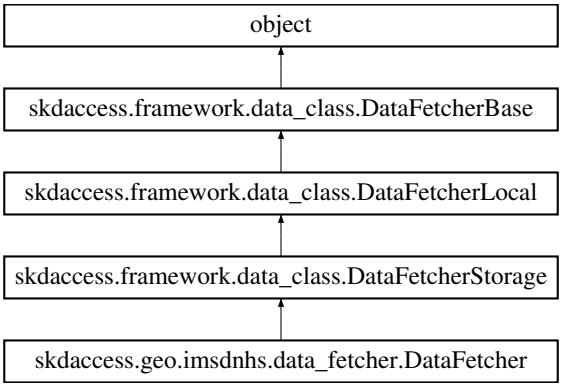
The documentation for this class was generated from the following file:

- geo/grace/mascon/cache/data_fetcher.py

6.29 skdaccess.geo.imsdnhs.DataFetcher Class Reference

Fetches data for the Interactive Multisensor Snow and Ice Mapping System Daily Northern Hemisphere Snow and Ice Analysis.

Inheritance diagram for skdaccess.geo.imsdnhs.DataFetcher:



Public Member Functions

- def __init__ (self, coordinate_dict, start_date, end_date)

  Intializes the Data Fetcher.
- def output (self)

  Fetch snow coverage data for coordinates.
- def downloadFullDataset (cls, out_file, use_file=None)

  Abstract function used to download full data set.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- coordinate_dict
- start_date
- end_date
- ap_paramList
- verbose

6.29.1 Detailed Description

Fetches data for the Interactive Multisensor Snow and Ice Mapping System Daily Northern Hemisphere Snow and Ice Analysis.

6.29.2 Constructor & Destructor Documentation

6.29.2.1 __init__()

```
def skdaccess.geo.imsdnhs.DataFetcher.__init__ (
              self,
              coordinate_dict,
              start_date,
              end_date )
```

Intializes the Data Fetcher.

Parameters

coordinate_dict	Dictionary of locations where the names are the keys and the items are lists containing the latitude and longitude are the values
start_date	Starting date
end_date	Ending date

6.29.3 Member Function Documentation

6.29.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.29.3.2 downloadFullDataset()

```
def skdaccess.framework.data_class.DataFetcherStorage.downloadFullDataset (
              cls,
              out_file,
              use_file = None )  [inherited]
```

Abstract function used to download full data set.

Parameters

out_file	output file name
use_file	Use previously downloaded data

Returns

Absolute path of parsed data

6.29.3.3 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.29.3.4 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.29.3.5 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.29.3.6 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStorage.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.29.3.7 output()

```
def skdaccess.geo.imsdnhs.DataFetcher.output (
              self )
```

Fetch snow coverage data for coordinates.

Returns

Data wrapper for snow coverage

6.29.3.8 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.29.3.9 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.29.3.10 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.29.3.11 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.29.3.12 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.29.4 Member Data Documentation

6.29.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.29.4.2 coordinate_dict

```
skdaccess.geo.imsdnhs.DataFetcher.coordinate_dict
```

6.29.4.3 end_date

```
skdaccess.geo.imsdnhs.DataFetcher.end_date
```

6.29.4.4 start_date

```
skdaccess.geo.imsdnhs.DataFetcher.start_date
```

6.29.4.5 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

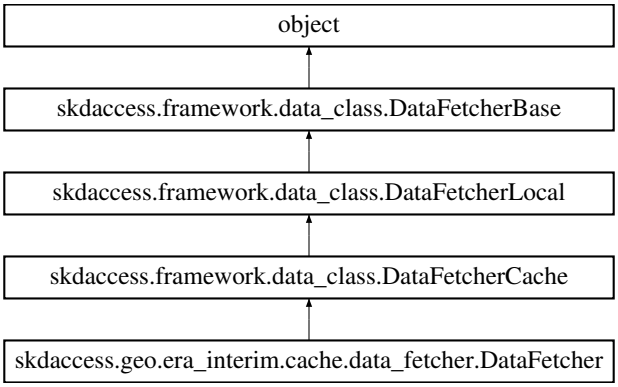
The documentation for this class was generated from the following file:

- geo/imsdnhs/data_fetcher.py

6.30 skdaccess.geo.era_interim.cache.DataFetcher Class Reference

DataFetcher for retrieving ERA-I data.

Inheritance diagram for skdaccess.geo.era_interim.cache.DataFetcher:



Public Member Functions

- def __init__ (self, date_list, data_names, username, password)

  Initialize Data Fetcher.
- def output (self)

  Generate data wrapper.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- date_list
- data_names
- username
- password
- ap_paramList
- verbose

6.30.1 Detailed Description

DataFetcher for retrieving ERA-I data.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 __init__()

```
def skdaccess.geo.era_interim.cache.DataFetcher.__init__ (
              self,
              date_list,
              data_names,
              username,
              password )
```

Initialize Data Fetcher.

Parameters

date_list	list of dates
data_names	list of data names
username	UCAR username
password	UCAR password

6.30.3 Member Function Documentation

6.30.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.30.3.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.30.3.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.30.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.30.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.30.3.6 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.30.3.7 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.30.3.8 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.30.3.9 output()

```
def skdaccess.geo.era_interim.cache.DataFetcher.output (
              self )
```

Generate data wrapper.

Returns

Era-I weather in a data wrapper

6.30.3.10 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.30.3.11 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.30.3.12 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.30.3.13 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.30.3.14 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.30.4 Member Data Documentation

6.30.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.30.4.2 data_names

```
skdaccess.geo.era_interim.cache.DataFetcher.data_names
```

6.30.4.3 date_list

```
skdaccess.geo.era_interim.cache.DataFetcher.date_list
```

6.30.4.4 password

```
skdaccess.geo.era_interim.cache.DataFetcher.password
```

6.30.4.5 username

```
skdaccess.geo.era_interim.cache.DataFetcher.username
```

6.30.4.6 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

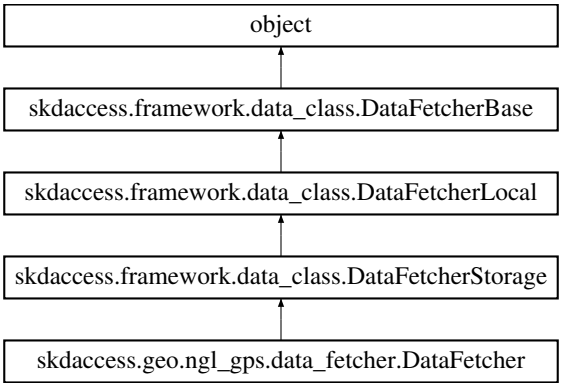
The documentation for this class was generated from the following file:

- geo/era_interim/cache/data_fetcher.py

6.31 skdaccess.geo.ngl_gps.DataFetcher Class Reference

Data fetcher for GPS data from Neveda Geodetic Laboratory.

Inheritance diagram for skdaccess.geo.ngl_gps.DataFetcher:



Public Member Functions

- def __init__ (self, start_date, end_date, lat_range, lon_range, mdyratio=0.7, data_type='ngl_gps')

  Consctruct NGL data fetcher.
- def getStationMetadata ()

  Get station metadata.
- def getAntennaLogs ()

  Retrieve information about antenna changes.
- def output (self)

  Construct NGL GPS data wrapper.
- def downloadFullDataset (cls, out_file, use_file=None)

  Abstract function used to download full data set.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- start_date
- end_date
- lat_range
- lon_range
- mdyratio
- data_type
- ap_paramList
- verbose

6.31.1 Detailed Description

Data fetcher for GPS data from Neveda Geodetic Laboratory.

6.31.2 Constructor & Destructor Documentation

6.31.2.1 __init__()

```
def skdaccess.geo.ngl_gps.DataFetcher.__init__ (
              self,
              start_date,
              end_date,
              lat_range,
              lon_range,
              mdyratio = 0.7,
              data_type = 'ngl_gps' )
```

Consctruct NGL data fetcher.

Parameters

start_date	Starting date (string: '2002-01-01')
end_date	Ending date (string: '2015-01-01')
lat_range	Tuple containing latitude range
lon_range	Tuple containing longitude range
mdyratio	Choose stations whose ratio of valid/total is greater than mdyratio
data_type	Either 24 hour product ('ngl_gps') or 5 minute product ('ngl_5min')

6.31.3 Member Function Documentation

6.31.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.31.3.2 downloadFullDataset()

```
def skdaccess.framework.data_class.DataFetcherStorage.downloadFullDataset (
              cls,
              out_file,
              use_file = None )  [inherited]
```

Abstract function used to download full data set.

Parameters

out_file	output file name
use_file	Use previously downloaded data

Returns

Absolute path of parsed data

6.31.3.3 getAntennaLogs()

```
def skdaccess.geo.ngl_gps.DataFetcher.getAntennaLogs ( )
```

Retrieve information about antenna changes.

Returns

dictionary of antenna changes

6.31.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.31.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.31.3.6 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.31.3.7 getStationMetadata()

```
def skdaccess.geo.ngl_gps.DataFetcher.getStationMetadata ( )
```

Get station metadata.

Returns

data frame of station metadata

6.31.3.8 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStorage.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.31.3.9 output()

```
def skdaccess.geo.ngl_gps.DataFetcher.output (
              self )
```

Construct NGL GPS data wrapper.

Returns

NGL GPS data wrapper

6.31.3.10 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.31.3.11 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.31.3.12 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.31.3.13 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.31.3.14 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.31.4 Member Data Documentation

6.31.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.31.4.2 data_type

```
skdaccess.geo.ngl_gps.DataFetcher.data_type
```

6.31.4.3 end_date

`skdaccess.geo.ngl_gps.DataFetcher.end_date`

6.31.4.4 lat_range

`skdaccess.geo.ngl_gps.DataFetcher.lat_range`

6.31.4.5 lon_range

`skdaccess.geo.ngl_gps.DataFetcher.lon_range`

6.31.4.6 mdyratio

`skdaccess.geo.ngl_gps.DataFetcher.mdyratio`

6.31.4.7 start_date

`skdaccess.geo.ngl_gps.DataFetcher.start_date`

6.31.4.8 verbose

`skdaccess.framework.data_class.DataFetcherBase.verbose [inherited]`

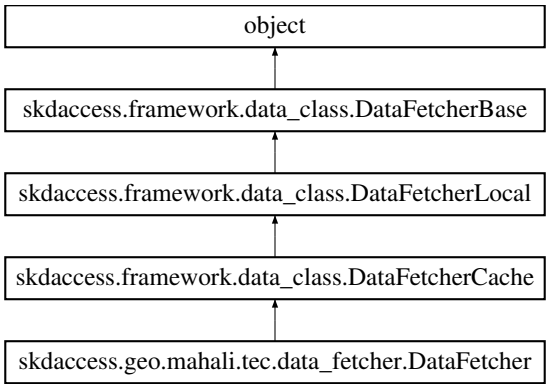
The documentation for this class was generated from the following file:

- geo/ngl_gps/data_fetcher.py

6.32 skdaccess.geo.mahali.tec.DataFetcher Class Reference

Data Fetcher for Mahali Data.

Inheritance diagram for skdaccess.geo.mahali.tec.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList=[ ], start_date=None, end_date=None)

  Initialize Mahali Data Fetcher.
- def output (self)

  Generate data wrapper for Mahali tec data.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

Retrieve skdaccess configuration.

- def writeConfig (conf)

  Write config to disk.

- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- start_date
- end_date
- date_range
- ap_paramList
- verbose

6.32.1 Detailed Description

Data Fetcher for Mahali Data.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 __init__()

```
def skdaccess.geo.mahali.tec.DataFetcher.__init__ (
              self,
              ap_paramList = [],
              start_date = None,
              end_date = None )
```

Initialize Mahali Data Fetcher.

Parameters

ap_paramList[stations]	Autolist of stations (Defaults to all stations)
start_date	Starting date for seelcting data (Defaults to beginning of available data)
end_date	Ending date for selecting data (Defaults to end of available data)

6.32.3 Member Function Documentation

6.32.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.32.3.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.32.3.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.32.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.32.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.32.3.6 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.32.3.7 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.32.3.8 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.32.3.9 output()

```
def skdaccess.geo.mahali.tec.DataFetcher.output (
              self )
```

Generate data wrapper for Mahali tec data.

Returns

Mahali data wrapper

6.32.3.10 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.32.3.11 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.32.3.12 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.32.3.13 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.32.3.14 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.32.4 Member Data Documentation

6.32.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.32.4.2 date_range

```
skdaccess.geo.mahali.tec.DataFetcher.date_range
```

6.32.4.3 end_date

```
skdaccess.geo.mahali.tec.DataFetcher.end_date
```

6.32.4.4 start_date

```
skdaccess.geo.mahali.tec.DataFetcher.start_date
```

6.32.4.5 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

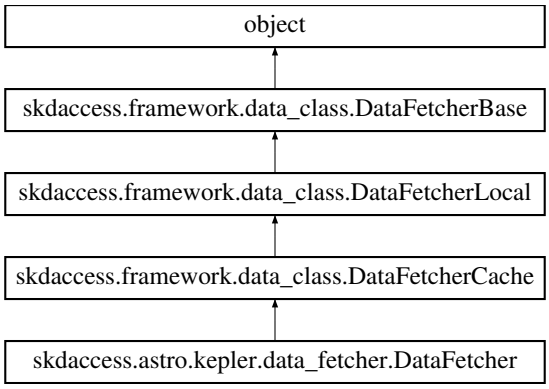
The documentation for this class was generated from the following file:

- geo/mahali/tec/data_fetcher.py

6.33 skdaccess.astro.kepler.DataFetcher Class Reference

Data Fetcher for Kepler light curve data.

Inheritance diagram for skdaccess.astro.kepler.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, quarter_list=None)

  Initialize Kepler Data Fetcher.
- def downloadKeplerData (self, kid_list)

  Download and parse Kepler data for a list of kepler id's.
- def cacheData (self, data_specification)

  Cache Kepler data locally.
- def output (self)

  Output kepler data wrapper.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.

- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- quarter_list
- ap_paramList
- verbose

6.33.1 Detailed Description

Data Fetcher for Kepler light curve data.

6.33.2 Constructor & Destructor Documentation

6.33.2.1 __init__()

```
def skdaccess.astro.kepler.DataFetcher.__init__ (
              self,
              ap_paramList,
              quarter_list = None )
```

Initialize Kepler Data Fetcher.

Parameters

ap_paramList[kepler_id_list]	List of kepler id's
quarter_list	List of quarters (0-17) (default: all quarters)

6.33.3 Member Function Documentation

6.33.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.33.3.2 cacheData() [1/2]

```
def skdaccess.astro.kepler.DataFetcher.cacheData (
              self,
              data_specification )
```

Cache Kepler data locally.

Parameters

data_specification	List of kepler IDs

6.33.3.3 cacheData() [2/2]

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.33.3.4 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.33.3.5 downloadKeplerData()

```
def skdaccess.astro.kepler.DataFetcher.downloadKeplerData (
              self,
              kid_list )
```

Download and parse Kepler data for a list of kepler id's.

Parameters

kid_list	List of Kepler ID's to download

Returns

dictionary of kepler data

6.33.3.6 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.33.3.7 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.33.3.8 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.33.3.9 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.33.3.10 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.33.3.11 output()

```
def skdaccess.astro.kepler.DataFetcher.output (
              self )
```

Output kepler data wrapper.

Returns

DataWrapper

6.33.3.12 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.33.3.13 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.33.3.14 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.33.3.15 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.33.3.16 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.33.4 Member Data Documentation

6.33.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.33.4.2 quarter_list

```
skdaccess.astro.kepler.DataFetcher.quarter_list
```

6.33.4.3 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

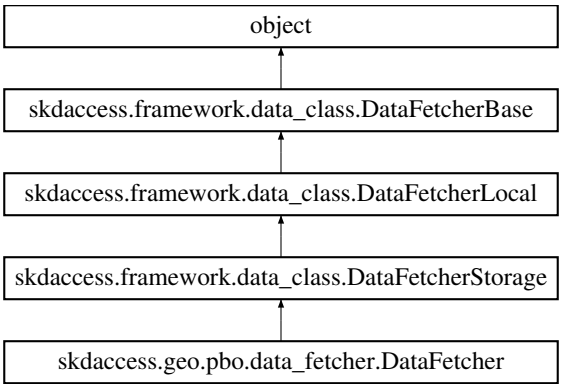
The documentation for this class was generated from the following file:

- astro/kepler/data_fetcher.py

6.34 skdaccess.geo.pbo.DataFetcher Class Reference

Data fetcher for PBO GPS data.

Inheritance diagram for skdaccess.geo.pbo.DataFetcher:



Public Member Functions

- def __init__ (self, start_time, end_time, ap_paramList, mdyratio=.5, default_columns=['dN', dE, dU, default_error_columns=['Sn', Se, Su, use_progress_bar=True, index_date_only=True)

  Initialize a DataFetcher.
- def setStationList (self, station_list)

  Set the list of stations to use.
- def getInfo (self)

  Get information about the stations and geo_point.
- def output (self)

  Generate PBO Data Wrapper.
- def __str__ (self)

  print the parameter values
- def getStationMetadata (data_frame=False)

  Read in the metadata and convert to dictionary.
- def getAntennaLogs ()

  Get antenna logs.
- def downloadFullDataset (cls, out_file='pbo_data.h5', use_file=None)

  Download and parse data from the Plate Boundary Observatory.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

Set all parameters to initial value.

- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- station_list
- default_columns
- default_error_columns
- use_progress_bar
- index_date_only
- antenna_info
- meta_data
- ap_paramList
- verbose

6.34.1 Detailed Description

Data fetcher for PBO GPS data.

6.34.2 Constructor & Destructor Documentation

6.34.2.1 __init__()

```
def skdaccess.geo.pbo.DataFetcher.__init__ (
              self,
              start_time,
              end_time,
              ap_paramList,
              mdyratio = .5,
              default_columns = ['dN',
              dE,
              dU,
              default_error_columns = ['Sn',
              Se,
              Su,
              use_progress_bar = True,
              index_date_only = True )
```

Initialize a DataFetcher.

Parameters

start_time	String of starting date in the form of "2005-01-01"
end_time	String of ending date in the form of "2014-12-31"
ap_paramList[lat_range]	AutoList, Latitude range used to select stabilization sites
ap_paramList[lon_range]	AutoList, Longitude range used to select stabilization sites
mdyratio	Only keep stations that have mdyratio of data in the specified time range
default_columns	Default columns to process
default_error_columns	Default error columns to process
use_progress_bar	Use a progress bar when loading data
index_date_only	Create a index using date only (no hour information)

6.34.3 Member Function Documentation

6.34.3.1 __str__()

```
def skdaccess.geo.pbo.DataFetcher.__str__ (
              self )
```

print the parameter values

Returns

String representation of Data Fetcher

6.34.3.2 downloadFullDataset()

```
def skdaccess.geo.pbo.DataFetcher.downloadFullDataset (
              cls,
              out_file = 'pbo_data.h5',
              use_file = None )
```

Download and parse data from the Plate Boundary Observatory.

Parameters

out_file	Output filename for parsed data
use_file	Use already downloaded data. If None, data will be downloaded.

Returns

Absolute path of parsed data

6.34.3.3 getAntennaLogs()

```
def skdaccess.geo.pbo.DataFetcher.getAntennaLogs ( )
```

Get antenna logs.

Returns

dictionary of data frames containing antenna logs

6.34.3.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.34.3.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.34.3.6 getInfo()

```
def skdaccess.geo.pbo.DataFetcher.getInfo (
              self )
```

Get information about the stations and geo_point.

Returns

tuple containing station list and geo_point

6.34.3.7 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.34.3.8 getStationMetadata()

```
def skdaccess.geo.pbo.DataFetcher.getStationMetadata (
              data_frame = False )
```

Read in the metadata and convert to dictionary.

Returns

dictionary of PBO metadata

6.34.3.9 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStorage.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.34.3.10 output()

```
def skdaccess.geo.pbo.DataFetcher.output (
              self )
```

Generate PBO Data Wrapper.

Returns

PBO Data Wrapper

6.34.3.11 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.34.3.12 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.34.3.13 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.34.3.14 setStationList()

```
def skdaccess.geo.pbo.DataFetcher.setStationList (
              self,
              station_list )
```

Set the list of stations to use.

Parameters

station_list	List of stations to fetch

6.34.3.15 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.34.3.16 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.34.4 Member Data Documentation

6.34.4.1 antenna_info

```
skdaccess.geo.pbo.DataFetcher.antenna_info
```

6.34.4.2 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.34.4.3 default_columns

```
skdaccess.geo.pbo.DataFetcher.default_columns
```

6.34.4.4 default_error_columns

```
skdaccess.geo.pbo.DataFetcher.default_error_columns
```

6.34.4.5 index_date_only

```
skdaccess.geo.pbo.DataFetcher.index_date_only
```

6.34.4.6 meta_data

```
skdaccess.geo.pbo.DataFetcher.meta_data
```

6.34.4.7 station_list

```
skdaccess.geo.pbo.DataFetcher.station_list
```

6.34.4.8 use_progress_bar

```
skdaccess.geo.pbo.DataFetcher.use_progress_bar
```

6.34.4.9 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

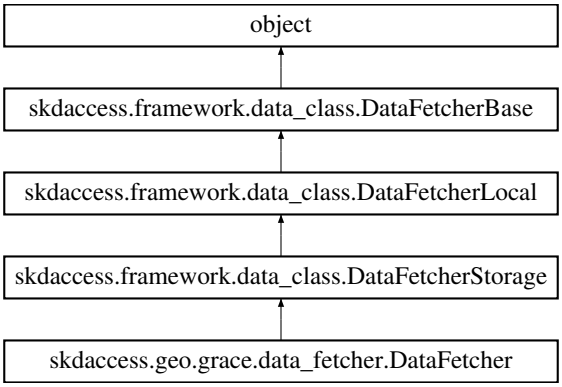
The documentation for this class was generated from the following file:

- geo/pbo/data_fetcher.py

6.35 skdaccess.geo.grace.DataFetcher Class Reference

Data Fetcher for GRACE data.

Inheritance diagram for skdaccess.geo.grace.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList, start_date=None, end_date=None)

  Construct a Grace Data Fetcher.
- def output (self)

  Create data wrapper of grace data for specified geopoints.
- def __str__ (self)

  String representation of data fetcher.
- def downloadFullDataset (cls, out_file='grace.h5', use_file=None)

  Download and parse data from the Gravity Recovery and Climate Experiment.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- start_date
- end_date
- ap_paramList
- verbose

6.35.1 Detailed Description

Data Fetcher for GRACE data.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 __init__()

```
def skdaccess.geo.grace.DataFetcher.__init__ (
              self,
              ap_paramList,
              start_date = None,
              end_date = None )
```

Construct a Grace Data Fetcher.

Parameters

ap_paramList[geo_point]	AutoList of geographic location tuples (lat,lon)
start_date	Beginning date
end_date	Ending date

6.35.3 Member Function Documentation

6.35.3.1 __str__()

```
def skdaccess.geo.grace.DataFetcher.__str__ (
              self )
```

String representation of data fetcher.

Returns

String listing the name and geopoint of data fetcher

6.35.3.2 downloadFullDataset()

```
def skdaccess.geo.grace.DataFetcher.downloadFullDataset (
              cls,
              out_file = 'grace.h5',
              use_file = None )
```

Download and parse data from the Gravity Recovery and Climate Experiment.

Parameters

out_file	Output filename for parsed data
use_file	Directory of already downloaded data. If None, data will be downloaded.

Returns

Absolute path of parsed data

6.35.3.3 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.35.3.4 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.35.3.5 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.35.3.6 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStorage.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.35.3.7 output()

```
def skdaccess.geo.grace.DataFetcher.output (
              self )
```

Create data wrapper of grace data for specified geopoints.

Returns

Grace Data Wrapper

6.35.3.8 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.35.3.9 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.35.3.10 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.35.3.11 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.35.3.12 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.35.4 Member Data Documentation

6.35.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.35.4.2 end_date

```
skdaccess.geo.grace.DataFetcher.end_date
```

6.35.4.3 start_date

```
skdaccess.geo.grace.DataFetcher.start_date
```

6.35.4.4 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

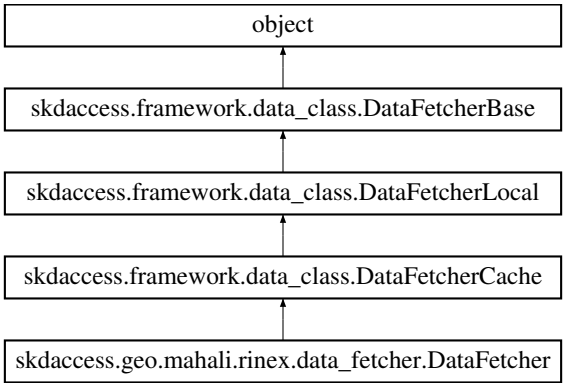
The documentation for this class was generated from the following file:

- geo/grace/data_fetcher.py

6.36 skdaccess.geo.mahali.rinex.DataFetcher Class Reference

Data Fetcher for Mahali Data.

Inheritance diagram for skdaccess.geo.mahali.rinex.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList=[ ], start_date=None, end_date=None, generate_links=False)

  Initialize Mahali Data Fetcher.
- def cacheData (self)

  Downloads all needed data.
- def output (self)

  Generate data wrapper for Mahali data.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- start_date
- end_date
- date_range
- generate_links
- ap_paramList
- verbose

6.36.1 Detailed Description

Data Fetcher for Mahali Data.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 __init__()

```
def skdaccess.geo.mahali.rinex.DataFetcher.__init__ (
              self,
              ap_paramList = [],
              start_date = None,
              end_date = None,
              generate_links = False )
```

Initialize Mahali Data Fetcher.

Parameters

ap_paramList[stations]	Autolist of stations (Defaults to all stations)
start_date	Starting date for seelcting data (Defaults to beginning of available data)
end_date	Ending date for selecting data (Defaults to end of available data)
generate_links	Generate links to data instead of downloading data

6.36.3 Member Function Documentation

6.36.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.36.3.2 cacheData() [1/2]

```
def skdaccess.geo.mahali.rinex.DataFetcher.cacheData (
              self )
```

Downloads all needed data.

Called by output().

6.36.3.3 cacheData() [2/2]

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.36.3.4 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.36.3.5 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.36.3.6 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.36.3.7 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.36.3.8 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.36.3.9 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.36.3.10 output()

```
def skdaccess.geo.mahali.rinex.DataFetcher.output (
              self )
```

Generate data wrapper for Mahali data.

Returns

Mahali data wrapper

6.36.3.11 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.36.3.12 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.36.3.13 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.36.3.14 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.36.3.15 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.36.4 Member Data Documentation

6.36.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.36.4.2 date_range

```
skdaccess.geo.mahali.rinex.DataFetcher.date_range
```

6.36.4.3 end_date

```
skdaccess.geo.mahali.rinex.DataFetcher.end_date
```

6.36.4.4 generate_links

```
skdaccess.geo.mahali.rinex.DataFetcher.generate_links
```

6.36.4.5 start_date

```
skdaccess.geo.mahali.rinex.DataFetcher.start_date
```

6.36.4.6 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

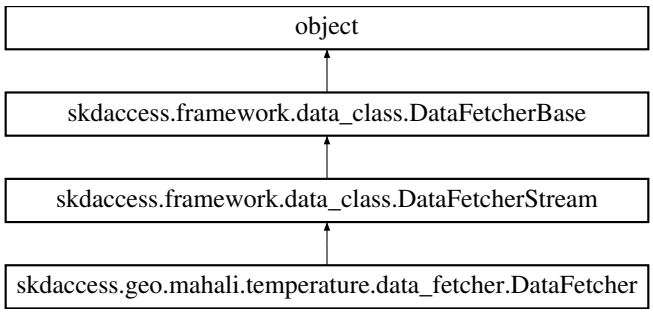
The documentation for this class was generated from the following file:

- geo/mahali/rinex/data_fetcher.py

6.37 skdaccess.geo.mahali.temperature.DataFetcher Class Reference

Data Fetcher for Mahali temperature data.

Inheritance diagram for skdaccess.geo.mahali.temperature.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList=[ ], start_date=None, end_date=None)

  Initialize Mahali temperature data fetcher.
- def retrieveOnlineData (self, data_specification)

  Load data in from a remote source.
- def output (self)

  Generate data wrapper for Mahali temperatures.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- start_date
- end_date
- ap_paramList
- verbose

6.37.1 Detailed Description

Data Fetcher for Mahali temperature data.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 __init__()

```
def skdaccess.geo.mahali.temperature.DataFetcher.__init__ (
              self,
              ap_paramList = [],
              start_date = None,
              end_date = None )
```

Initialize Mahali temperature data fetcher.

Parameters

ap_paramList[stations]	Autolist of stations (Defaults to all stations)
start_date	Starting date for seelcting data (Defaults to beginning of available data)
end_date	Ending date for selecting data (Defaults to end of available data)

6.37.3 Member Function Documentation

6.37.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.37.3.2 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.37.3.3 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.37.3.4 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStream.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.37.3.5 output()

```
def skdaccess.geo.mahali.temperature.DataFetcher.output (
              self )
```

Generate data wrapper for Mahali temperatures.

Returns

Mahali temperature data wrapper

6.37.3.6 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.37.3.7 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.37.3.8 retrieveOnlineData()

```
def skdaccess.geo.mahali.temperature.DataFetcher.retrieveOnlineData (
              self,
              data_specification )
```

Load data in from a remote source.

Parameters

data_specification	Pandas dataframe containing the columns 'station', 'date', and 'filename'

Returns

Ordered dictionary for each station (key) which cointains a pandas data frame of the temperature

6.37.3.9 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.37.3.10 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.37.4 Member Data Documentation

6.37.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.37.4.2 end_date

```
skdaccess.geo.mahali.temperature.DataFetcher.end_date
```

6.37.4.3 start_date

```
skdaccess.geo.mahali.temperature.DataFetcher.start_date
```

6.37.4.4 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

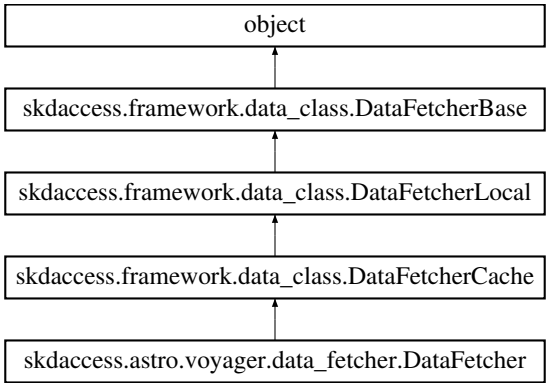
The documentation for this class was generated from the following file:

- geo/mahali/temperature/data_fetcher.py

6.38 skdaccess.astro.voyager.DataFetcher Class Reference

Data Fetcher for Mahali temperature data.

Inheritance diagram for skdaccess.astro.voyager.DataFetcher:



Public Member Functions

- def __init__ (self, start_year, end_year, spacecraft='both')

  Initialize Voyager data fetcher.
- def generateURL (self, spacecraft, in_year)

  Generate url for voyager data.
- def parseVoyagerData (self, spacecraft, in_filename)

  Parse Voyager Data.
- def parseVoyagerMetadata (self, in_file)

  Parse voyager metadata.
- def getMetadataFiles (self)

  Get path to metadata file.
- def output (self)

  Generate data wrapper.
- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

  Download and store specified data to local disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- year_list
- spacecraft_list
- field_names
- field_widths
- base_url
- ap_paramList
- verbose

6.38.1 Detailed Description

Data Fetcher for Mahali temperature data.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 __init__()

```
def skdaccess.astro.voyager.DataFetcher.__init__ (
              self,
              start_year,
              end_year,
              spacecraft = 'both' )
```

Initialize Voyager data fetcher.

Parameters

start_year	Starting year
end_year	Ending year
spacecraft	Which spaceraft to use (voyager1, voyager2, or both).

6.38.3 Member Function Documentation

6.38.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.38.3.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )  [inherited]
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.38.3.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )  [inherited]
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.38.3.4 generateURL()

```
def skdaccess.astro.voyager.DataFetcher.generateURL (
              self,
              spacecraft,
              in_year )
```

Generate url for voyager data.

Parameters

spacecraft	Voyager spacecraft (vy1 or vy2)
in_year	Input year (or 'metadata')

Returns

Url of data location

6.38.3.5 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.38.3.6 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.38.3.7 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )  [inherited]
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.38.3.8 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.38.3.9 getMetadataFiles()

```
def skdaccess.astro.voyager.DataFetcher.getMetadataFiles (
              self )
```

Get path to metadata file.

Metadata will download if necessary

Returns

List containing file path(s) for the metadata

6.38.3.10 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.38.3.11 output()

```
def skdaccess.astro.voyager.DataFetcher.output (
              self )
```

Generate data wrapper.

Returns

data wrapper of voyager data

6.38.3.12 parseVoyagerData()

```
def skdaccess.astro.voyager.DataFetcher.parseVoyagerData (
              self,
              spacecraft,
              in_filename )
```

Parse Voyager Data.

Parameters

spacecraft	Voyager spacecraft (vy1 or vy2)
in_filename	Input voyager data filename

Returns

Pandas Dataframe of Voyager data

6.38.3.13 parseVoyagerMetadata()

```
def skdaccess.astro.voyager.DataFetcher.parseVoyagerMetadata (
              self,
              in_file )
```

Parse voyager metadata.

Parameters

in_file	Input filename

Returns

Dictionary containing metadata

6.38.3.14 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.38.3.15 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.38.3.16 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.38.3.17 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

$*$*args*	Arguments to pass to print
$**$*kwargs*	Keyword arguments to pass to print

6.38.3.18 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.38.4 Member Data Documentation

6.38.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.38.4.2 base_url

```
skdaccess.astro.voyager.DataFetcher.base_url
```

6.38.4.3 field_names

```
skdaccess.astro.voyager.DataFetcher.field_names
```

6.38.4.4 field_widths

```
skdaccess.astro.voyager.DataFetcher.field_widths
```

6.38.4.5 spacecraft_list

```
skdaccess.astro.voyager.DataFetcher.spacecraft_list
```

6.38.4.6 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

6.38.4.7 year_list

```
skdaccess.astro.voyager.DataFetcher.year_list
```

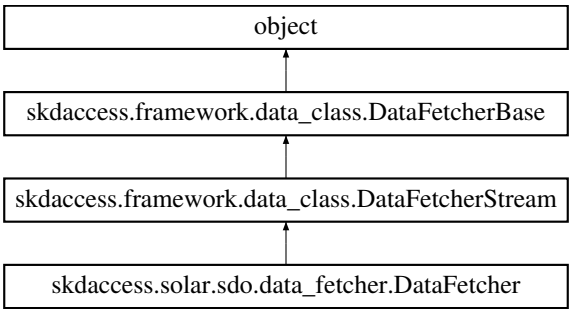
The documentation for this class was generated from the following file:

- astro/voyager/data_fetcher.py

6.39 skdaccess.solar.sdo.DataFetcher Class Reference

Data Fetcher for Mahali temperature data.

Inheritance diagram for skdaccess.solar.sdo.DataFetcher:



Public Member Functions

- def __init__ (self, ap_paramList)

  Initialize Solar Dynamics Observatory.
- def output (self)

  Generate data wrapper.
- def retrieveOnlineData (self, data_specification)

  Method for downloading data into memory.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- ap_paramList
- verbose

6.39.1 Detailed Description

Data Fetcher for Mahali temperature data.

6.39.2 Constructor & Destructor Documentation

6.39.2.1 __init__()

```
def skdaccess.solar.sdo.DataFetcher.__init__ (
              self,
              ap_paramList )
```

Initialize Solar Dynamics Observatory.

Parameters

ap_paramList[url_list]	Autolist of URLS to access

6.39.3 Member Function Documentation

6.39.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.39.3.2 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.39.3.3 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.39.3.4 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStream.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.39.3.5 output()

```
def skdaccess.solar.sdo.DataFetcher.output (
              self )
```

Generate data wrapper.

Returns

data wrapper of SDO data

6.39.3.6 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.39.3.7 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.39.3.8 retrieveOnlineData()

```
def skdaccess.framework.data_class.DataFetcherStream.retrieveOnlineData (
              self,
              data_specification )  [inherited]
```

Method for downloading data into memory.

Parameters

data_specification	Url list of data to be retrieved

Returns

Retrieved data

6.39.3.9 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.39.3.10 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.39.4 Member Data Documentation

6.39.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.39.4.2 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

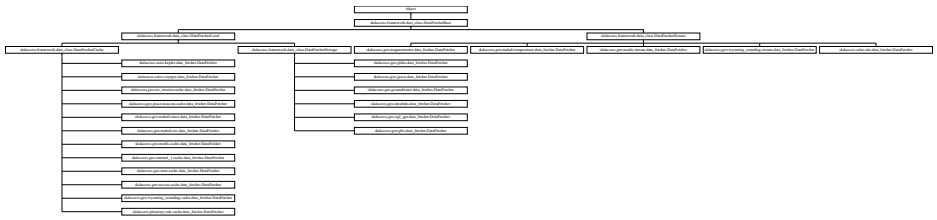
The documentation for this class was generated from the following file:

- solar/sdo/data_fetcher.py

6.40 skdaccess.framework.data_class.DataFetcherBase Class Reference

Base class for all data fetchers.

Inheritance diagram for skdaccess.framework.data_class.DataFetcherBase:



Public Member Functions

- def __init__ (self, ap_paramList=[ ], verbose=False)

  Initialize data fetcher with parameter list.
- def output (self)

  Output data wrapper.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- ap_paramList
- verbose

6.40.1 Detailed Description

Base class for all data fetchers.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 __init__()

```
def skdaccess.framework.data_class.DataFetcherBase.__init__ (
              self,
              ap_paramList = [],
              verbose = False )
```

Initialize data fetcher with parameter list.

Parameters

ap_paramList	List of parameters
verbose	Output extra information

6.40.3 Member Function Documentation

6.40.3.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )
```

Generate string description.

6.40.3.2 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.40.3.3 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.40.3.4 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherBase.multirun_enabled (
              self )
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.40.3.5 output()

```
def skdaccess.framework.data_class.DataFetcherBase.output (
              self )
```

Output data wrapper.

Returns

Datawrapper

6.40.3.6 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )
```

Perturb parameters.

6.40.3.7 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )
```

Set all parameters to initial value.

6.40.3.8 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.40.3.9 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.40.4 Member Data Documentation

6.40.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList
```

6.40.4.2 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose
```

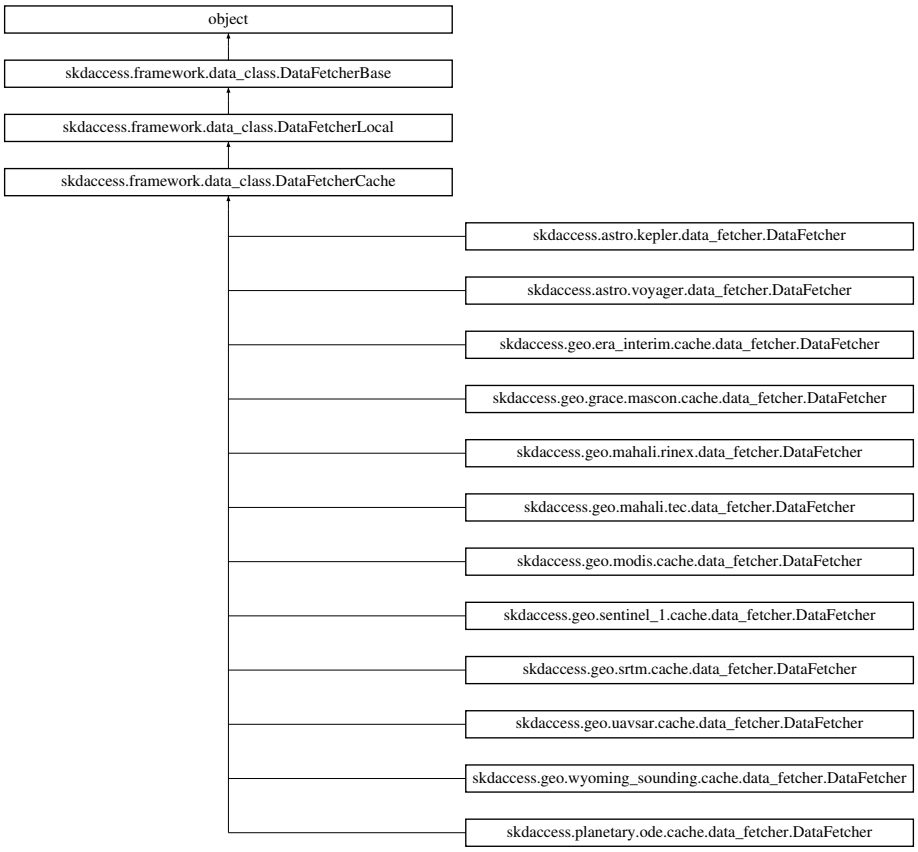
The documentation for this class was generated from the following file:

- framework/data_class.py

6.41 skdaccess.framework.data_class.DataFetcherCache Class Reference

Data fetcher base class for downloading data and caching results on hard disk.

Inheritance diagram for skdaccess.framework.data_class.DataFetcherCache:



Public Member Functions

- def checkIfDataExists (self, in_file_name)

  Checks if the file exists on the filesystem and the file is not empty.
- def cacheData (self, keyname, online_path_list, username=None, password=None, authentication_url=None, cookiejar=None, use_requests=False, use_progress_bar=True)

Download and store specified data to local disk.

- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getHDFStorage (self, keyname)

  Retrieve a Pandas HDF Store for a dataset.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def output (self)

  Output data wrapper.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- ap_paramList
- verbose

6.41.1 Detailed Description

Data fetcher base class for downloading data and caching results on hard disk.

6.41.2 Member Function Documentation

6.41.2.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.41.2.2 cacheData()

```
def skdaccess.framework.data_class.DataFetcherCache.cacheData (
              self,
              keyname,
              online_path_list,
              username = None,
              password = None,
              authentication_url = None,
              cookiejar = None,
              use_requests = False,
              use_progress_bar = True )
```

Download and store specified data to local disk.

Parameters

keyname	Name of dataset in configuration file
online_path_list	List of urls to data
username	Username for accessing online resources
password	Password for accessing online resources
authentication_url	The url used for authentication (unused when use_requests=True)
cookiejar	The cookiejar that stores credentials (unused when use_requests=True)
use_requests	Use the requests library instead of the standard library for accessing resources
use_progress_bar	Use a progress bar to show number of items downloaded

Returns

List of downloaded file locations

6.41.2.3 checkIfDataExists()

```
def skdaccess.framework.data_class.DataFetcherCache.checkIfDataExists (
              self,
              in_file_name )
```

Checks if the file exists on the filesystem and the file is not empty.

Parameters

in_file_name	Input filename to test

Returns

True if data exists and False otherwise

6.41.2.4 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.41.2.5 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.41.2.6 getHDFStorage()

```
def skdaccess.framework.data_class.DataFetcherCache.getHDFStorage (
              self,
              keyname )
```

Retrieve a Pandas HDF Store for a dataset.

Parameters

keyname	Key name of HDF store

Returns

Pandas HDF Store

6.41.2.7 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.41.2.8 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherCache.multirun_enabled (
              self )
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.41.2.9 output()

```
def skdaccess.framework.data_class.DataFetcherBase.output (
              self )  [inherited]
```

Output data wrapper.

Returns

Datawrapper

6.41.2.10 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.41.2.11 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.41.2.12 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.41.2.13 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.41.2.14 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.41.3 Member Data Documentation

6.41.3.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.41.3.2 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

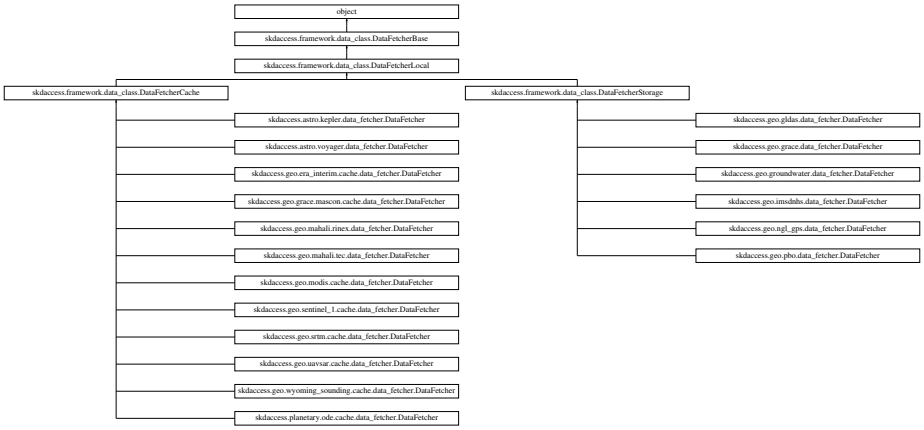
The documentation for this class was generated from the following file:

- framework/data_class.py

6.42 skdaccess.framework.data_class.DataFetcherLocal Class Reference

Data fetcher base class for use when storing data locally.

Inheritance diagram for skdaccess.framework.data_class.DataFetcherLocal:



Public Member Functions

- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def output (self)

  Output data wrapper.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- ap_paramList
- verbose

6.42.1 Detailed Description

Data fetcher base class for use when storing data locally.

6.42.2 Member Function Documentation

6.42.2.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.42.2.2 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.42.2.3 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.42.2.4 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.42.2.5 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherBase.multirun_enabled (
              self )  [inherited]
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.42.2.6 output()

```
def skdaccess.framework.data_class.DataFetcherBase.output (
              self )  [inherited]
```

Output data wrapper.

Returns

Datawrapper

6.42.2.7 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.42.2.8 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.42.2.9 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.42.2.10 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.42.2.11 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.42.3 Member Data Documentation

6.42.3.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.42.3.2 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

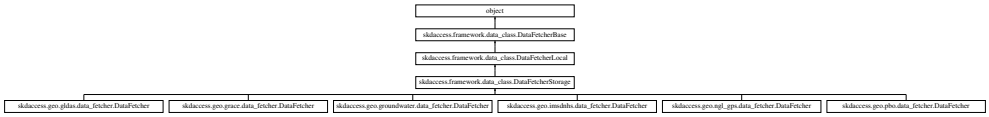
The documentation for this class was generated from the following file:

- framework/data_class.py

6.43 skdaccess.framework.data_class.DataFetcherStorage Class Reference

Data fetcher base class for use when entire data set is downloaded.

Inheritance diagram for skdaccess.framework.data_class.DataFetcherStorage:



Public Member Functions

- def downloadFullDataset (cls, out_file, use_file=None)

  Abstract function used to download full data set.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def getDataLocation (data_name)

  Get the location of data set.
- def setDataLocation (data_name, location, key='data_location')

  Set the location of a data set.
- def output (self)

  Output data wrapper.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- ap_paramList
- verbose

6.43.1 Detailed Description

Data fetcher base class for use when entire data set is downloaded.

6.43.2 Member Function Documentation

6.43.2.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.43.2.2 downloadFullDataset()

```
def skdaccess.framework.data_class.DataFetcherStorage.downloadFullDataset (
              cls,
              out_file,
              use_file = None )
```

Abstract function used to download full data set.

Parameters

out_file	output file name
use_file	Use previously downloaded data

Returns

Absolute path of parsed data

6.43.2.3 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.43.2.4 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
              data_name )  [inherited]
```

Get the location of data set.

Parameters

data_name	Name of data set

Returns

string of data location, None if not found

6.43.2.5 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.43.2.6 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStorage.multirun_enabled (
              self )
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.43.2.7 output()

```
def skdaccess.framework.data_class.DataFetcherBase.output (
              self )  [inherited]
```

Output data wrapper.

Returns

Datawrapper

6.43.2.8 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.43.2.9 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.43.2.10 setDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.setDataLocation (
              data_name,
              location,
              key = 'data_location' )  [inherited]
```

Set the location of a data set.

Parameters

data_name	Name of data set
location	Location of data set
key	Key of configuration option

6.43.2.11 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.43.2.12 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.43.3 Member Data Documentation

6.43.3.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.43.3.2 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

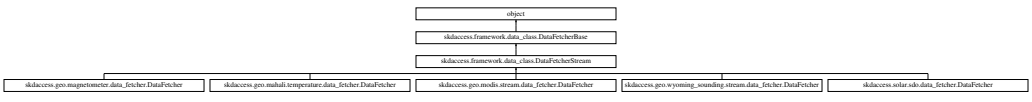
The documentation for this class was generated from the following file:

- framework/data_class.py

6.44 skdaccess.framework.data_class.DataFetcherStream Class Reference

Data fetcher base class for downloading data into memory.

Inheritance diagram for skdaccess.framework.data_class.DataFetcherStream:



Public Member Functions

- def retrieveOnlineData (self, data_specification)

  Method for downloading data into memory.
- def multirun_enabled (self)

  Returns whether or not this data fetcher is multirun enabled.
- def output (self)

  Output data wrapper.
- def perturb (self)

  Perturb parameters.
- def reset (self)

  Set all parameters to initial value.
- def __str__ (self)

  Generate string description.
- def getMetadata (self)

  Return metadata about Data Fetcher.
- def getConfig ()

  Retrieve skdaccess configuration.
- def writeConfig (conf)

  Write config to disk.
- def verbose_print (self, args, kwargs)

  Print statement if verbose flag is set.

Public Attributes

- ap_paramList
- verbose

6.44.1 Detailed Description

Data fetcher base class for downloading data into memory.

6.44.2 Member Function Documentation

6.44.2.1 __str__()

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
              self )  [inherited]
```

Generate string description.

6.44.2.2 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )  [inherited]
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.44.2.3 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
              self )  [inherited]
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.44.2.4 multirun_enabled()

```
def skdaccess.framework.data_class.DataFetcherStream.multirun_enabled (
              self )
```

Returns whether or not this data fetcher is multirun enabled.

Returns

Boolean indicating whether or not this data fetcher is multirun enabled

6.44.2.5 output()

```
def skdaccess.framework.data_class.DataFetcherBase.output (
              self )  [inherited]
```

Output data wrapper.

Returns

Datawrapper

6.44.2.6 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
              self )  [inherited]
```

Perturb parameters.

6.44.2.7 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
              self )  [inherited]
```

Set all parameters to initial value.

6.44.2.8 retrieveOnlineData()

```
def skdaccess.framework.data_class.DataFetcherStream.retrieveOnlineData (
              self,
              data_specification )
```

Method for downloading data into memory.

Parameters

data_specification	Url list of data to be retrieved

Returns

Retrieved data

6.44.2.9 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
              self,
              args,
              kwargs )  [inherited]
```

Print statement if verbose flag is set.

Parameters

**args*	Arguments to pass to print
***kwargs*	Keyword arguments to pass to print

6.44.2.10 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
              conf )  [inherited]
```

Write config to disk.

Parameters

conf	configparser.ConfigParser object

6.44.3 Member Data Documentation

6.44.3.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList  [inherited]
```

6.44.3.2 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose  [inherited]
```

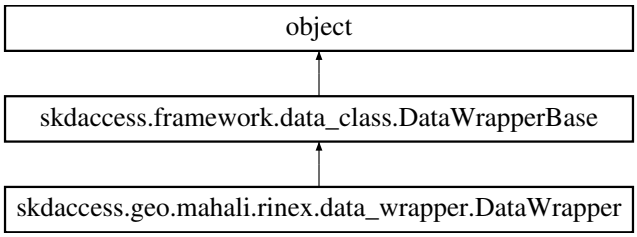
The documentation for this class was generated from the following file:

- framework/data_class.py

6.45 skdaccess.geo.mahali.rinex.data_wrapper.DataWrapper Class Reference

Data wrapper for Mahali data.

Inheritance diagram for skdaccess.geo.mahali.rinex.data_wrapper.DataWrapper:



Public Member Functions

- def getIterator (self)

  Get iterator to Mahali data.
- def update (self, obj)

  Updated wrapped data.
- def updateMetadata (self, new_metadata)

  Update metadata.
- def get (self)

  Retrieve stored data.
- def getResults (self)

  Retrieve accumulated results, if any.
- def addResult (self, rkey, rres)

  Add a result to the data wrapper.
- def reset (self)

  Reset data back to original state.
- def info (self, key=None)

  Get information about data wrapper.
- def __len__ (self)

  Get length of wrapped data.
- def getRunID (self)

  Get the Run ID.

Public Attributes

- data
- results
- constants
- run_id
- meta_data

6.45.1 Detailed Description

Data wrapper for Mahali data.

6.45.2 Member Function Documentation

6.45.2.1 __len__()

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
              self )  [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.45.2.2 addResult()

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
              self,
              rkey,
              rres )  [inherited]
```

Add a result to the data wrapper.

Parameters

rkey	Result key
rres	Result

6.45.2.3 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
              self )  [inherited]
```

Retrieve stored data.

Returns

Stored data

6.45.2.4 getIterator()

```
def skdaccess.geo.mahali.rinex.data_wrapper.DataWrapper.getIterator (
              self )
```

Get iterator to Mahali data.

Returns

Iterator yielding (site,date,nav,obs)

6.45.2.5 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
              self )  [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.45.2.6 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
              self )  [inherited]
```

Get the Run ID.

Returns

run_id

6.45.2.7 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
              self,
              key = None )  [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.45.2.8 reset()

```
def skdaccess.framework.data_class.DataWrapperBase.reset (
              self )  [inherited]
```

Reset data back to original state.

6.45.2.9 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
              self,
              obj )  [inherited]
```

Updated wrapped data.

Parameters

obj	New data for wrapper

6.45.2.10 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
              self,
              new_metadata )  [inherited]
```

Update metadata.

Parameters

new_metadata	New metadata

6.45.3 Member Data Documentation

6.45.3.1 constants

`skdaccess.framework.data_class.DataWrapperBase.constants` [inherited]

6.45.3.2 data

`skdaccess.framework.data_class.DataWrapperBase.data` [inherited]

6.45.3.3 meta_data

`skdaccess.framework.data_class.DataWrapperBase.meta_data` [inherited]

6.45.3.4 results

`skdaccess.framework.data_class.DataWrapperBase.results` [inherited]

6.45.3.5 run_id

`skdaccess.framework.data_class.DataWrapperBase.run_id` [inherited]

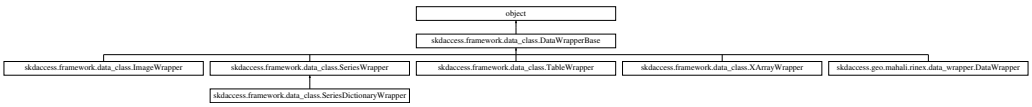
The documentation for this class was generated from the following file:

- geo/mahali/rinex/data_wrapper.py

6.46 skdaccess.framework.data_class.DataWrapperBase Class Reference

Base class for wrapping data for use in DiscoveryPipeline.

Inheritance diagram for skdaccess.framework.data_class.DataWrapperBase:



Public Member Functions

- def __init__ (self, obj_wrap, run_id=-1, meta_data=None)

  Construct wrapper from input data.
- def update (self, obj)

  Updated wrapped data.
- def updateMetadata (self, new_metadata)

  Update metadata.
- def get (self)

  Retrieve stored data.
- def getResults (self)

  Retrieve accumulated results, if any.
- def addResult (self, rkey, rres)

  Add a result to the data wrapper.
- def reset (self)

  Reset data back to original state.
- def info (self, key=None)

  Get information about data wrapper.
- def getIterator (self)

  Get an iterator to the data.
- def __len__ (self)

  Get length of wrapped data.
- def getRunID (self)

  Get the Run ID.

Public Attributes

- data
- results
- constants
- run_id
- meta_data

6.46.1 Detailed Description

Base class for wrapping data for use in DiscoveryPipeline.

6.46.2 Constructor & Destructor Documentation

6.46.2.1 __init__()

```
def skdaccess.framework.data_class.DataWrapperBase.__init__ (
              self,
              obj_wrap,
              run_id = -1,
              meta_data = None )
```

Construct wrapper from input data.

Parameters

obj_wrap	Data to be wrapped
run_id	ID of the run
meta_data	Metadata to store with data

6.46.3 Member Function Documentation

6.46.3.1 __len__()

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
              self )
```

Get length of wrapped data.

Returns

length of wrapped data

6.46.3.2 addResult()

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
              self,
              rkey,
              rres )
```

Add a result to the data wrapper.

Parameters

rkey	Result key
rres	Result

6.46.3.3 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
              self )
```

Retrieve stored data.

Returns

Stored data

6.46.3.4 getIterator()

```
def skdaccess.framework.data_class.DataWrapperBase.getIterator (
              self )
```

Get an iterator to the data.

Returns

iterator to data

6.46.3.5 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
              self )
```

Retrieve accumulated results, if any.

Returns

store results

6.46.3.6 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
              self )
```

Get the Run ID.

Returns

run_id

6.46.3.7 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
              self,
              key = None )
```

Get information about data wrapper.

Returns

The stored metadata

6.46.3.8 reset()

```
def skdaccess.framework.data_class.DataWrapperBase.reset (
              self )
```

Reset data back to original state.

6.46.3.9 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
              self,
              obj )
```

Updated wrapped data.

Parameters

obj	New data for wrapper

6.46.3.10 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
              self,
              new_metadata )
```

Update metadata.

Parameters

new_metadata	New metadata

6.46.4 Member Data Documentation

6.46.4.1 constants

```
skdaccess.framework.data_class.DataWrapperBase.constants
```

6.46.4.2 data

```
skdaccess.framework.data_class.DataWrapperBase.data
```

6.46.4.3 meta_data

```
skdaccess.framework.data_class.DataWrapperBase.meta_data
```

6.46.4.4 results

```
skdaccess.framework.data_class.DataWrapperBase.results
```

6.46.4.5 run_id

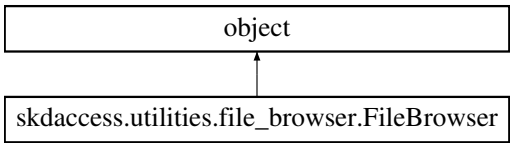
```
skdaccess.framework.data_class.DataWrapperBase.run_id
```

The documentation for this class was generated from the following file:

- framework/data_class.py

6.47 skdaccess.utilities.file_browser.FileBrowser Class Reference

Inheritance diagram for skdaccess.utilities.file_browser.FileBrowser:



Public Member Functions

- def __init__ (self)
- def widget (self)

Public Attributes

- path
- files
- dirs

6.47.1 Constructor & Destructor Documentation

6.47.1.1 __init__()

```
def skdaccess.utilities.file_browser.FileBrowser.__init__ (
              self )
```

6.47.2 Member Function Documentation

6.47.2.1 widget()

```
def skdaccess.utilities.file_browser.FileBrowser.widget (
              self )
```

6.47.3 Member Data Documentation

6.47.3.1 dirs

```
skdaccess.utilities.file_browser.FileBrowser.dirs
```

6.47.3.2 files

```
skdaccess.utilities.file_browser.FileBrowser.files
```

6.47.3.3 path

```
skdaccess.utilities.file_browser.FileBrowser.path
```

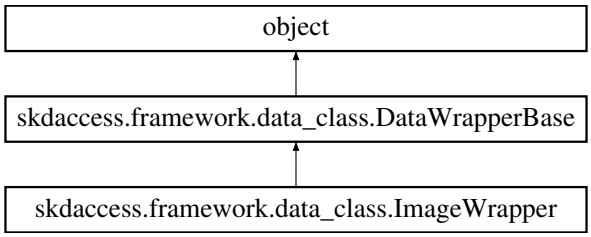
The documentation for this class was generated from the following file:

- utilities/file_browser.py

6.48 skdaccess.framework.data_class.ImageWrapper Class Reference

Wrapper for image data.

Inheritance diagram for skdaccess.framework.data_class.ImageWrapper:



Public Member Functions

- def getIterator (self)

  Get an iterator to the data.
- def updateData (self, label, new_data)

  Change image.
- def deleteData (self, label)

  Delete image.
- def update (self, obj)

  Updated wrapped data.
- def updateMetadata (self, new_metadata)

  Update metadata.
- def get (self)

  Retrieve stored data.
- def getResults (self)

  Retrieve accumulated results, if any.
- def addResult (self, rkey, rres)

  Add a result to the data wrapper.
- def reset (self)

  Reset data back to original state.
- def info (self, key=None)

  Get information about data wrapper.
- def __len__ (self)

  Get length of wrapped data.
- def getRunID (self)

  Get the Run ID.

Public Attributes

- data
- results
- constants
- run_id
- meta_data

6.48.1 Detailed Description

Wrapper for image data.

6.48.2 Member Function Documentation

6.48.2.1 __len__()

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
              self )  [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.48.2.2 addResult()

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
              self,
              rkey,
              rres )  [inherited]
```

Add a result to the data wrapper.

Parameters

rkey	Result key
rres	Result

6.48.2.3 deleteData()

```
def skdaccess.framework.data_class.ImageWrapper.deleteData (
              self,
              label )
```

Delete image.

Parameters

label	Delete image with label

6.48.2.4 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
```

```
              self )  [inherited]
```

Retrieve stored data.

Returns

Stored data

6.48.2.5 getIterator()

```
def skdaccess.framework.data_class.ImageWrapper.getIterator (
              self )
```

Get an iterator to the data.

Returns

Iterator yielding (label, image_data)

6.48.2.6 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
              self )  [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.48.2.7 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
              self )  [inherited]
```

Get the Run ID.

Returns

run_id

6.48.2.8 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
              self,
              key = None )  [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.48.2.9 reset()

```
def skdaccess.framework.data_class.DataWrapperBase.reset (
              self )  [inherited]
```

Reset data back to original state.

6.48.2.10 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
              self,
              obj )  [inherited]
```

Updated wrapped data.

Parameters

obj	New data for wrapper

6.48.2.11 updateData()

```
def skdaccess.framework.data_class.ImageWrapper.updateData (
              self,
              label,
              new_data )
```

Change image.

Parameters

label	Label of data to be changed
new_data	New data to replace old data

6.48.2.12 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
            self,
            new_metadata )  [inherited]
```

Update metadata.

Parameters

new_metadata	New metadata

6.48.3 Member Data Documentation

6.48.3.1 constants

```
skdaccess.framework.data_class.DataWrapperBase.constants  [inherited]
```

6.48.3.2 data

```
skdaccess.framework.data_class.DataWrapperBase.data  [inherited]
```

6.48.3.3 meta_data

```
skdaccess.framework.data_class.DataWrapperBase.meta_data  [inherited]
```

6.48.3.4 results

```
skdaccess.framework.data_class.DataWrapperBase.results  [inherited]
```

6.48.3.5 run_id

```
skdaccess.framework.data_class.DataWrapperBase.run_id  [inherited]
```

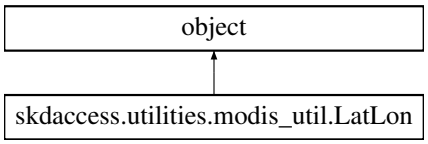
The documentation for this class was generated from the following file:

- framework/data_class.py

6.49 skdaccess.utilities.modis_util.LatLon Class Reference

Calculates Lat/Lon position from y,x pixel coordinate.

Inheritance diagram for skdaccess.utilities.modis_util.LatLon:



Public Member Functions

- def __init__ (self, metadata, x_offset=0, y_offset=0)

  Initialize getLatLon object.
- def __call__ (self, y, x)

  Convert pixel coordinates to lat/lon.

Public Attributes

- x_offset
- y_offset
- lat_data
- lon_data
- alat
- alon

6.49.1 Detailed Description

Calculates Lat/Lon position from y,x pixel coordinate.

6.49.2 Constructor & Destructor Documentation

6.49.2.1 __init__()

```
def skdaccess.utilities.modis_util.LatLon.__init__ (
              self,
              metadata,
              x_offset = 0,
              y_offset = 0 )
```

Initialize getLatLon object.

Parameters

metadata	Image metadata
x_offset	Pixel offset (used when gridding data)
y_offset	Pixel offset (used when gridding data)

6.49.3 Member Function Documentation

6.49.3.1 __call__()

```
def skdaccess.utilities.modis_util.LatLon.__call__ (
              self,
              y,
              x )
```

Convert pixel coordinates to lat/lon.

Parameters

y	y coordinate
x	x coordinate

Returns

(lat, lon)

6.49.4 Member Data Documentation

6.49.4.1 alat

```
skdaccess.utilities.modis_util.LatLon.alat
```

6.49.4.2 alon

```
skdaccess.utilities.modis_util.LatLon.alon
```

6.49.4.3 lat_data

```
skdaccess.utilities.modis_util.LatLon.lat_data
```

6.49.4.4 lon_data

```
skdaccess.utilities.modis_util.LatLon.lon_data
```

6.49.4.5 x_offset

```
skdaccess.utilities.modis_util.LatLon.x_offset
```

6.49.4.6 y_offset

```
skdaccess.utilities.modis_util.LatLon.y_offset
```

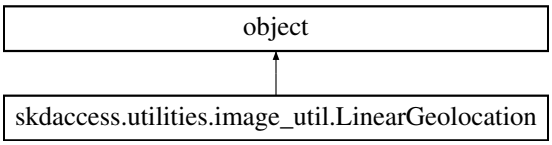
The documentation for this class was generated from the following file:

- utilities/modis_util.py

6.50 skdaccess.utilities.image_util.LinearGeolocation Class Reference

This class provides functions to convert between pixel and geodetic coordinates.

Inheritance diagram for skdaccess.utilities.image_util.LinearGeolocation:



Public Member Functions

- def __init__ (self, data, extents, x_offset=0, y_offset=0, flip_y=False)

  Initialize Linear Geolocation object.
- def getLatLon (self, y, x)

  Retrive the latitude and longitude from pixel coordinates.
- def getYX (self, lat, lon)

  Retrive the pixel coordinates from the latitude and longitude.
- def getExtents (self)

  Retrieve the extents of the data.

Public Attributes

- flip_y
- lon_extents
- lat_extents
- lat_pixel_size
- lon_pixel_size
- start_lat
- start_lon
- x_offset
- y_offset
- len_x
- len_y

6.50.1 Detailed Description

This class provides functions to convert between pixel and geodetic coordinates.

Assumes a linear relationship between pixel and geodetic coordinates

6.50.2 Constructor & Destructor Documentation

6.50.2.1 __init__()

```
def skdaccess.utilities.image_util.LinearGeolocation.__init__ (
              self,
              data,
              extents,
              x_offset = 0,
              y_offset = 0,
              flip_y = False )
```

Initialize Linear Geolocation object.

Parameters

data	Numpy 2d data
extents	Latitude and longitude extents
x_offset	Pixel offset in x
y_offset	Pixel offset in y
flip_y	The y axis has been flipped so that increasing y values are decreasing in latitude

6.50.3 Member Function Documentation

6.50.3.1 getExtents()

```
def skdaccess.utilities.image_util.LinearGeolocation.getExtents (
              self )
```

Retrieve the extents of the data.

Returns

(minimum_longitude, maximum_longitude, minimum_latitude, maximum_latitude)

6.50.3.2 getLatLon()

```
def skdaccess.utilities.image_util.LinearGeolocation.getLatLon (
              self,
              y,
              x )
```

Retrive the latitude and longitude from pixel coordinates.

Parameters

y	The y pixel
x	The x pixel

Returns

(latitude, longitude) of the pixel coordinate

6.50.3.3 getYX()

```
def skdaccess.utilities.image_util.LinearGeolocation.getYX (
              self,
              lat,
              lon )
```

Retrive the pixel coordinates from the latitude and longitude.

Parameters

lat	The Latitude
lon	The Longitude

Returns

(y, x) pixel coordinates of the input latitude and longitude

6.50.4 Member Data Documentation

6.50.4.1 flip_y

```
skdaccess.utilities.image_util.LinearGeolocation.flip_y
```

6.50.4.2 lat_extents

```
skdaccess.utilities.image_util.LinearGeolocation.lat_extents
```

6.50.4.3 lat_pixel_size

```
skdaccess.utilities.image_util.LinearGeolocation.lat_pixel_size
```

6.50.4.4 len_x

```
skdaccess.utilities.image_util.LinearGeolocation.len_x
```

6.50.4.5 len_y

```
skdaccess.utilities.image_util.LinearGeolocation.len_y
```

6.50.4.6 lon_extents

```
skdaccess.utilities.image_util.LinearGeolocation.lon_extents
```

6.50.4.7 lon_pixel_size

```
skdaccess.utilities.image_util.LinearGeolocation.lon_pixel_size
```

6.50.4.8 start_lat

```
skdaccess.utilities.image_util.LinearGeolocation.start_lat
```

6.50.4.9 start_lon

```
skdaccess.utilities.image_util.LinearGeolocation.start_lon
```

6.50.4.10 x_offset

```
skdaccess.utilities.image_util.LinearGeolocation.x_offset
```

6.50.4.11 y_offset

```
skdaccess.utilities.image_util.LinearGeolocation.y_offset
```

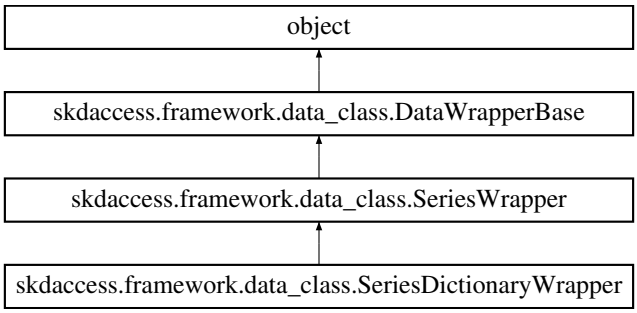
The documentation for this class was generated from the following file:

- utilities/image_util.py

6.51 skdaccess.framework.data_class.SeriesDictionaryWrapper Class Reference

Data wrapper for series data using a dictionary of data frames.

Inheritance diagram for skdaccess.framework.data_class.SeriesDictionaryWrapper:



Public Member Functions

- def getIterator (self)

  Get an iterator to the data.
- def getIndices (self)

  Get the indices of the data.
- def getLength (self)

  Get total number of series that the iterate will loop over.
- def update (self, obj)

  Updated wrapped data.
- def updateMetadata (self, new_metadata)

  Update metadata.
- def get (self)

  Retrieve stored data.

- def getResults (self)

  Retrieve accumulated results, if any.
- def addResult (self, rkey, rres)

  Add a result to the data wrapper.
- def reset (self)

  Reset data back to original state.
- def info (self, key=None)

  Get information about data wrapper.
- def __len__ (self)

  Get length of wrapped data.
- def getRunID (self)

  Get the Run ID.

Public Attributes

- data_names
- error_names
- data
- results
- constants
- run_id
- meta_data

6.51.1 Detailed Description

Data wrapper for series data using a dictionary of data frames.

6.51.2 Member Function Documentation

6.51.2.1 __len__()

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
              self )  [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.51.2.2 addResult()

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
              self,
              rkey,
              rres )  [inherited]
```

Add a result to the data wrapper.

Parameters

rkey	Result key
rres	Result

6.51.2.3 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
              self )  [inherited]
```

Retrieve stored data.

Returns

Stored data

6.51.2.4 getIndices()

```
def skdaccess.framework.data_class.SeriesDictionaryWrapper.getIndices (
              self )
```

Get the indices of the data.

Returns

index of data

6.51.2.5 getIterator()

```
def skdaccess.framework.data_class.SeriesDictionaryWrapper.getIterator (
              self )
```

Get an iterator to the data.

Returns

Iterator (label, data, errors) that will cycle over data and error names

6.51.2.6 getLength()

```
def skdaccess.framework.data_class.SeriesDictionaryWrapper.getLength (
              self )
```

Get total number of series that the iterate will loop over.

Returns

Number of series iterator will traverse over

6.51.2.7 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
              self )  [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.51.2.8 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
              self )  [inherited]
```

Get the Run ID.

Returns

run_id

6.51.2.9 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
              self,
              key = None )  [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.51.2.10 reset()

```
def skdaccess.framework.data_class.DataWrapperBase.reset (
              self )  [inherited]
```

Reset data back to original state.

6.51.2.11 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
              self,
              obj )  [inherited]
```

Updated wrapped data.

Parameters

obj	New data for wrapper

6.51.2.12 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
              self,
              new_metadata )  [inherited]
```

Update metadata.

Parameters

new_metadata	New metadata

6.51.3 Member Data Documentation

6.51.3.1 constants

```
skdaccess.framework.data_class.DataWrapperBase.constants  [inherited]
```

6.51.3.2 data

```
skdaccess.framework.data_class.DataWrapperBase.data  [inherited]
```

6.51.3.3 data_names

```
skdaccess.framework.data_class.SeriesWrapper.data_names  [inherited]
```

6.51.3.4 error_names

```
skdaccess.framework.data_class.SeriesWrapper.error_names  [inherited]
```

6.51.3.5 meta_data

```
skdaccess.framework.data_class.DataWrapperBase.meta_data  [inherited]
```

6.51.3.6 results

```
skdaccess.framework.data_class.DataWrapperBase.results  [inherited]
```

6.51.3.7 run_id

```
skdaccess.framework.data_class.DataWrapperBase.run_id  [inherited]
```

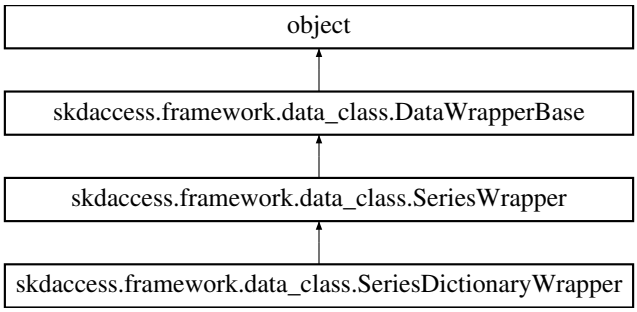
The documentation for this class was generated from the following file:

- framework/data_class.py

6.52 skdaccess.framework.data_class.SeriesWrapper Class Reference

Data wrapper for series data using a data panel.

Inheritance diagram for skdaccess.framework.data_class.SeriesWrapper:



Public Member Functions

- def __init__ (self, obj_wrap, data_names, error_names=None, meta_data=None, run_id=-1)

  Initialize Series Wrapper.
- def getIterator (self)

  Get an iterator to the data.
- def getIndices (self)

  Get the indicies of the data.
- def getLength (self)

  Get total number of series that the iterate will loop over.
- def update (self, obj)

  Updated wrapped data.
- def updateMetadata (self, new_metadata)

  Update metadata.
- def get (self)

  Retrieve stored data.
- def getResults (self)

  Retrieve accumulated results, if any.
- def addResult (self, rkey, rres)

  Add a result to the data wrapper.
- def reset (self)

  Reset data back to original state.
- def info (self, key=None)

  Get information about data wrapper.
- def __len__ (self)

  Get length of wrapped data.
- def getRunID (self)

  Get the Run ID.

Public Attributes

- data_names
- error_names
- data
- results
- constants
- run_id
- meta_data

6.52.1 Detailed Description

Data wrapper for series data using a data panel.

6.52.2 Constructor & Destructor Documentation

6.52.2.1 __init__()

```
def skdaccess.framework.data_class.SeriesWrapper.__init__ (
              self,
              obj_wrap,
              data_names,
              error_names = None,
              meta_data = None,
              run_id = -1 )
```

Initialize Series Wrapper.

Parameters

obj_wrap	Pandas data panel to wrap
data_names	List of data column names
error_names	List of error column names
meta_data	Metadata
run_id	ID of run

6.52.3 Member Function Documentation

6.52.3.1 __len__()

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
              self )  [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.52.3.2 addResult()

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
              self,
              rkey,
              rres )  [inherited]
```

Add a result to the data wrapper.

Parameters

rkey	Result key
rres	Result

6.52.3.3 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
              self )  [inherited]
```

Retrieve stored data.

Returns

Stored data

6.52.3.4 getIndices()

```
def skdaccess.framework.data_class.SeriesWrapper.getIndices (
              self )
```

Get the indicies of the data.

Returns

index of data

6.52.3.5 getIterator()

```
def skdaccess.framework.data_class.SeriesWrapper.getIterator (
              self )
```

Get an iterator to the data.

Returns

Iterator (label, data, errors) that will cycle over data and error names

6.52.3.6 getLength()

```
def skdaccess.framework.data_class.SeriesWrapper.getLength (
              self )
```

Get total number of series that the iterate will loop over.

Returns

Number of series iterator will traverse over

6.52.3.7 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
              self )  [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.52.3.8 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
              self )  [inherited]
```

Get the Run ID.

Returns

run_id

6.52.3.9 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
              self,
              key = None )  [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.52.3.10 reset()

```
def skdaccess.framework.data_class.DataWrapperBase.reset (
              self )  [inherited]
```

Reset data back to original state.

6.52.3.11 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
              self,
              obj )  [inherited]
```

Updated wrapped data.

Parameters

obj	New data for wrapper

6.52.3.12 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
              self,
              new_metadata )  [inherited]
```

Update metadata.

Parameters

new_metadata	New metadata

6.52.4 Member Data Documentation

6.52.4.1 constants

```
skdaccess.framework.data_class.DataWrapperBase.constants  [inherited]
```

6.52.4.2 data

```
skdaccess.framework.data_class.DataWrapperBase.data  [inherited]
```

6.52.4.3 data_names

```
skdaccess.framework.data_class.SeriesWrapper.data_names
```

6.52.4.4 error_names

```
skdaccess.framework.data_class.SeriesWrapper.error_names
```

6.52.4.5 meta_data

```
skdaccess.framework.data_class.DataWrapperBase.meta_data  [inherited]
```

6.52.4.6 results

```
skdaccess.framework.data_class.DataWrapperBase.results  [inherited]
```

6.52.4.7 run_id

```
skdaccess.framework.data_class.DataWrapperBase.run_id  [inherited]
```

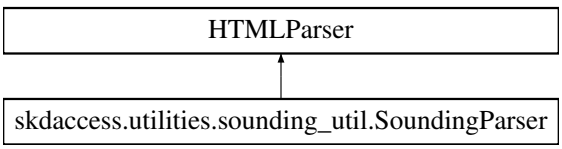
The documentation for this class was generated from the following file:

- framework/data_class.py

6.53 skdaccess.utilities.sounding_util.SoundingParser Class Reference

This class parses Wyoming Sounding data.

Inheritance diagram for skdaccess.utilities.sounding_util.SoundingParser:



Public Member Functions

- def __init__ (self)

  Initialize SoundingParser.
- def handle_starttag (self, tag, attrs)

  Function called everytime a start tag is encountered.
- def handle_endtag (self, tag)

  Function called everytime an end tag is encountered.
- def handle_data (self, data)

  Function to parse data between <pre> tags.

Public Attributes

- data_dict
- metadata_dict
- label
- in_pre_tag
- in_header
- read_data
- tmp

6.53.1 Detailed Description

This class parses Wyoming Sounding data.

6.53.2 Constructor & Destructor Documentation

6.53.2.1 __init__()

```
def skdaccess.utilities.sounding_util.SoundingParser.__init__ (
              self )
```

Initialize SoundingParser.

6.53.3 Member Function Documentation

6.53.3.1 handle_data()

```
def skdaccess.utilities.sounding_util.SoundingParser.handle_data (
              self,
              data )
```

Function to parse data between <pre> tags.

Parameters

data	Input data

6.53.3.2 handle_endtag()

```
def skdaccess.utilities.sounding_util.SoundingParser.handle_endtag (
              self,
              tag )
```

Function called everytime an end tag is encountered.

Parameters

tag	Ending tag

6.53.3.3 handle_starttag()

```
def skdaccess.utilities.sounding_util.SoundingParser.handle_starttag (
              self,
              tag,
              attrs )
```

Function called everytime a start tag is encountered.

Parameters

tag	Starting tag
attrs	Tag attributes

6.53.4 Member Data Documentation

6.53.4.1 data_dict

```
skdaccess.utilities.sounding_util.SoundingParser.data_dict
```

6.53.4.2 in_header

```
skdaccess.utilities.sounding_util.SoundingParser.in_header
```

6.53.4.3 in_pre_tag

```
skdaccess.utilities.sounding_util.SoundingParser.in_pre_tag
```

6.53.4.4 label

```
skdaccess.utilities.sounding_util.SoundingParser.label
```

6.53.4.5 metadata_dict

```
skdaccess.utilities.sounding_util.SoundingParser.metadata_dict
```

6.53.4.6 read_data

```
skdaccess.utilities.sounding_util.SoundingParser.read_data
```

6.53.4.7 tmp

```
skdaccess.utilities.sounding_util.SoundingParser.tmp
```

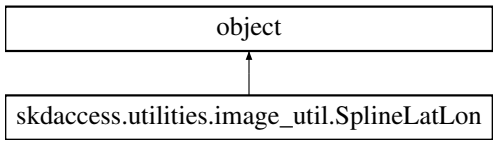
The documentation for this class was generated from the following file:

- utilities/sounding_util.py

6.54 skdaccess.utilities.image_util.SplineLatLon Class Reference

Holds a 2d spline for interpolating lat/lon grid.

Inheritance diagram for skdaccess.utilities.image_util.SplineLatLon:



Public Member Functions

- def __init__ (self, lat_func=None, lon_func=None, lat_grid=None, lon_grid=None, x_points=None, y_points=None, lat_extents=None, lon_extents=None, y_num_pixels=None, x_num_pixels=None, x_offset=0, y_offset=0, interp_type='grid')

  Initialize SplineLatLon with premade lat/lon functions or information about the latitude and longitude.
- def __call__ (self, y, x)

  Convert pixel coordinates to lat/lon.

Public Attributes

- lat_func
- lon_func
- x_offset
- y_offset

6.54.1 Detailed Description

Holds a 2d spline for interpolating lat/lon grid.

6.54.2 Constructor & Destructor Documentation

6.54.2.1 __init__()

```
def skdaccess.utilities.image_util.SplineLatLon.__init__ (
              self,
              lat_func = None,
              lon_func = None,
              lat_grid = None,
              lon_grid = None,
              x_points = None,
              y_points = None,
              lat_extents = None,
              lon_extents = None,
              y_num_pixels = None,
              x_num_pixels = None,
              x_offset = 0,
              y_offset = 0,
              interp_type = 'grid' )
```

Initialize SplineLatLon with premade lat/lon functions or information about the latitude and longitude.

Parameters

lat_func	Latitude spline function
lon_func	Longitude spline function
lat_grid	Latitude grid
lon_grid	Longitude grid
x_points	1d array of x coordinates
y_points	1d array of y coordinates
lon_extents	Extent of data in longitude
lat_extents	Extent of data in latitude
y_num_pixels	Number of y coordinates
x_num_pixels	Number of x coordinates
x_offset	Offset in the x coordinate
y_offset	Offset in the y coordinate
interp_type	Interpolate type. Currently only 'grid' type is supported

6.54.3 Member Function Documentation

6.54.3.1 __call__()

```
def skdaccess.utilities.image_util.SplineLatLon.__call__ (
              self,
              y,
              x )
```

Convert pixel coordinates to lat/lon.

Parameters

y	y coordinate
x	x coordinate

Returns

(lat, lon)

6.54.4 Member Data Documentation

6.54.4.1 lat_func

```
skdaccess.utilities.image_util.SplineLatLon.lat_func
```

6.54.4.2 lon_func

```
skdaccess.utilities.image_util.SplineLatLon.lon_func
```

6.54.4.3 x_offset

```
skdaccess.utilities.image_util.SplineLatLon.x_offset
```

6.54.4.4 y_offset

```
skdaccess.utilities.image_util.SplineLatLon.y_offset
```

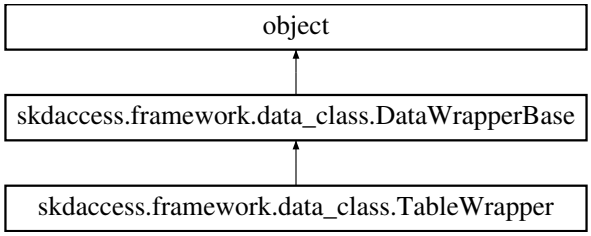
The documentation for this class was generated from the following file:

- utilities/image_util.py

6.55 skdaccess.framework.data_class.TableWrapper Class Reference

Data wrapper for table data using an ordered dictionary.

Inheritance diagram for skdaccess.framework.data_class.TableWrapper:



Public Member Functions

- def __init__ (self, obj_wrap, run_id=-1, meta_data=None, default_columns=None, default_error_columns=None)

  Construct object from input data.
- def getIterator (self)

  Iterator access to data.
- def getLength (self)

  Get number of data frames.
- def updateData (self, label, index, column_names, new_data)

  Update wrapped data.
- def addColumn (self, label, column_names, new_data)

  Add new column to data.
- def getDefaultColumns (self)

  Get the default columns of data.
- def getDefaultErrorColumns (self)

  Get the default error columns of data.
- def removeFrames (self, label_list)

  Remove Data Frames from wrapper.
- def updateFrames (self, label_list, frame_list)

  Update data frames.
- def update (self, obj)

  Updated wrapped data.
- def updateMetadata (self, new_metadata)

  Update metadata.
- def get (self)

  Retrieve stored data.
- def getResults (self)

  Retrieve accumulated results, if any.
- def addResult (self, rkey, rres)

  Add a result to the data wrapper.
- def reset (self)

  Reset data back to original state.

- def info (self, key=None)

  Get information about data wrapper.
- def __len__ (self)

  Get length of wrapped data.
- def getRunID (self)

  Get the Run ID.

Public Attributes

- default_columns
- default_error_columns
- data
- results
- constants
- run_id
- meta_data

6.55.1 Detailed Description

Data wrapper for table data using an ordered dictionary.

6.55.2 Constructor & Destructor Documentation

6.55.2.1 __init__()

```
def skdaccess.framework.data_class.TableWrapper.__init__ (
              self,
              obj_wrap,
              run_id = -1,
              meta_data = None,
              default_columns = None,
              default_error_columns = None )
```

Construct object from input data.

Parameters

obj_wrap	Data to be wrapped
run_id	ID of the run
meta_data	Metadata to store with data
default_columns	Default columns for pipeline items
default_error_columns	Default error columns for pipeline items

6.55.3 Member Function Documentation

6.55.3.1 __len__()

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
              self )  [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.55.3.2 addColumn()

```
def skdaccess.framework.data_class.TableWrapper.addColumn (
              self,
              label,
              column_names,
              new_data )
```

Add new column to data.

Parameters

label	Data label
column_names	Names of columns to update
new_data	New data to add

6.55.3.3 addResult()

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
              self,
              rkey,
              rres )  [inherited]
```

Add a result to the data wrapper.

Parameters

rkey	Result key
rres	Result

6.55.3.4 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
              self )  [inherited]
```

Retrieve stored data.

Returns

Stored data

6.55.3.5 getDefaultColumns()

```
def skdaccess.framework.data_class.TableWrapper.getDefaultColumns (
              self )
```

Get the default columns of data.

Returns

List of default columns

6.55.3.6 getDefaultErrorColumns()

```
def skdaccess.framework.data_class.TableWrapper.getDefaultErrorColumns (
              self )
```

Get the default error columns of data.

Returns

List of default error columns

6.55.3.7 getIterator()

```
def skdaccess.framework.data_class.TableWrapper.getIterator (
              self )
```

Iterator access to data.

Returns

iterator to (label, data frame) from Dictionary

6.55.3.8 getLength()

```
def skdaccess.framework.data_class.TableWrapper.getLength (
              self )
```

Get number of data frames.

Returns

Number of data frames

6.55.3.9 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
              self )  [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.55.3.10 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
              self )  [inherited]
```

Get the Run ID.

Returns

run_id

6.55.3.11 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
              self,
              key = None )  [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.55.3.12 removeFrames()

```
def skdaccess.framework.data_class.TableWrapper.removeFrames (
              self,
              label_list )
```

Remove Data Frames from wrapper.

Parameters

label_list	List of labels to remove

6.55.3.13 reset()

```
def skdaccess.framework.data_class.DataWrapperBase.reset (
              self )  [inherited]
```

Reset data back to original state.

6.55.3.14 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
              self,
              obj )  [inherited]
```

Updated wrapped data.

Parameters

obj	New data for wrapper

6.55.3.15 updateData()

```
def skdaccess.framework.data_class.TableWrapper.updateData (
              self,
              label,
              index,
              column_names,
              new_data )
```

Update wrapped data.

Parameters

label	Data label
index	Index of data to update
column_names	Names of columns to update
new_data	Data to replace the old data

6.55.3.16 updateFrames()

```
def skdaccess.framework.data_class.TableWrapper.updateFrames (
              self,
              label_list,
              frame_list )
```

Update data frames.

Parameters

label_list	List of labels to update
frame_list	List of updated frames

6.55.3.17 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
```

```
self,
new_metadata ) [inherited]
```

Update metadata.

Parameters

new_metadata	New metadata

6.55.4 Member Data Documentation

6.55.4.1 constants

```
skdaccess.framework.data_class.DataWrapperBase.constants [inherited]
```

6.55.4.2 data

```
skdaccess.framework.data_class.DataWrapperBase.data [inherited]
```

6.55.4.3 default_columns

```
skdaccess.framework.data_class.TableWrapper.default_columns
```

6.55.4.4 default_error_columns

```
skdaccess.framework.data_class.TableWrapper.default_error_columns
```

6.55.4.5 meta_data

```
skdaccess.framework.data_class.DataWrapperBase.meta_data [inherited]
```

6.55.4.6 results

```
skdaccess.framework.data_class.DataWrapperBase.results  [inherited]
```

6.55.4.7 run_id

```
skdaccess.framework.data_class.DataWrapperBase.run_id  [inherited]
```

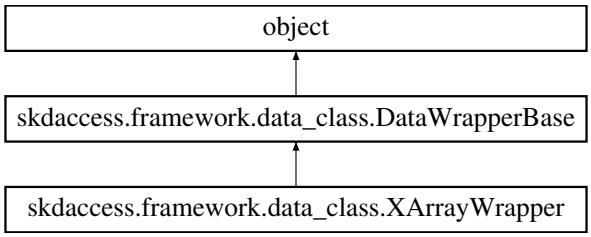
The documentation for this class was generated from the following file:

- framework/data_class.py

6.56 skdaccess.framework.data_class.XArrayWrapper Class Reference

Wrapper for xarrays.

Inheritance diagram for skdaccess.framework.data_class.XArrayWrapper:



Public Member Functions

- def __init__ (self, obj_wrap, index_list, run_id=-1)
- def getIterator (self)

  Get an iterator that iterators over the index.
- def info (self, key=None)

  Get information about xarray data wrapper.
- def update (self, obj)

  Updated wrapped data.
- def updateMetadata (self, new_metadata)

  Update metadata.
- def get (self)

  Retrieve stored data.
- def getResults (self)

  Retrieve accumulated results, if any.
- def addResult (self, rkey, rres)

  Add a result to the data wrapper.
- def reset (self)

  Reset data back to original state.
- def __len__ (self)

  Get length of wrapped data.
- def getRunID (self)

  Get the Run ID.

Public Attributes

- index_list
- data
- results
- constants
- run_id
- meta_data

6.56.1 Detailed Description

Wrapper for xarrays.

6.56.2 Constructor & Destructor Documentation

6.56.2.1 __init__()

```
def skdaccess.framework.data_class.XArrayWrapper.__init__ (
              self,
              obj_wrap,
              index_list,
              run_id = -1 )
```

6.56.3 Member Function Documentation

6.56.3.1 __len__()

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
              self )  [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.56.3.2 addResult()

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
              self,
              rkey,
              rres )  [inherited]
```

Add a result to the data wrapper.

Parameters

rkey	Result key
rres	Result

6.56.3.3 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
              self )  [inherited]
```

Retrieve stored data.

Returns

Stored data

6.56.3.4 getIterator()

```
def skdaccess.framework.data_class.XArrayWrapper.getIterator (
              self )
```

Get an iterator that iterators over the index.

Returns

iterator to data

6.56.3.5 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
              self )  [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.56.3.6 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
              self )  [inherited]
```

Get the Run ID.

Returns

run_id

6.56.3.7 info()

```
def skdaccess.framework.data_class.XArrayWrapper.info (
              self,
              key = None )
```

Get information about xarray data wrapper.

Returns

The stored metadata

6.56.3.8 reset()

```
def skdaccess.framework.data_class.DataWrapperBase.reset (
              self )  [inherited]
```

Reset data back to original state.

6.56.3.9 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
              self,
              obj )  [inherited]
```

Updated wrapped data.

Parameters

obj	New data for wrapper

6.56.3.10 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
              self,
              new_metadata )  [inherited]
```

Update metadata.

Parameters

new_metadata	New metadata

6.56.4 Member Data Documentation

6.56.4.1 constants

```
skdaccess.framework.data_class.DataWrapperBase.constants  [inherited]
```

6.56.4.2 data

```
skdaccess.framework.data_class.DataWrapperBase.data  [inherited]
```

6.56.4.3 index_list

```
skdaccess.framework.data_class.XArrayWrapper.index_list
```

6.56.4.4 meta_data

```
skdaccess.framework.data_class.DataWrapperBase.meta_data  [inherited]
```

6.56.4.5 results

```
skdaccess.framework.data_class.DataWrapperBase.results  [inherited]
```

6.56.4.6 run_id

```
skdaccess.framework.data_class.DataWrapperBase.run_id  [inherited]
```

The documentation for this class was generated from the following file:

- framework/data_class.py

Chapter 7

File Documentation

7.1 framework/data_class.py File Reference

Classes

- class skdaccess.framework.data_class.DataFetcherBase

  Base class for all data fetchers.
- class skdaccess.framework.data_class.DataFetcherLocal

  Data fetcher base class for use when storing data locally.
- class skdaccess.framework.data_class.DataFetcherStorage

  Data fetcher base class for use when entire data set is downloaded.
- class skdaccess.framework.data_class.DataFetcherStream

  Data fetcher base class for downloading data into memory.
- class skdaccess.framework.data_class.DataFetcherCache

  Data fetcher base class for downloading data and caching results on hard disk.
- class skdaccess.framework.data_class.DataWrapperBase

  Base class for wrapping data for use in DiscoveryPipeline.
- class skdaccess.framework.data_class.SeriesWrapper

  Data wrapper for series data using a data panel.
- class skdaccess.framework.data_class.SeriesDictionaryWrapper

  Data wrapper for series data using a dictionary of data frames.
- class skdaccess.framework.data_class.TableWrapper

  Data wrapper for table data using an ordered dictionary.
- class skdaccess.framework.data_class.ImageWrapper

  Wrapper for image data.
- class skdaccess.framework.data_class.XArrayWrapper

  Wrapper for xarrays.

Namespaces

- skdaccess.framework.data_class

7.2 framework/param_class.py File Reference

Classes

- class skdaccess.framework.param_class.AutoParam

  Defines a tunable parameter class inherited by specific subclasses.
- class skdaccess.framework.param_class.AutoParamMinMax

  A tunable parameter with min and max ranges, perturbs to a random value in range.
- class skdaccess.framework.param_class.AutoParamList

  A tunable parameter with a specified list of choices that can be randomly selected via perturb.
- class skdaccess.framework.param_class.AutoParamListCycle

  Cycles through a list of paramters.
- class skdaccess.framework.param_class.AutoList

  Specifies a list for returning selections of lists, as opposed to a single element.
- class skdaccess.framework.param_class.AutoListSubset

  An AutoList perturber that creates random subsets of a list.
- class skdaccess.framework.param_class.AutoListPermute

  A perturber that permutes a list.
- class skdaccess.framework.param_class.AutoListRemove

  Removes a different single element from the initial list at each perturb call.
- class skdaccess.framework.param_class.AutoListCycle

  An Autolist that cycles through different lists.

Namespaces

- skdaccess.framework.param_class

7.3 geo/mahali/rinex/data_wrapper.py File Reference

Classes

- class skdaccess.geo.mahali.rinex.data_wrapper.DataWrapper

  Data wrapper for Mahali data.

Namespaces

- skdaccess.geo.mahali.rinex.data_wrapper

7.4 solar/sdo/data_fetcher.py File Reference

Classes

- class skdaccess.solar.sdo.DataFetcher

  Data Fetcher for Mahali temperature data.

Namespaces

- skdaccess.solar.sdo.data_fetcher

7.5 planetary/ode/cache/data_fetcher.py File Reference

Classes

- class skdaccess.planetary.ode.cache.DataFetcher

  Data Fetcher from the Orbital Data Explorer (ODE)

Namespaces

- skdaccess.planetary.ode.cache.data_fetcher

7.6 geo/grace/mascon/cache/data_fetcher.py File Reference

Classes

- class skdaccess.geo.grace.mascon.cache.DataFetcher

  Data Fetcher for GRACE mascon data.

Namespaces

- skdaccess.geo.grace.mascon.cache.data_fetcher

7.7 geo/grace/data_fetcher.py File Reference

Classes

- class skdaccess.geo.grace.DataFetcher

  Data Fetcher for GRACE data.

Namespaces

- skdaccess.geo.grace.data_fetcher

7.8 geo/mahali/tec/data_fetcher.py File Reference

Classes

- class skdaccess.geo.mahali.tec.DataFetcher

  Data Fetcher for Mahali Data.

Namespaces

- skdaccess.geo.mahali.tec.data_fetcher

7.9 geo/mahali/rinex/data_fetcher.py File Reference

Classes

- class skdaccess.geo.mahali.rinex.DataFetcher

  Data Fetcher for Mahali Data.

Namespaces

- skdaccess.geo.mahali.rinex.data_fetcher

7.10 geo/mahali/temperature/data_fetcher.py File Reference

Classes

- class skdaccess.geo.mahali.temperature.DataFetcher

  Data Fetcher for Mahali temperature data.

Namespaces

- skdaccess.geo.mahali.temperature.data_fetcher

7.11 geo/ngl_gps/data_fetcher.py File Reference

Classes

- class skdaccess.geo.ngl_gps.DataFetcher

  Data fetcher for GPS data from Neveda Geodetic Laboratory.

Namespaces

- skdaccess.geo.ngl_gps.data_fetcher

7.12 geo/era_interim/cache/data_fetcher.py File Reference

Classes

- class skdaccess.geo.era_interim.cache.DataFetcher

  DataFetcher for retrieving ERA-I data.

Namespaces

- skdaccess.geo.era_interim.cache.data_fetcher

7.13 geo/imsdnhs/data_fetcher.py File Reference

Classes

- class skdaccess.geo.imsdnhs.DataFetcher

  Fetches data for the Interactive Multisensor Snow and Ice Mapping System Daily Northern Hemisphere Snow and Ice Analysis.

Namespaces

- skdaccess.geo.imsdnhs.data_fetcher

7.14 geo/gldas/data_fetcher.py File Reference

Classes

- class skdaccess.geo.gldas.DataFetcher

  Data Fetcher for GLDAS data.

Namespaces

- skdaccess.geo.gldas.data_fetcher

7.15 geo/sentinel_1/cache/data_fetcher.py File Reference

Classes

- class skdaccess.geo.sentinel_1.cache.DataFetcher

  DataFetcher for retrieving Sentinel SLC data.

Namespaces

- skdaccess.geo.sentinel_1.cache.data_fetcher

7.16 geo/magnetometer/data_fetcher.py File Reference

Classes

- class skdaccess.geo.magnetometer.DataFetcher

  Data fetcher for USGS geomagnetic observatories.

Namespaces

- skdaccess.geo.magnetometer.data_fetcher

7.17 geo/wyoming_sounding/cache/data_fetcher.py File Reference

Classes

- class skdaccess.geo.wyoming_sounding.cache.DataFetcher

  DataFetcher for retrieving Wyoming Sounding data.

Namespaces

- skdaccess.geo.wyoming_sounding.cache.data_fetcher

7.18 geo/wyoming_sounding/stream/data_fetcher.py File Reference

Classes

- class skdaccess.geo.wyoming_sounding.stream.DataFetcher

  DataFetcher for retrieving Wyoming Sounding data.

Namespaces

- skdaccess.geo.wyoming_sounding.stream.data_fetcher

7.19 geo/modis/cache/cloud_opacity/data_fetcher.py File Reference

Classes

- class skdaccess.geo.modis.cache.cloud_opacity.DataFetcher

  Data Fetcher for MODIS Cloud Opacity.

Namespaces

- skdaccess.geo.modis.cache.cloud_opacity.data_fetcher

7.20 geo/modis/cache/cloud_mask/data_fetcher.py File Reference

Classes

- class skdaccess.geo.modis.cache.cloud_mask.DataFetcher

  Data Fetcher for MODIS Cloud Mask.

Namespaces

- skdaccess.geo.modis.cache.cloud_mask.data_fetcher

7.21 geo/modis/cache/reflectance/data_fetcher.py File Reference

Classes

- class skdaccess.geo.modis.cache.reflectance.DataFetcher

  Data fetcher for the modis surface reflectance product ('09', 1 km resolution)

Namespaces

- skdaccess.geo.modis.cache.reflectance.data_fetcher

7.22 geo/modis/cache/data_fetcher.py File Reference

Classes

- class skdaccess.geo.modis.cache.DataFetcher

  Data Fetcher for MODIS data.

Namespaces

- skdaccess.geo.modis.cache.data_fetcher

7.23 geo/modis/stream/cloud_opacity/data_fetcher.py File Reference

Classes

- class skdaccess.geo.modis.stream.cloud_opacity.DataFetcher

  Data Fetcher for MODIS Cloud Opacity.

Namespaces

- skdaccess.geo.modis.stream.cloud_opacity.data_fetcher

7.24 geo/modis/stream/cloud_mask/data_fetcher.py File Reference

Classes

- class skdaccess.geo.modis.stream.cloud_mask.DataFetcher

  Data Fetcher for MODIS Cloud Mask.

Namespaces

- skdaccess.geo.modis.stream.cloud_mask.data_fetcher

7.25 geo/modis/stream/reflectance/data_fetcher.py File Reference

Classes

- class skdaccess.geo.modis.stream.reflectance.DataFetcher

  Data fetcher for the modis surface reflectance product ('09', 1 km resolution)

Namespaces

- skdaccess.geo.modis.stream.reflectance.data_fetcher

7.26 geo/modis/stream/data_fetcher.py File Reference

Classes

- class skdaccess.geo.modis.stream.DataFetcher

  Data Fetcher for MODIS data.

Namespaces

- skdaccess.geo.modis.stream.data_fetcher

7.27 geo/uavsar/cache/data_fetcher.py File Reference

Classes

- class skdaccess.geo.uavsar.cache.DataFetcher

  Data Fetcher for UAVSAR data.

Namespaces

- skdaccess.geo.uavsar.cache.data_fetcher

7.28 geo/srtm/cache/data_fetcher.py File Reference

Classes

- class skdaccess.geo.srtm.cache.DataFetcher

  DataFetcher for retrieving data from the Shuttle Radar Topography Mission.

Namespaces

- skdaccess.geo.srtm.cache.data_fetcher

7.29 geo/groundwater/data_fetcher.py File Reference

Classes

- class skdaccess.geo.groundwater.DataFetcher

  Generates Data Wrappers of groundwater measurements taken in the US.

Namespaces

- skdaccess.geo.groundwater.data_fetcher

7.30 geo/pbo/data_fetcher.py File Reference

Classes

- class skdaccess.geo.pbo.DataFetcher

  Data fetcher for PBO GPS data.

Namespaces

- skdaccess.geo.pbo.data_fetcher

7.31 astro/kepler/data_fetcher.py File Reference

Classes

- class skdaccess.astro.kepler.DataFetcher

  Data Fetcher for Kepler light curve data.

Namespaces

- skdaccess.astro.kepler.data_fetcher

7.32 astro/voyager/data_fetcher.py File Reference

Classes

- class skdaccess.astro.voyager.DataFetcher

  Data Fetcher for Mahali temperature data.

Namespaces

- skdaccess.astro.voyager.data_fetcher

7.33 utilities/file_browser.py File Reference

Classes

- class skdaccess.utilities.file_browser.FileBrowser

Namespaces

- skdaccess.utilities.file_browser

7.34 utilities/grace_util.py File Reference

Namespaces

- skdaccess.utilities.grace_util

Functions

- def skdaccess.utilities.grace_util.averageDates (dates, round_nearest_day=False)

  Compute the average of a pandas series of timestamps.
- def skdaccess.utilities.grace_util.dateMismatch (dates, days=10)

  Check if dates are not within a certain number of days of each other.
- def skdaccess.utilities.grace_util.computeEWD (grace_data, scale_factor, round_nearest_day=False)

  Compute scale corrected equivalent water depth.
- def skdaccess.utilities.grace_util.readTellusData (filename, lat_lon_list, lat_name, lon_name, data_name, data↩_label=None, time_name=None, lat_bounds_name=None, lon_bounds_name=None, uncertainty_name=None, lat_bounds=None, lon_bounds=None)

  This function reads in netcdf data provided by GRACE Tellus.
- def skdaccess.utilities.grace_util.getStartEndDate (in_data)

7.35 utilities/gw_util.py File Reference

Namespaces

- skdaccess.utilities.gw_util

Functions

- def skdaccess.utilities.gw_util.combine_water_heights (in_data)

  Combine median and average water heights.

7.36 utilities/image_util.py File Reference

Classes

- class skdaccess.utilities.image_util.SplineLatLon

  Holds a 2d spline for interpolating lat/lon grid.
- class skdaccess.utilities.image_util.LinearGeolocation

  This class provides functions to convert between pixel and geodetic coordinates.
- class skdaccess.utilities.image_util.AffineGlobalCoords

  Convert between projected and pixel coordinates using an affine transformation.

Namespaces

- skdaccess.utilities.image_util

Functions

- def skdaccess.utilities.image_util.SplineGeolocation (object)

  This class holds splines to convert between 2d cartesian and geodetic coordinates.
- def skdaccess.utilities.image_util.getExtentsFromCentersPlateCarree (westmost_pixel_lon, eastmost_pixel_lon, southmost_pixel_lat, northmost_pixel_lat, lon_grid_spacing, lat_grid_spacing)
- def skdaccess.utilities.image_util.convertBinCentersToEdges (bin_centers, dtype=None)

  Calculate edges of a set of bins from their centers.
- def skdaccess.utilities.image_util.getGeoTransform (extents, x_size, y_size, y_flipped=True)

  Get 6 geotransform coefficients from the extents of an image and its shape.

Variables

- skdaccess.utilities.image_util.x_offset
- skdaccess.utilities.image_util.y_offset
- skdaccess.utilities.image_util.lat_spline
- skdaccess.utilities.image_util.lon_spline
- skdaccess.utilities.image_util.x_spline
- skdaccess.utilities.image_util.y_spline

7.37 utilities/kepler_util.py File Reference

Namespaces

- skdaccess.utilities.kepler_util

Functions

- def skdaccess.utilities.kepler_util.normalize (in_data, column='PDCSAP_FLUX', group_column='QUARTER')

  This function normalizes PDCSAP_FLUX data by quarter by dividing the flux by the median for the quarter.

7.38 utilities/mahali_util.py File Reference

Namespaces

- skdaccess.utilities.mahali_util

Functions

- def skdaccess.utilities.mahali_util.convert_date (in_date)

  Converts input string to pandas date time, ignores other types of objects.
- def skdaccess.utilities.mahali_util.parseIonoFile (in_file, compression='infer')

7.39 utilities/modis_util.py File Reference

Classes

- class skdaccess.utilities.modis_util.LatLon

  Calculates Lat/Lon position from y,x pixel coordinate.

Namespaces

- skdaccess.utilities.modis_util

Functions

- def skdaccess.utilities.modis_util.getImageType (in_data)

  Determine what type of modis data is being processed.
- def skdaccess.utilities.modis_util.calibrateModis (data, metadata)

  This function calibrates input modis data.
- def skdaccess.utilities.modis_util.rescale (in_array, max_val=0.9, min_val=-0.01)

  This function rescales an image to fall between 0 and 1.
- def skdaccess.utilities.modis_util.checkBit (data, bit)

  Get the bit value from a bit flag.
- def skdaccess.utilities.modis_util.createGrid (data, y_start, y_end, x_start, x_end, y_grid, x_grid, dtype, grid_↩ fill=np.nan)

  Subsets image data into a smaller image.
- def skdaccess.utilities.modis_util.getFileIDs (modis_identifier, start_date, end_date, lat, lon, daynightboth)

  Retrieve file IDs for images matching search parameters.
- def skdaccess.utilities.modis_util.getFileURLs (file_ids)

  Retrieve the ftp location for a list of file IDs.
- def skdaccess.utilities.modis_util.getModisData (dataset, variable_name)

  Loads modis data.
- def skdaccess.utilities.modis_util.readMODISData (modis_list, variables, grid, grid_fill, use_long_name, platform, product_id)

  Retrieve a list of modis data.

7.40 utilities/ode_util.py File Reference

Namespaces

- skdaccess.utilities.ode_util

Functions

- def skdaccess.utilities.ode_util.query_yes_no (question, default="yes")
- def skdaccess.utilities.ode_util.get_query_url (target, mission, instrument, product_type, western_lon, eastern_↩ lon, min_lat, max_lat, min_ob_time, max_ob_time, product_id, query_type, output, results, number_product_limit, result_offset_number)
- def skdaccess.utilities.ode_util.get_files_urls (query_url, file_name=' ∗', print_info=False)
- def skdaccess.utilities.ode_util.query_files_urls (target, mission, instrument, product_type, western_lon, eastern_lon, min_lat, max_lat, min_ob_time, max_ob_time, product_id, file_name, number_product_limit, result_offset_number)

  Retrieve the URL locations based on a query using ODE REST interface.
- def skdaccess.utilities.ode_util.correct_CRISM_label (label_file_location)
- def skdaccess.utilities.ode_util.correct_file_name_case_in_label (label_file_location, other_file_locations)
- def skdaccess.utilities.ode_util.correct_label_file (label_file_location, other_file_locations=[ ])

  Correct a label file if GDAL cannot open the corresponding data file.
- def skdaccess.utilities.ode_util.get_raster_array (gdal_raster, remove_ndv=True)

  Get a NumPy array from a raster opened with GDAL.
- def skdaccess.utilities.ode_util.get_raster_extent (gdal_raster)

  Get the extent of a raster opened with GDAL.

7.41 utilities/pbo_util.py File Reference

Namespaces

- skdaccess.utilities.pbo_util

Functions

- def skdaccess.utilities.pbo_util.getStationCoords (pbo_info, station_list)

  Get the station coordinates for a list of stations.
- def skdaccess.utilities.pbo_util.getLatLonRange (pbo_info, station_list)

  Retrive the range of latitude and longitude occupied by a set of stations.
- def skdaccess.utilities.pbo_util.getROIstations (geo_point, radiusParam, data, header)

  This function returns the 4ID station codes for the stations in a region.
- def skdaccess.utilities.pbo_util.stab_sys (data_iterator, metadata, stab_min_NE=.0005, stab_min_U=.005, sigsc=2, errProp=1)

  Stabilize GPS data to a region.
- def skdaccess.utilities.pbo_util.propagateErrors (R, sc, stationCovs)

  Propagate GPS errors.
- def skdaccess.utilities.pbo_util.nostab_sys (allH, allD, timerng, indx=1, mdyratio=.7, use_progress_bar=True, index_date_only=False)

  Do not apply stabilization and simply returns stations after checking for sufficient amount of data.
- def skdaccess.utilities.pbo_util.removeAntennaOffset (antenna_offsets, data, window_start=pd.to_timedelta('4↩ D'), window_end=pd.to_timedelta('4D'), min_diff=0.005, debug=False)

  Remove offsets caused by changes in antennas.

7.42 utilities/sentinel_1_util.py File Reference

Namespaces

- skdaccess.utilities.sentinel_1_util

Functions

- def skdaccess.utilities.sentinel_1_util.parseSatelliteData (in_satellite_file)

  Parse Sentinel satelllite data.

7.43 utilities/sounding_util.py File Reference

Classes

- class skdaccess.utilities.sounding_util.SoundingParser

  This class parses Wyoming Sounding data.

Namespaces

- skdaccess.utilities.sounding_util

Functions

- def skdaccess.utilities.sounding_util.generateQueries (station_number, year_list, month_list, day_start, day_end, start_hour, end_hour)

  Generate url queries for sounding data.

7.44 utilities/srtm_util.py File Reference

Namespaces

- skdaccess.utilities.srtm_util

Functions

- def skdaccess.utilities.srtm_util.merge_srtm_tiles (srtm_tiles, lon_min, lon_max, lat_min, lat_max)
- def skdaccess.utilities.srtm_util.getSRTMLatLon (lat_min, lat_max, lon_min, lon_max)

  Retrieve parameters that encompass area when creating SRTM data fetcher.
- def skdaccess.utilities.srtm_util.getSRTMData (srtmdw, lat_start, lat_end, lon_start, lon_end)

  Select SRTM data in a latitude/longitude box.

7.45 utilities/support.py File Reference

Namespaces

- skdaccess.utilities.support

Functions

- def skdaccess.utilities.support.retrieveCommonDatesHDF (support_data_filename, key_list, in_date_list)

  Get a list of all dates that have data available.
- def skdaccess.utilities.support.progress_bar (in_iterable, total=None, enabled=True)

  Progess bar using tqdm.
- def skdaccess.utilities.support.convertToStr (in_value, zfill=0)

7.46 utilities/uavsar_util.py File Reference

Namespaces

- skdaccess.utilities.uavsar_util

Functions

- def skdaccess.utilities.uavsar_util.readUAVSARMetadata (in_file)

  Parse UAVSAR metadata.

Index