

Scikit Data Access

Generated by Doxygen 1.8.13

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	5
2.1	Class Hierarchy	5
3	Class Index	7
3.1	Class List	7
4	File Index	11
4.1	File List	11
5	Namespace Documentation	13
5.1	skdaccess Namespace Reference	13
5.2	skdaccess.astro Namespace Reference	13
5.3	skdaccess.astro.kepler Namespace Reference	13
5.4	skdaccess.astro.kepler.data_fetcher Namespace Reference	14
5.5	skdaccess.astro.spectra Namespace Reference	14
5.6	skdaccess.astro.spectra.stream Namespace Reference	14
5.7	skdaccess.astro.voyager Namespace Reference	14
5.8	skdaccess.astro.voyager.data_fetcher Namespace Reference	14
5.9	skdaccess.engineering Namespace Reference	14
5.10	skdaccess.engineering.la Namespace Reference	15

5.11	skdaccess.engineering.la.generic Namespace Reference	15
5.12	skdaccess.engineering.la.generic.stream Namespace Reference	15
5.13	skdaccess.engineering.la.traffic_counts Namespace Reference	15
5.14	skdaccess.engineering.la.traffic_counts.stream Namespace Reference	15
5.15	skdaccess.engineering.webcam Namespace Reference	15
5.16	skdaccess.engineering.webcam.mit_sailing Namespace Reference	16
5.17	skdaccess.engineering.webcam.mit_sailing.stream Namespace Reference	16
5.18	skdaccess.finance Namespace Reference	16
5.19	skdaccess.finance.timeseries Namespace Reference	16
5.20	skdaccess.finance.timeseries.stream Namespace Reference	16
5.21	skdaccess.framework Namespace Reference	16
5.22	skdaccess.framework.data_class Namespace Reference	17
5.23	skdaccess.framework.param_class Namespace Reference	17
5.24	skdaccess.geo Namespace Reference	18
5.25	skdaccess.geo.era_interim Namespace Reference	18
5.26	skdaccess.geo.era_interim.cache Namespace Reference	18
5.27	skdaccess.geo.era_interim.cache.data_fetcher Namespace Reference	18
5.28	skdaccess.geo.gldas Namespace Reference	18
5.29	skdaccess.geo.gldas.data_fetcher Namespace Reference	19
5.30	skdaccess.geo.grace Namespace Reference	19
5.31	skdaccess.geo.grace.data_fetcher Namespace Reference	19
5.32	skdaccess.geo.grace.mascon Namespace Reference	19
5.33	skdaccess.geo.grace.mascon.cache Namespace Reference	19
5.34	skdaccess.geo.grace.mascon.cache.data_fetcher Namespace Reference	19
5.35	skdaccess.geo.groundwater Namespace Reference	20
5.36	skdaccess.geo.groundwater.data_fetcher Namespace Reference	20
5.37	skdaccess.geo.imsdnhs Namespace Reference	20
5.38	skdaccess.geo.imsdnhs.data_fetcher Namespace Reference	20

5.39	skdaccess.geo.magnetometer Namespace Reference	20
5.40	skdaccess.geo.magnetometer.data_fetcher Namespace Reference	20
5.41	skdaccess.geo.mahali Namespace Reference	21
5.42	skdaccess.geo.mahali.rinex Namespace Reference	21
5.43	skdaccess.geo.mahali.rinex.data_fetcher Namespace Reference	21
5.44	skdaccess.geo.mahali.rinex.data_wrapper Namespace Reference	21
5.45	skdaccess.geo.mahali.tec Namespace Reference	21
5.46	skdaccess.geo.mahali.tec.data_fetcher Namespace Reference	21
5.47	skdaccess.geo.mahali.temperature Namespace Reference	22
5.48	skdaccess.geo.mahali.temperature.data_fetcher Namespace Reference	22
5.49	skdaccess.geo.modis Namespace Reference	22
5.50	skdaccess.geo.modis.cache Namespace Reference	22
5.51	skdaccess.geo.modis.cache.cloud_mask Namespace Reference	22
5.52	skdaccess.geo.modis.cache.cloud_mask.data_fetcher Namespace Reference	22
5.53	skdaccess.geo.modis.cache.cloud_opacity Namespace Reference	23
5.54	skdaccess.geo.modis.cache.cloud_opacity.data_fetcher Namespace Reference	23
5.55	skdaccess.geo.modis.cache.data_fetcher Namespace Reference	23
5.56	skdaccess.geo.modis.cache.reflectance Namespace Reference	23
5.57	skdaccess.geo.modis.cache.reflectance.data_fetcher Namespace Reference	23
5.58	skdaccess.geo.modis.stream Namespace Reference	23
5.59	skdaccess.geo.modis.stream.cloud_mask Namespace Reference	24
5.60	skdaccess.geo.modis.stream.cloud_mask.data_fetcher Namespace Reference	24
5.61	skdaccess.geo.modis.stream.cloud_opacity Namespace Reference	24
5.62	skdaccess.geo.modis.stream.cloud_opacity.data_fetcher Namespace Reference	24
5.63	skdaccess.geo.modis.stream.data_fetcher Namespace Reference	24
5.64	skdaccess.geo.modis.stream.reflectance Namespace Reference	24
5.65	skdaccess.geo.modis.stream.reflectance.data_fetcher Namespace Reference	25
5.66	skdaccess.geo.ngl_gps Namespace Reference	25

5.67 skdaccess.geo.ngl_gps.data_fetcher Namespace Reference	25
5.68 skdaccess.geo.pbo Namespace Reference	25
5.69 skdaccess.geo.pbo.data_fetcher Namespace Reference	25
5.70 skdaccess.geo.sentinel_1 Namespace Reference	25
5.71 skdaccess.geo.sentinel_1.cache Namespace Reference	26
5.72 skdaccess.geo.sentinel_1.cache.data_fetcher Namespace Reference	26
5.73 skdaccess.geo.srtm Namespace Reference	26
5.74 skdaccess.geo.srtm.cache Namespace Reference	26
5.75 skdaccess.geo.srtm.cache.data_fetcher Namespace Reference	26
5.76 skdaccess.geo.uavsar Namespace Reference	26
5.77 skdaccess.geo.uavsar.cache Namespace Reference	27
5.78 skdaccess.geo.uavsar.cache.data_fetcher Namespace Reference	27
5.79 skdaccess.geo.wyoming_sounding Namespace Reference	27
5.80 skdaccess.geo.wyoming_sounding.cache Namespace Reference	27
5.81 skdaccess.geo.wyoming_sounding.cache.data_fetcher Namespace Reference	27
5.82 skdaccess.geo.wyoming_sounding.stream Namespace Reference	27
5.83 skdaccess.geo.wyoming_sounding.stream.data_fetcher Namespace Reference	28
5.84 skdaccess.planetary Namespace Reference	28
5.85 skdaccess.planetary.ode Namespace Reference	28
5.86 skdaccess.planetary.ode.cache Namespace Reference	28
5.87 skdaccess.planetary.ode.cache.data_fetcher Namespace Reference	28
5.88 skdaccess.solar Namespace Reference	28
5.89 skdaccess.solar.sdo Namespace Reference	29
5.90 skdaccess.solar.sdo.data_fetcher Namespace Reference	29
5.91 skdaccess.utilities Namespace Reference	29
5.92 skdaccess.utilities.file_browser Namespace Reference	29
5.93 skdaccess.utilities.file_util Namespace Reference	30
5.93.1 Function Documentation	30

5.93.1.1	openPandasHDFStoreLocking()	30
5.94	skdaccess.utilities.grace_util Namespace Reference	30
5.94.1	Function Documentation	31
5.94.1.1	averageDates()	31
5.94.1.2	computeEWD()	31
5.94.1.3	dateMismatch()	32
5.94.1.4	getStartEndDate()	32
5.94.1.5	readTellusData()	32
5.95	skdaccess.utilities.gw_util Namespace Reference	33
5.95.1	Function Documentation	33
5.95.1.1	combine_water_heights()	33
5.96	skdaccess.utilities.image_util Namespace Reference	34
5.96.1	Function Documentation	34
5.96.1.1	convertBinCentersToEdges()	34
5.96.1.2	getExtentsFromCentersPlateCarree()	35
5.96.1.3	getGeoTransform()	35
5.96.1.4	SplineGeolocation()	36
5.96.2	Variable Documentation	36
5.96.2.1	lat_spline	36
5.96.2.2	lon_spline	36
5.96.2.3	x_offset	36
5.96.2.4	x_spline	36
5.96.2.5	y_offset	36
5.96.2.6	y_spline	37
5.97	skdaccess.utilities.kepler_util Namespace Reference	37
5.97.1	Function Documentation	37
5.97.1.1	normalize()	37
5.98	skdaccess.utilities.mahali_util Namespace Reference	37

5.98.1	Function Documentation	37
5.98.1.1	convert_date()	38
5.98.1.2	parselonoFile()	38
5.99	skdaccess.utilities.modis_util Namespace Reference	38
5.99.1	Function Documentation	39
5.99.1.1	calibrateModis()	39
5.99.1.2	checkBit()	39
5.99.1.3	createGrid()	39
5.99.1.4	getFileIDs()	40
5.99.1.5	getFileURLs()	41
5.99.1.6	getImageType()	41
5.99.1.7	getModisData()	42
5.99.1.8	readMODISData()	42
5.99.1.9	rescale()	43
5.100	skdaccess.utilities.ode_util Namespace Reference	43
5.100.1	Function Documentation	43
5.100.1.1	correct_CRISM_label()	44
5.100.1.2	correct_file_name_case_in_label()	44
5.100.1.3	correct_label_file()	44
5.100.1.4	get_files_urls()	44
5.100.1.5	get_query_url()	45
5.100.1.6	get_raster_array()	45
5.100.1.7	get_raster_extent()	45
5.100.1.8	query_files_urls()	46
5.100.1.9	query_yes_no()	47
5.101	skdaccess.utilities.pbo_util Namespace Reference	47
5.101.1	Function Documentation	47
5.101.1.1	getLatLonRange()	47

5.101.1.2 getROIstations()	48
5.101.1.3 getStationCoords()	48
5.101.1.4 nostab_sys()	49
5.101.1.5 propagateErrors()	49
5.101.1.6 removeAntennaOffset()	50
5.101.1.7 stab_sys()	50
5.102skdaccess.utilities.sentinel_1_util Namespace Reference	51
5.102.1 Function Documentation	51
5.102.1.1 parseSatelliteData()	51
5.103skdaccess.utilities.sounding_util Namespace Reference	52
5.103.1 Function Documentation	52
5.103.1.1 generateQueries()	52
5.104skdaccess.utilities.srtm_util Namespace Reference	53
5.104.1 Function Documentation	53
5.104.1.1 getSRTMData()	53
5.104.1.2 getSRTMLatLon()	54
5.104.1.3 merge_srtm_tiles()	54
5.105skdaccess.utilities.support Namespace Reference	54
5.105.1 Function Documentation	55
5.105.1.1 convertToStr()	55
5.105.1.2 join_string()	55
5.105.1.3 progress_bar()	55
5.105.1.4 retrieveCommonDatesHDF()	56
5.106skdaccess.utilities.uavsar_util Namespace Reference	56
5.106.1 Function Documentation	56
5.106.1.1 readUAVSARMetadata()	56

6	Class Documentation	59
6.1	skdaccess.utilities.image_util.AffineGlobalCoords Class Reference	59
6.1.1	Detailed Description	59
6.1.2	Constructor & Destructor Documentation	59
6.1.2.1	__init__()	60
6.1.3	Member Function Documentation	60
6.1.3.1	getPixelYX()	60
6.1.3.2	getProjectedYX()	60
6.2	skdaccess.framework.param_class.AutoList Class Reference	61
6.2.1	Detailed Description	62
6.2.2	Constructor & Destructor Documentation	62
6.2.2.1	__init__()	62
6.2.3	Member Function Documentation	62
6.2.3.1	__call__()	62
6.2.3.2	__getitem__()	63
6.2.3.3	__len__()	63
6.2.3.4	__setitem__()	63
6.2.3.5	__str__()	64
6.2.3.6	getAllOptions()	64
6.2.3.7	perturb()	64
6.2.3.8	reset()	64
6.2.3.9	val()	65
6.2.4	Member Data Documentation	65
6.2.4.1	val_init	65
6.2.4.2	val_list	65
6.3	skdaccess.framework.param_class.AutoListCycle Class Reference	65
6.3.1	Detailed Description	66
6.3.2	Constructor & Destructor Documentation	66

6.3.2.1	<code>__init__()</code>	66
6.3.3	Member Function Documentation	67
6.3.3.1	<code>__call__()</code>	67
6.3.3.2	<code>__getitem__()</code>	67
6.3.3.3	<code>__len__()</code>	67
6.3.3.4	<code>__setitem__()</code>	68
6.3.3.5	<code>__str__()</code>	68
6.3.3.6	<code>getAllOptions()</code>	68
6.3.3.7	<code>perturb()</code>	69
6.3.3.8	<code>reset()</code>	69
6.3.3.9	<code>val()</code>	69
6.3.4	Member Data Documentation	69
6.3.4.1	<code>index</code>	69
6.3.4.2	<code>list_val_list</code>	69
6.3.4.3	<code>val_init</code>	70
6.3.4.4	<code>val_list</code>	70
6.4	<code>skdaccess.framework.param_class.AutoListPermute</code> Class Reference	70
6.4.1	Detailed Description	71
6.4.2	Member Function Documentation	71
6.4.2.1	<code>__call__()</code>	71
6.4.2.2	<code>__getitem__()</code>	71
6.4.2.3	<code>__len__()</code>	72
6.4.2.4	<code>__setitem__()</code>	72
6.4.2.5	<code>__str__()</code>	72
6.4.2.6	<code>getAllOptions()</code>	73
6.4.2.7	<code>perturb()</code>	73
6.4.2.8	<code>reset()</code>	73
6.4.2.9	<code>val()</code>	73

6.4.3	Member Data Documentation	73
6.4.3.1	val_init	74
6.4.3.2	val_list	74
6.5	skdaccess.framework.param_class.AutoListRemove Class Reference	74
6.5.1	Detailed Description	75
6.5.2	Constructor & Destructor Documentation	75
6.5.2.1	__init__()	75
6.5.3	Member Function Documentation	75
6.5.3.1	__call__()	75
6.5.3.2	__getitem__()	76
6.5.3.3	__len__()	76
6.5.3.4	__setitem__()	76
6.5.3.5	__str__()	77
6.5.3.6	getAllOptions()	77
6.5.3.7	perturb()	77
6.5.3.8	reset()	77
6.5.3.9	val()	78
6.5.4	Member Data Documentation	78
6.5.4.1	n	78
6.5.4.2	val_init	78
6.5.4.3	val_list	78
6.6	skdaccess.framework.param_class.AutoListSubset Class Reference	78
6.6.1	Detailed Description	79
6.6.2	Member Function Documentation	79
6.6.2.1	__call__()	79
6.6.2.2	__getitem__()	80
6.6.2.3	__len__()	80
6.6.2.4	__setitem__()	80

6.6.2.5	__str__()	81
6.6.2.6	getAllOptions()	81
6.6.2.7	perturb()	81
6.6.2.8	reset()	81
6.6.2.9	val()	82
6.6.3	Member Data Documentation	82
6.6.3.1	val_init	82
6.6.3.2	val_list	82
6.7	skdaccess.framework.param_class.AutoParam Class Reference	82
6.7.1	Detailed Description	83
6.7.2	Constructor & Destructor Documentation	83
6.7.2.1	__init__()	83
6.7.3	Member Function Documentation	84
6.7.3.1	__call__()	84
6.7.3.2	__str__()	84
6.7.3.3	perturb()	84
6.7.3.4	reset()	84
6.7.4	Member Data Documentation	85
6.7.4.1	val	85
6.7.4.2	val_init	85
6.8	skdaccess.framework.param_class.AutoParamList Class Reference	85
6.8.1	Detailed Description	86
6.8.2	Constructor & Destructor Documentation	86
6.8.2.1	__init__()	86
6.8.3	Member Function Documentation	86
6.8.3.1	__call__()	86
6.8.3.2	__str__()	87
6.8.3.3	perturb()	87

6.8.3.4	reset()	87
6.8.4	Member Data Documentation	87
6.8.4.1	val	87
6.8.4.2	val_init	87
6.8.4.3	val_list	88
6.9	skdaccess.framework.param_class.AutoParamListCycle Class Reference	88
6.9.1	Detailed Description	88
6.9.2	Constructor & Destructor Documentation	89
6.9.2.1	__init__()	89
6.9.3	Member Function Documentation	89
6.9.3.1	__call__()	89
6.9.3.2	__str__()	89
6.9.3.3	perturb()	90
6.9.3.4	reset()	90
6.9.4	Member Data Documentation	90
6.9.4.1	current_index	90
6.9.4.2	val	90
6.9.4.3	val_init	90
6.9.4.4	val_list	90
6.10	skdaccess.framework.param_class.AutoParamMinMax Class Reference	91
6.10.1	Detailed Description	91
6.10.2	Constructor & Destructor Documentation	91
6.10.2.1	__init__()	92
6.10.3	Member Function Documentation	92
6.10.3.1	__call__()	92
6.10.3.2	__str__()	92
6.10.3.3	perturb()	93
6.10.3.4	reset()	93

6.10.4	Member Data Documentation	93
6.10.4.1	decimals	93
6.10.4.2	n	93
6.10.4.3	n_max	93
6.10.4.4	val	93
6.10.4.5	val_init	94
6.10.4.6	val_max	94
6.10.4.7	val_min	94
6.11	skdaccess.astro.kepler.DataFetcher Class Reference	94
6.11.1	Detailed Description	96
6.11.2	Constructor & Destructor Documentation	96
6.11.2.1	__init__()	96
6.11.3	Member Function Documentation	96
6.11.3.1	__str__()	96
6.11.3.2	cacheData() [1/2]	96
6.11.3.3	cacheData() [2/2]	97
6.11.3.4	checkIfDataExists()	97
6.11.3.5	downloadKeplerData()	98
6.11.3.6	getConfig()	98
6.11.3.7	getConfigItem()	98
6.11.3.8	getDataLocation()	99
6.11.3.9	getHDFStorage()	99
6.11.3.10	getMetadata()	100
6.11.3.11	multirun_enabled()	100
6.11.3.12	output()	100
6.11.3.13	perturb()	100
6.11.3.14	reset()	101
6.11.3.15	setDataLocation()	101

6.11.3.16 verbose_print()	101
6.11.3.17 writeConfig()	101
6.11.3.18 writeConfigItem()	102
6.11.4 Member Data Documentation	102
6.11.4.1 ap_paramList	102
6.11.4.2 quarter_list	102
6.11.4.3 verbose	103
6.12 skdaccess.engineering.la.generic.stream.DataFetcher Class Reference	103
6.12.1 Detailed Description	104
6.12.2 Constructor & Destructor Documentation	104
6.12.2.1 __init__()	104
6.12.3 Member Function Documentation	105
6.12.3.1 __str__()	105
6.12.3.2 getConfig()	105
6.12.3.3 getConfigItem()	105
6.12.3.4 getMetadata()	106
6.12.3.5 multirun_enabled()	106
6.12.3.6 output()	106
6.12.3.7 perturb()	106
6.12.3.8 reset()	107
6.12.3.9 retrieveOnlineData()	107
6.12.3.10 verbose_print()	107
6.12.3.11 writeConfig()	107
6.12.3.12 writeConfigItem()	108
6.12.4 Member Data Documentation	108
6.12.4.1 ap_paramList	108
6.12.4.2 app_token	108
6.12.4.3 base_url	109

6.12.4.4	base_url_and_endpoint	109
6.12.4.5	label	109
6.12.4.6	pandas_kwargs	109
6.12.4.7	parameters	109
6.12.4.8	verbose	109
6.13	skdaccess.geo.gldas.DataFetcher Class Reference	110
6.13.1	Detailed Description	111
6.13.2	Constructor & Destructor Documentation	111
6.13.2.1	__init__()	111
6.13.3	Member Function Documentation	111
6.13.3.1	__str__()	111
6.13.3.2	downloadFullDataset()	112
6.13.3.3	getConfig()	112
6.13.3.4	getConfigItem()	112
6.13.3.5	getDataLocation()	113
6.13.3.6	getMetadata()	113
6.13.3.7	multirun_enabled()	114
6.13.3.8	output()	114
6.13.3.9	perturb()	114
6.13.3.10	reset()	114
6.13.3.11	setDataLocation()	114
6.13.3.12	verbose_print()	115
6.13.3.13	writeConfig()	115
6.13.3.14	writeConfigItem()	115
6.13.4	Member Data Documentation	116
6.13.4.1	ap_paramList	116
6.13.4.2	end_date	116
6.13.4.3	resample	116

6.13.4.4	<code>start_date</code>	116
6.13.4.5	<code>verbose</code>	116
6.14	<code>skdaccess.geo.sentinel_1.cache.DataFetcher Class Reference</code>	117
6.14.1	Detailed Description	118
6.14.2	Constructor & Destructor Documentation	118
6.14.2.1	<code>__init__()</code>	118
6.14.3	Member Function Documentation	119
6.14.3.1	<code>__str__()</code>	119
6.14.3.2	<code>cacheData()</code>	119
6.14.3.3	<code>checkIfDataExists()</code>	120
6.14.3.4	<code>getConfig()</code>	120
6.14.3.5	<code>getConfigItem()</code>	120
6.14.3.6	<code>getDataLocation()</code>	121
6.14.3.7	<code>getHDFStorage()</code>	121
6.14.3.8	<code>getMetadata()</code>	122
6.14.3.9	<code>multirun_enabled()</code>	122
6.14.3.10	<code>output()</code>	122
6.14.3.11	<code>perturb()</code>	122
6.14.3.12	<code>reset()</code>	123
6.14.3.13	<code>setDataLocation()</code>	123
6.14.3.14	<code>verbose_print()</code>	123
6.14.3.15	<code>writeConfig()</code>	123
6.14.3.16	<code>writeConfigItem()</code>	124
6.14.4	Member Data Documentation	124
6.14.4.1	<code>ap_paramList</code>	124
6.14.4.2	<code>local_paths</code>	124
6.14.4.3	<code>password</code>	125
6.14.4.4	<code>polarization</code>	125

6.14.4.5	satellite_url_list	125
6.14.4.6	swath	125
6.14.4.7	url_list	125
6.14.4.8	username	125
6.14.4.9	verbose	125
6.15	skdaccess.geo.magnetometer.DataFetcher Class Reference	126
6.15.1	Detailed Description	127
6.15.2	Constructor & Destructor Documentation	127
6.15.2.1	__init__()	127
6.15.3	Member Function Documentation	127
6.15.3.1	__str__()	128
6.15.3.2	getConfig()	128
6.15.3.3	getConfigItem()	128
6.15.3.4	getDataMetadata()	128
6.15.3.5	getMetadata()	129
6.15.3.6	multirun_enabled()	129
6.15.3.7	output()	129
6.15.3.8	perturb()	130
6.15.3.9	reset()	130
6.15.3.10	retrieveOnlineData()	130
6.15.3.11	verbose_print()	130
6.15.3.12	writeConfig()	131
6.15.3.13	writeConfigItem()	131
6.15.4	Member Data Documentation	131
6.15.4.1	ap_paramList	132
6.15.4.2	channels	132
6.15.4.3	data_type	132
6.15.4.4	end_time	132

6.15.4.5	interval	132
6.15.4.6	start_time	132
6.15.4.7	verbose	132
6.16	skdaccess.geo.wyoming_sounding.cache.DataFetcher Class Reference	133
6.16.1	Detailed Description	134
6.16.2	Constructor & Destructor Documentation	134
6.16.2.1	__init__()	134
6.16.3	Member Function Documentation	135
6.16.3.1	__str__()	135
6.16.3.2	cacheData()	135
6.16.3.3	checkIfDataExists()	136
6.16.3.4	getConfig()	136
6.16.3.5	getConfigItem()	136
6.16.3.6	getDataLocation()	137
6.16.3.7	getHDFStorage()	137
6.16.3.8	getMetadata()	138
6.16.3.9	multirun_enabled()	138
6.16.3.10	output()	138
6.16.3.11	perturb()	138
6.16.3.12	reset()	139
6.16.3.13	setDataLocation()	139
6.16.3.14	verbose_print()	139
6.16.3.15	writeConfig()	139
6.16.3.16	writeConfigItem()	140
6.16.4	Member Data Documentation	140
6.16.4.1	ap_paramList	140
6.16.4.2	day_end	140
6.16.4.3	day_start	141

6.16.4.4	end_hour	141
6.16.4.5	month_list	141
6.16.4.6	start_hour	141
6.16.4.7	station_number	141
6.16.4.8	verbose	141
6.16.4.9	year_list	141
6.17	skdaccess.engineering.la.traffic_counts.stream.DataFetcher Class Reference	142
6.17.1	Detailed Description	142
6.17.2	Constructor & Destructor Documentation	142
6.17.2.1	__init__()	142
6.18	skdaccess.geo.wyoming_sounding.stream.DataFetcher Class Reference	143
6.18.1	Detailed Description	144
6.18.2	Constructor & Destructor Documentation	144
6.18.2.1	__init__()	144
6.18.3	Member Function Documentation	145
6.18.3.1	__str__()	145
6.18.3.2	getConfig()	145
6.18.3.3	getConfigItem()	145
6.18.3.4	getMetadata()	146
6.18.3.5	multirun_enabled()	146
6.18.3.6	output() [1/2]	146
6.18.3.7	output() [2/2]	146
6.18.3.8	perturb()	147
6.18.3.9	reset()	147
6.18.3.10	retrieveOnlineData()	147
6.18.3.11	verbose_print()	147
6.18.3.12	writeConfig()	148
6.18.3.13	writeConfigItem()	148

6.18.4	Member Data Documentation	148
6.18.4.1	ap_paramList	149
6.18.4.2	day_end	149
6.18.4.3	day_start	149
6.18.4.4	end_hour	149
6.18.4.5	month_list	149
6.18.4.6	start_hour	149
6.18.4.7	station_number	149
6.18.4.8	verbose	150
6.18.4.9	year_list	150
6.19	skdaccess.geo.groundwater.DataFetcher Class Reference	150
6.19.1	Detailed Description	151
6.19.2	Constructor & Destructor Documentation	151
6.19.2.1	__init__()	151
6.19.3	Member Function Documentation	152
6.19.3.1	__str__()	152
6.19.3.2	downloadFullDataset()	152
6.19.3.3	getConfig()	153
6.19.3.4	getConfigItem()	153
6.19.3.5	getDataLocation()	153
6.19.3.6	getMetadata()	154
6.19.3.7	getStationMetadata()	154
6.19.3.8	multirun_enabled()	154
6.19.3.9	output()	155
6.19.3.10	perturb()	155
6.19.3.11	reset()	155
6.19.3.12	setDataLocation()	155
6.19.3.13	verbose_print()	156

6.19.3.14 writeConfig()	156
6.19.3.15 writeConfigItem()	156
6.19.4 Member Data Documentation	157
6.19.4.1 ap_paramList	157
6.19.4.2 cutoff	157
6.19.4.3 end_date	157
6.19.4.4 start_date	157
6.19.4.5 verbose	157
6.20 skdaccess.geo.srtm.cache.DataFetcher Class Reference	158
6.20.1 Detailed Description	159
6.20.2 Constructor & Destructor Documentation	159
6.20.2.1 __init__()	159
6.20.3 Member Function Documentation	160
6.20.3.1 __str__()	160
6.20.3.2 cacheData()	160
6.20.3.3 checkIfDataExists()	161
6.20.3.4 getConfig()	161
6.20.3.5 getConfigItem()	161
6.20.3.6 getDataLocation()	162
6.20.3.7 getHDFStorage()	162
6.20.3.8 getMetadata()	163
6.20.3.9 multirun_enabled()	163
6.20.3.10 output()	163
6.20.3.11 perturb()	163
6.20.3.12 reset()	164
6.20.3.13 setDataLocation()	164
6.20.3.14 verbose_print()	164
6.20.3.15 writeConfig()	164

6.20.3.16 writeConfigItem()	165
6.20.4 Member Data Documentation	165
6.20.4.1 ap_paramList	165
6.20.4.2 arcsecond_sampling	165
6.20.4.3 lat_tile_end	166
6.20.4.4 lat_tile_start	166
6.20.4.5 lon_tile_end	166
6.20.4.6 lon_tile_start	166
6.20.4.7 mask_water	166
6.20.4.8 password	167
6.20.4.9 store_geolocation_grids	167
6.20.4.10 username	167
6.20.4.11 verbose	167
6.21 skdaccess.geo.uavsar.cache.DataFetcher Class Reference	167
6.21.1 Detailed Description	169
6.21.2 Constructor & Destructor Documentation	169
6.21.2.1 __init__()	169
6.21.3 Member Function Documentation	169
6.21.3.1 __str__()	169
6.21.3.2 cacheData()	170
6.21.3.3 checkIfDataExists()	170
6.21.3.4 getConfig()	171
6.21.3.5 getConfigItem()	171
6.21.3.6 getDataLocation()	171
6.21.3.7 getHDFStorage()	172
6.21.3.8 getMetadata()	172
6.21.3.9 multirun_enabled()	173
6.21.3.10 output()	173

6.21.3.11 perturb()	173
6.21.3.12 reset()	173
6.21.3.13 setDataLocation()	173
6.21.3.14 verbose_print()	174
6.21.3.15 writeConfig()	174
6.21.3.16 writeConfigItem()	174
6.21.4 Member Data Documentation	175
6.21.4.1 ap_paramList	175
6.21.4.2 llh_url	175
6.21.4.3 memmap	175
6.21.4.4 metadata_url_list	175
6.21.4.5 slc_url_list	175
6.21.4.6 verbose	176
6.22 skdaccess.geo.modis.cache.cloud_opacity.DataFetcher Class Reference	176
6.22.1 Detailed Description	176
6.22.2 Constructor & Destructor Documentation	176
6.22.2.1 __init__()	176
6.23 skdaccess.geo.modis.stream.reflectance.DataFetcher Class Reference	177
6.23.1 Detailed Description	177
6.23.2 Constructor & Destructor Documentation	177
6.23.2.1 __init__()	178
6.24 skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher Class Reference	178
6.24.1 Detailed Description	179
6.24.2 Constructor & Destructor Documentation	179
6.24.2.1 __init__()	179
6.24.3 Member Function Documentation	180
6.24.3.1 __str__()	180
6.24.3.2 getConfig()	180

6.24.3.3	getConfigItem()	180
6.24.3.4	getMetadata()	181
6.24.3.5	multirun_enabled()	181
6.24.3.6	output()	181
6.24.3.7	perturb()	182
6.24.3.8	reset()	182
6.24.3.9	retrieveOnlineData()	182
6.24.3.10	verbose_print()	182
6.24.3.11	writeConfig()	183
6.24.3.12	writeConfigItem()	183
6.24.4	Member Data Documentation	183
6.24.4.1	ap_paramList	184
6.24.4.2	camera_list	184
6.24.4.3	verbose	184
6.25	skdaccess.geo.modis.cache.cloud_mask.DataFetcher Class Reference	184
6.25.1	Detailed Description	184
6.25.2	Constructor & Destructor Documentation	185
6.25.2.1	__init__()	185
6.26	skdaccess.geo.modis.cache.reflectance.DataFetcher Class Reference	185
6.26.1	Detailed Description	186
6.26.2	Constructor & Destructor Documentation	186
6.26.2.1	__init__()	186
6.27	skdaccess.geo.modis.cache.DataFetcher Class Reference	187
6.27.1	Detailed Description	188
6.27.2	Constructor & Destructor Documentation	188
6.27.2.1	__init__()	189
6.27.3	Member Function Documentation	189
6.27.3.1	__str__()	189

6.27.3.2	cacheData() [1/2]	190
6.27.3.3	cacheData() [2/2]	190
6.27.3.4	checkIfDataExists()	191
6.27.3.5	find_data()	191
6.27.3.6	getConfig()	191
6.27.3.7	getConfigItem()	192
6.27.3.8	getDataLocation()	192
6.27.3.9	getHDFSStorage()	192
6.27.3.10	getMetadata()	193
6.27.3.11	multirun_enabled()	193
6.27.3.12	output()	193
6.27.3.13	perturb()	194
6.27.3.14	reset()	194
6.27.3.15	setDataLocation()	194
6.27.3.16	verbose_print()	194
6.27.3.17	writeConfig()	195
6.27.3.18	writeConfigItem()	195
6.27.4	Member Data Documentation	195
6.27.4.1	ap_paramList	196
6.27.4.2	daynightboth	196
6.27.4.3	end_date	196
6.27.4.4	grid	196
6.27.4.5	grid_fill	196
6.27.4.6	modis_id	196
6.27.4.7	modis_identifier	196
6.27.4.8	modis_platform	197
6.27.4.9	start_date	197
6.27.4.10	use_long_name	197

6.27.4.11	<code>variable_list</code>	197
6.27.4.12	<code>verbose</code>	197
6.28	<code>skdaccess.geo.modis.stream.cloud_opacity.DataFetcher</code> Class Reference	197
6.28.1	Detailed Description	198
6.28.2	Constructor & Destructor Documentation	198
6.28.2.1	<code>__init__()</code>	198
6.29	<code>skdaccess.geo.modis.stream.cloud_mask.DataFetcher</code> Class Reference	198
6.29.1	Detailed Description	199
6.29.2	Constructor & Destructor Documentation	199
6.29.2.1	<code>__init__()</code>	199
6.30	<code>skdaccess.planetary.ode.cache.DataFetcher</code> Class Reference	200
6.30.1	Detailed Description	201
6.30.2	Constructor & Destructor Documentation	201
6.30.2.1	<code>__init__()</code>	202
6.30.3	Member Function Documentation	202
6.30.3.1	<code>__str__()</code>	202
6.30.3.2	<code>cacheData()</code>	202
6.30.3.3	<code>checkIfDataExists()</code>	203
6.30.3.4	<code>getConfig()</code>	203
6.30.3.5	<code>getConfigItem()</code>	204
6.30.3.6	<code>getDataLocation()</code>	204
6.30.3.7	<code>getHDFStorage()</code>	204
6.30.3.8	<code>getMetadata()</code>	205
6.30.3.9	<code>multirun_enabled()</code>	205
6.30.3.10	<code>output()</code>	205
6.30.3.11	<code>perturb()</code>	206
6.30.3.12	<code>reset()</code>	206
6.30.3.13	<code>setDataLocation()</code>	206

6.30.3.14	<code>verbose_print()</code>	206
6.30.3.15	<code>writeConfig()</code>	207
6.30.3.16	<code>writeConfigItem()</code>	207
6.30.4	Member Data Documentation	207
6.30.4.1	<code>ap_paramList</code>	208
6.30.4.2	<code>eastern_lon</code>	208
6.30.4.3	<code>file_name</code>	208
6.30.4.4	<code>instrument</code>	208
6.30.4.5	<code>max_lat</code>	208
6.30.4.6	<code>max_ob_time</code>	208
6.30.4.7	<code>min_lat</code>	208
6.30.4.8	<code>min_ob_time</code>	209
6.30.4.9	<code>mission</code>	209
6.30.4.10	<code>number_product_limit</code>	209
6.30.4.11	<code>product_id</code>	209
6.30.4.12	<code>product_type</code>	209
6.30.4.13	<code>remove_ndv</code>	209
6.30.4.14	<code>result_offset_number</code>	209
6.30.4.15	<code>target</code>	210
6.30.4.16	<code>verbose</code>	210
6.30.4.17	<code>western_lon</code>	210
6.31	<code>skdaccess.geo.modis.stream.DataFetcher</code> Class Reference	210
6.31.1	Detailed Description	211
6.31.2	Constructor & Destructor Documentation	212
6.31.2.1	<code>__init__()</code>	212
6.31.3	Member Function Documentation	212
6.31.3.1	<code>__str__()</code>	212
6.31.3.2	<code>getConfig()</code>	213

6.31.3.3	getConfigItem()	213
6.31.3.4	getMetadata()	213
6.31.3.5	multirun_enabled()	214
6.31.3.6	output()	214
6.31.3.7	perturb()	214
6.31.3.8	reset()	214
6.31.3.9	retrieveOnlineData()	214
6.31.3.10	verbose_print()	215
6.31.3.11	writeConfig()	215
6.31.3.12	writeConfigItem()	215
6.31.4	Member Data Documentation	216
6.31.4.1	ap_paramList	216
6.31.4.2	daynightboth	216
6.31.4.3	end_date	216
6.31.4.4	grid	216
6.31.4.5	grid_fill	217
6.31.4.6	modis_id	217
6.31.4.7	modis_identifier	217
6.31.4.8	modis_platform	217
6.31.4.9	start_date	217
6.31.4.10	use_long_name	217
6.31.4.11	variable_list	217
6.31.4.12	verbose	218
6.32	skdaccess.geo.grace.mascon.cache.DataFetcher Class Reference	218
6.32.1	Detailed Description	219
6.32.2	Constructor & Destructor Documentation	219
6.32.2.1	__init__()	219
6.32.3	Member Function Documentation	220

6.32.3.1	__str__()	220
6.32.3.2	cacheData()	220
6.32.3.3	checkIfDataExists()	221
6.32.3.4	getConfig()	221
6.32.3.5	getConfigItem()	221
6.32.3.6	getDataLocation()	222
6.32.3.7	getHDFSStorage()	222
6.32.3.8	getMasconPlacement()	223
6.32.3.9	getMetadata()	223
6.32.3.10	multirun_enabled()	223
6.32.3.11	output()	223
6.32.3.12	perturb()	224
6.32.3.13	reset()	224
6.32.3.14	setDataLocation()	224
6.32.3.15	verbose_print()	224
6.32.3.16	writeConfig()	225
6.32.3.17	writeConfigItem()	225
6.32.4	Member Data Documentation	225
6.32.4.1	ap_paramList	226
6.32.4.2	end_date	226
6.32.4.3	mascon_placement_url	226
6.32.4.4	mascon_url	226
6.32.4.5	scale_factor_url	226
6.32.4.6	start_date	226
6.32.4.7	verbose	226
6.33	skdaccess.geo.mahali.tec.DataFetcher Class Reference	227
6.33.1	Detailed Description	228
6.33.2	Constructor & Destructor Documentation	228

6.33.2.1	<code>__init__()</code>	228
6.33.3	Member Function Documentation	229
6.33.3.1	<code>__str__()</code>	229
6.33.3.2	<code>cacheData()</code>	229
6.33.3.3	<code>checkIfDataExists()</code>	230
6.33.3.4	<code>getConfig()</code>	230
6.33.3.5	<code>getConfigItem()</code>	230
6.33.3.6	<code>getDataLocation()</code>	231
6.33.3.7	<code>getHDFSStorage()</code>	231
6.33.3.8	<code>getMetadata()</code>	231
6.33.3.9	<code>multirun_enabled()</code>	232
6.33.3.10	<code>output()</code>	232
6.33.3.11	<code>perturb()</code>	232
6.33.3.12	<code>reset()</code>	232
6.33.3.13	<code>setDataLocation()</code>	232
6.33.3.14	<code>verbose_print()</code>	233
6.33.3.15	<code>writeConfig()</code>	233
6.33.3.16	<code>writeConfigItem()</code>	233
6.33.4	Member Data Documentation	234
6.33.4.1	<code>ap_paramList</code>	234
6.33.4.2	<code>date_range</code>	234
6.33.4.3	<code>end_date</code>	234
6.33.4.4	<code>start_date</code>	234
6.33.4.5	<code>verbose</code>	234
6.34	<code>skdaccess.geo.pbo.DataFetcher</code> Class Reference	235
6.34.1	Detailed Description	236
6.34.2	Constructor & Destructor Documentation	236
6.34.2.1	<code>__init__()</code>	237

6.34.3	Member Function Documentation	238
6.34.3.1	__str__()	238
6.34.3.2	downloadFullDataset()	238
6.34.3.3	getAntennaLogs()	239
6.34.3.4	getConfig()	239
6.34.3.5	getConfigItem()	239
6.34.3.6	getDataLocation()	240
6.34.3.7	getInfo()	240
6.34.3.8	getMetadata()	240
6.34.3.9	getStationMetadata()	241
6.34.3.10	multirun_enabled()	241
6.34.3.11	output()	241
6.34.3.12	perturb()	241
6.34.3.13	reset()	242
6.34.3.14	setDataLocation()	242
6.34.3.15	setStationList()	242
6.34.3.16	verbose_print()	242
6.34.3.17	writeConfig()	243
6.34.3.18	writeConfigItem()	243
6.34.4	Member Data Documentation	243
6.34.4.1	antenna_info	244
6.34.4.2	ap_paramList	244
6.34.4.3	default_columns	244
6.34.4.4	default_error_columns	244
6.34.4.5	index_date_only	244
6.34.4.6	meta_data	244
6.34.4.7	station_list	244
6.34.4.8	use_progress_bar	245

6.34.4.9	verbose	245
6.35	skdaccess.geo.imsdnhs.DataFetcher Class Reference	245
6.35.1	Detailed Description	246
6.35.2	Constructor & Destructor Documentation	246
6.35.2.1	__init__()	246
6.35.3	Member Function Documentation	247
6.35.3.1	__str__()	247
6.35.3.2	downloadFullDataset()	247
6.35.3.3	getConfig()	247
6.35.3.4	getConfigItem()	248
6.35.3.5	getDataLocation()	248
6.35.3.6	getMetadata()	248
6.35.3.7	multirun_enabled()	249
6.35.3.8	output()	249
6.35.3.9	perturb()	249
6.35.3.10	reset()	249
6.35.3.11	setDataLocation()	249
6.35.3.12	verbose_print()	250
6.35.3.13	writeConfig()	250
6.35.3.14	writeConfigItem()	250
6.35.4	Member Data Documentation	251
6.35.4.1	ap_paramList	251
6.35.4.2	coordinate_dict	251
6.35.4.3	end_date	251
6.35.4.4	start_date	251
6.35.4.5	verbose	251
6.36	skdaccess.geo.era_interim.cache.DataFetcher Class Reference	252
6.36.1	Detailed Description	253

6.36.2	Constructor & Destructor Documentation	253
6.36.2.1	__init__()	253
6.36.3	Member Function Documentation	254
6.36.3.1	__str__()	254
6.36.3.2	cacheData()	254
6.36.3.3	checkIfDataExists()	255
6.36.3.4	getConfig()	255
6.36.3.5	getConfigItem()	255
6.36.3.6	getDataLocation()	256
6.36.3.7	getHDFSStorage()	256
6.36.3.8	getMetadata()	256
6.36.3.9	multirun_enabled()	257
6.36.3.10	output()	257
6.36.3.11	perturb()	257
6.36.3.12	reset()	257
6.36.3.13	setDataLocation()	257
6.36.3.14	verbose_print()	258
6.36.3.15	writeConfig()	258
6.36.3.16	writeConfigItem()	258
6.36.4	Member Data Documentation	259
6.36.4.1	ap_paramList	259
6.36.4.2	data_names	259
6.36.4.3	date_list	259
6.36.4.4	password	259
6.36.4.5	username	259
6.36.4.6	verbose	260
6.37	skdaccess.geo.ngl_gps.DataFetcher Class Reference	260
6.37.1	Detailed Description	261

6.37.2	Constructor & Destructor Documentation	261
6.37.2.1	__init__()	261
6.37.3	Member Function Documentation	262
6.37.3.1	__str__()	262
6.37.3.2	downloadFullDataset()	262
6.37.3.3	getAntennaLogs()	263
6.37.3.4	getConfig()	263
6.37.3.5	getConfigItem()	263
6.37.3.6	getDataLocation()	264
6.37.3.7	getMetadata()	264
6.37.3.8	getStationMetadata()	264
6.37.3.9	multirun_enabled()	265
6.37.3.10	output()	265
6.37.3.11	perturb()	265
6.37.3.12	reset()	265
6.37.3.13	setDataLocation()	265
6.37.3.14	verbose_print()	266
6.37.3.15	writeConfig()	266
6.37.3.16	writeConfigItem()	266
6.37.4	Member Data Documentation	267
6.37.4.1	ap_paramList	267
6.37.4.2	data_type	267
6.37.4.3	end_date	267
6.37.4.4	lat_range	267
6.37.4.5	lon_range	267
6.37.4.6	mdyratio	268
6.37.4.7	start_date	268
6.37.4.8	verbose	268

6.38	skdaccess.astro.spectra.stream.DataFetcher Class Reference	268
6.38.1	Detailed Description	269
6.38.2	Constructor & Destructor Documentation	269
6.38.2.1	__init__()	269
6.38.3	Member Function Documentation	270
6.38.3.1	__str__()	270
6.38.3.2	getConfig()	270
6.38.3.3	getConfigItem()	270
6.38.3.4	getMetadata()	271
6.38.3.5	multirun_enabled()	271
6.38.3.6	output()	271
6.38.3.7	perturb()	272
6.38.3.8	reset()	272
6.38.3.9	retrieveOnlineData()	272
6.38.3.10	verbose_print()	272
6.38.3.11	writeConfig()	273
6.38.3.12	writeConfigItem()	273
6.38.4	Member Data Documentation	273
6.38.4.1	ap_paramList	274
6.38.4.2	verbose	274
6.39	skdaccess.geo.grace.DataFetcher Class Reference	274
6.39.1	Detailed Description	275
6.39.2	Constructor & Destructor Documentation	275
6.39.2.1	__init__()	275
6.39.3	Member Function Documentation	276
6.39.3.1	__str__()	276
6.39.3.2	downloadFullDataset()	276
6.39.3.3	getConfig()	277

6.39.3.4	getConfigItem()	277
6.39.3.5	getDataLocation()	277
6.39.3.6	getMetadata()	278
6.39.3.7	multirun_enabled()	278
6.39.3.8	output()	278
6.39.3.9	perturb()	278
6.39.3.10	reset()	279
6.39.3.11	setDataLocation()	279
6.39.3.12	verbose_print()	279
6.39.3.13	writeConfig()	279
6.39.3.14	writeConfigItem()	280
6.39.4	Member Data Documentation	280
6.39.4.1	ap_paramList	280
6.39.4.2	end_date	280
6.39.4.3	start_date	281
6.39.4.4	verbose	281
6.40	skdaccess.geo.mahali.rinex.DataFetcher Class Reference	281
6.40.1	Detailed Description	282
6.40.2	Constructor & Destructor Documentation	282
6.40.2.1	__init__()	283
6.40.3	Member Function Documentation	283
6.40.3.1	__str__()	283
6.40.3.2	cacheData() [1/2]	283
6.40.3.3	cacheData() [2/2]	284
6.40.3.4	checkIfDataExists()	284
6.40.3.5	getConfig()	285
6.40.3.6	getConfigItem()	285
6.40.3.7	getDataLocation()	285

6.40.3.8	getHDFSStorage()	286
6.40.3.9	getMetadata()	286
6.40.3.10	multirun_enabled()	287
6.40.3.11	output()	287
6.40.3.12	perturb()	287
6.40.3.13	reset()	287
6.40.3.14	setDataLocation()	287
6.40.3.15	verbose_print()	288
6.40.3.16	writeConfig()	288
6.40.3.17	writeConfigItem()	288
6.40.4	Member Data Documentation	289
6.40.4.1	ap_paramList	289
6.40.4.2	date_range	289
6.40.4.3	end_date	289
6.40.4.4	generate_links	289
6.40.4.5	start_date	289
6.40.4.6	verbose	290
6.41	skdaccess.finance.timeseries.stream.DataFetcher Class Reference	290
6.41.1	Detailed Description	291
6.41.2	Constructor & Destructor Documentation	291
6.41.2.1	__init__()	291
6.41.3	Member Function Documentation	291
6.41.3.1	__str__()	292
6.41.3.2	getConfig()	292
6.41.3.3	getConfigItem()	292
6.41.3.4	getMetadata()	292
6.41.3.5	multirun_enabled()	293
6.41.3.6	output()	293

6.41.3.7	<code>perturb()</code>	293
6.41.3.8	<code>reset()</code>	293
6.41.3.9	<code>retrieveOnlineData()</code>	293
6.41.3.10	<code>verbose_print()</code>	294
6.41.3.11	<code>writeConfig()</code>	294
6.41.3.12	<code>writeConfigItem()</code>	294
6.41.4	Member Data Documentation	295
6.41.4.1	<code>ap_paramList</code>	295
6.41.4.2	<code>data_type</code>	295
6.41.4.3	<code>end_date</code>	295
6.41.4.4	<code>interval</code>	295
6.41.4.5	<code>possible_data_types</code>	296
6.41.4.6	<code>possible_intervals</code>	296
6.41.4.7	<code>start_date</code>	296
6.41.4.8	<code>verbose</code>	296
6.42	<code>skdaccess.astro.voyager.DataFetcher</code> Class Reference	296
6.42.1	Detailed Description	298
6.42.2	Constructor & Destructor Documentation	298
6.42.2.1	<code>__init__()</code>	298
6.42.3	Member Function Documentation	298
6.42.3.1	<code>__str__()</code>	298
6.42.3.2	<code>cacheData()</code>	299
6.42.3.3	<code>checkIfDataExists()</code>	299
6.42.3.4	<code>generateURL()</code>	300
6.42.3.5	<code>getConfig()</code>	300
6.42.3.6	<code>getConfigItem()</code>	300
6.42.3.7	<code>getDataLocation()</code>	301
6.42.3.8	<code>getHDFStorage()</code>	301

6.42.3.9	getMetadata()	302
6.42.3.10	getMetadataFiles()	302
6.42.3.11	multirun_enabled()	302
6.42.3.12	output()	303
6.42.3.13	parseVoyagerData()	303
6.42.3.14	parseVoyagerMetadata()	303
6.42.3.15	perturb()	304
6.42.3.16	reset()	304
6.42.3.17	setDataLocation()	304
6.42.3.18	verbose_print()	304
6.42.3.19	writeConfig()	305
6.42.3.20	writeConfigItem()	305
6.42.4	Member Data Documentation	305
6.42.4.1	ap_paramList	306
6.42.4.2	base_url	306
6.42.4.3	field_names	306
6.42.4.4	field_widths	306
6.42.4.5	spacecraft_list	306
6.42.4.6	verbose	306
6.42.4.7	year_list	306
6.43	skdaccess.geo.mahali.temperature.DataFetcher Class Reference	307
6.43.1	Detailed Description	308
6.43.2	Constructor & Destructor Documentation	308
6.43.2.1	__init__()	308
6.43.3	Member Function Documentation	308
6.43.3.1	__str__()	308
6.43.3.2	getConfig()	309
6.43.3.3	getConfigItem()	309

6.43.3.4	getMetadata()	309
6.43.3.5	multirun_enabled()	310
6.43.3.6	output()	310
6.43.3.7	perturb()	310
6.43.3.8	reset()	310
6.43.3.9	retrieveOnlineData()	310
6.43.3.10	verbose_print()	311
6.43.3.11	writeConfig()	311
6.43.3.12	writeConfigItem()	311
6.43.4	Member Data Documentation	312
6.43.4.1	ap_paramList	312
6.43.4.2	end_date	312
6.43.4.3	start_date	312
6.43.4.4	verbose	312
6.44	skdaccess.solar.sdo.DataFetcher Class Reference	313
6.44.1	Detailed Description	314
6.44.2	Constructor & Destructor Documentation	314
6.44.2.1	__init__()	314
6.44.3	Member Function Documentation	314
6.44.3.1	__str__()	314
6.44.3.2	getConfig()	315
6.44.3.3	getConfigItem()	315
6.44.3.4	getMetadata()	315
6.44.3.5	multirun_enabled()	316
6.44.3.6	output()	316
6.44.3.7	perturb()	316
6.44.3.8	reset()	316
6.44.3.9	retrieveOnlineData()	316

6.44.3.10	verbose_print()	317
6.44.3.11	writeConfig()	317
6.44.3.12	writeConfigItem()	317
6.44.4	Member Data Documentation	318
6.44.4.1	ap_paramList	318
6.44.4.2	verbose	318
6.45	skdaccess.framework.data_class.DataFetcherBase Class Reference	318
6.45.1	Detailed Description	319
6.45.2	Constructor & Destructor Documentation	319
6.45.2.1	__init__()	319
6.45.3	Member Function Documentation	320
6.45.3.1	__str__()	320
6.45.3.2	getConfig()	320
6.45.3.3	getConfigItem()	320
6.45.3.4	getMetadata()	321
6.45.3.5	multirun_enabled()	321
6.45.3.6	output()	321
6.45.3.7	perturb()	322
6.45.3.8	reset()	322
6.45.3.9	verbose_print()	322
6.45.3.10	writeConfig()	322
6.45.3.11	writeConfigItem()	323
6.45.4	Member Data Documentation	323
6.45.4.1	ap_paramList	323
6.45.4.2	verbose	323
6.46	skdaccess.framework.data_class.DataFetcherCache Class Reference	324
6.46.1	Detailed Description	325
6.46.2	Member Function Documentation	325

6.46.2.1	<code>__str__()</code>	325
6.46.2.2	<code>cacheData()</code>	326
6.46.2.3	<code>checkIfDataExists()</code>	326
6.46.2.4	<code>getConfig()</code>	327
6.46.2.5	<code>getConfigItem()</code>	327
6.46.2.6	<code>getDataLocation()</code>	327
6.46.2.7	<code>getHDFSStorage()</code>	328
6.46.2.8	<code>getMetadata()</code>	328
6.46.2.9	<code>multirun_enabled()</code>	329
6.46.2.10	<code>output()</code>	329
6.46.2.11	<code>perturb()</code>	329
6.46.2.12	<code>reset()</code>	329
6.46.2.13	<code>setDataLocation()</code>	329
6.46.2.14	<code>verbose_print()</code>	330
6.46.2.15	<code>writeConfig()</code>	330
6.46.2.16	<code>writeConfigItem()</code>	330
6.46.3	Member Data Documentation	331
6.46.3.1	<code>ap_paramList</code>	331
6.46.3.2	<code>verbose</code>	331
6.47	<code>skdaccess.framework.data_class.DataFetcherLocal</code> Class Reference	331
6.47.1	Detailed Description	332
6.47.2	Member Function Documentation	332
6.47.2.1	<code>__str__()</code>	333
6.47.2.2	<code>getConfig()</code>	333
6.47.2.3	<code>getConfigItem()</code>	333
6.47.2.4	<code>getDataLocation()</code>	333
6.47.2.5	<code>getMetadata()</code>	334
6.47.2.6	<code>multirun_enabled()</code>	334

6.47.2.7	output()	334
6.47.2.8	perturb()	335
6.47.2.9	reset()	335
6.47.2.10	setDataLocation()	335
6.47.2.11	verbose_print()	335
6.47.2.12	writeConfig()	336
6.47.2.13	writeConfigItem()	336
6.47.3	Member Data Documentation	336
6.47.3.1	ap_paramList	337
6.47.3.2	verbose	337
6.48	skdaccess.framework.data_class.DataFetcherStorage Class Reference	337
6.48.1	Detailed Description	338
6.48.2	Member Function Documentation	338
6.48.2.1	__str__()	338
6.48.2.2	downloadFullDataset()	338
6.48.2.3	getConfig()	339
6.48.2.4	getConfigItem()	339
6.48.2.5	getDataLocation()	340
6.48.2.6	getMetadata()	340
6.48.2.7	multirun_enabled()	340
6.48.2.8	output()	341
6.48.2.9	perturb()	341
6.48.2.10	reset()	341
6.48.2.11	setDataLocation()	341
6.48.2.12	verbose_print()	342
6.48.2.13	writeConfig()	342
6.48.2.14	writeConfigItem()	342
6.48.3	Member Data Documentation	343

6.48.3.1	ap_paramList	343
6.48.3.2	verbose	343
6.49	skdaccess.framework.data_class.DataFetcherStream Class Reference	343
6.49.1	Detailed Description	344
6.49.2	Member Function Documentation	344
6.49.2.1	__str__()	345
6.49.2.2	getConfig()	345
6.49.2.3	getConfigItem()	345
6.49.2.4	getMetadata()	345
6.49.2.5	multirun_enabled()	346
6.49.2.6	output()	346
6.49.2.7	perturb()	346
6.49.2.8	reset()	346
6.49.2.9	retrieveOnlineData()	346
6.49.2.10	verbose_print()	347
6.49.2.11	writeConfig()	347
6.49.2.12	writeConfigItem()	347
6.49.3	Member Data Documentation	348
6.49.3.1	ap_paramList	348
6.49.3.2	verbose	348
6.50	skdaccess.geo.mahali.rinex.data_wrapper.DataWrapper Class Reference	348
6.50.1	Detailed Description	349
6.50.2	Member Function Documentation	349
6.50.2.1	__len__()	350
6.50.2.2	addResult()	350
6.50.2.3	get()	350
6.50.2.4	getIterator()	351
6.50.2.5	getResults()	351

6.50.2.6	getRunID()	351
6.50.2.7	info()	351
6.50.2.8	reset()	352
6.50.2.9	update()	352
6.50.2.10	updateMetadata()	352
6.50.3	Member Data Documentation	352
6.50.3.1	constants	352
6.50.3.2	data	353
6.50.3.3	meta_data	353
6.50.3.4	results	353
6.50.3.5	run_id	353
6.51	skdaccess.framework.data_class.DataWrapperBase Class Reference	353
6.51.1	Detailed Description	354
6.51.2	Constructor & Destructor Documentation	354
6.51.2.1	__init__()	354
6.51.3	Member Function Documentation	355
6.51.3.1	__len__()	355
6.51.3.2	addResult()	355
6.51.3.3	get()	355
6.51.3.4	getIterator()	356
6.51.3.5	getResults()	356
6.51.3.6	getRunID()	356
6.51.3.7	info()	357
6.51.3.8	reset()	357
6.51.3.9	update()	357
6.51.3.10	updateMetadata()	357
6.51.4	Member Data Documentation	358
6.51.4.1	constants	358

6.51.4.2	data	358
6.51.4.3	meta_data	358
6.51.4.4	results	358
6.51.4.5	run_id	358
6.52	skdaccess.utilities.file_browser.FileBrowser Class Reference	359
6.52.1	Constructor & Destructor Documentation	359
6.52.1.1	__init__()	359
6.52.2	Member Function Documentation	359
6.52.2.1	widget()	359
6.52.3	Member Data Documentation	360
6.52.3.1	dirs	360
6.52.3.2	files	360
6.52.3.3	path	360
6.53	skdaccess.framework.data_class.ImageWrapper Class Reference	360
6.53.1	Detailed Description	361
6.53.2	Member Function Documentation	361
6.53.2.1	__len__()	362
6.53.2.2	addResult()	362
6.53.2.3	deleteData()	362
6.53.2.4	get()	362
6.53.2.5	getIterator()	363
6.53.2.6	getResults()	363
6.53.2.7	getRunID()	363
6.53.2.8	info()	364
6.53.2.9	reset()	364
6.53.2.10	update()	364
6.53.2.11	updateData()	364
6.53.2.12	updateMetadata()	365

6.53.3	Member Data Documentation	365
6.53.3.1	constants	365
6.53.3.2	data	365
6.53.3.3	meta_data	365
6.53.3.4	results	366
6.53.3.5	run_id	366
6.54	skdaccess.utilities.modis_util.LatLon Class Reference	366
6.54.1	Detailed Description	367
6.54.2	Constructor & Destructor Documentation	367
6.54.2.1	__init__()	367
6.54.3	Member Function Documentation	367
6.54.3.1	__call__()	367
6.54.4	Member Data Documentation	368
6.54.4.1	alat	368
6.54.4.2	alon	368
6.54.4.3	lat_data	368
6.54.4.4	lon_data	368
6.54.4.5	x_offset	368
6.54.4.6	y_offset	368
6.55	skdaccess.utilities.image_util.LinearGeolocation Class Reference	369
6.55.1	Detailed Description	369
6.55.2	Constructor & Destructor Documentation	370
6.55.2.1	__init__()	370
6.55.3	Member Function Documentation	370
6.55.3.1	getExtents()	370
6.55.3.2	getLatLon()	370
6.55.3.3	getYX()	371
6.55.4	Member Data Documentation	371

6.55.4.1	flip_y	371
6.55.4.2	lat_extents	371
6.55.4.3	lat_pixel_size	372
6.55.4.4	len_x	372
6.55.4.5	len_y	372
6.55.4.6	lon_extents	372
6.55.4.7	lon_pixel_size	372
6.55.4.8	start_lat	372
6.55.4.9	start_lon	372
6.55.4.10	x_offset	373
6.55.4.11	y_offset	373
6.56	skdaccess.framework.data_class.SeriesDictionaryWrapper Class Reference	373
6.56.1	Detailed Description	374
6.56.2	Member Function Documentation	374
6.56.2.1	__len__()	374
6.56.2.2	addResult()	374
6.56.2.3	get()	375
6.56.2.4	getIndices()	375
6.56.2.5	getIterator()	375
6.56.2.6	getLength()	376
6.56.2.7	getResults()	376
6.56.2.8	getRunID()	376
6.56.2.9	info()	376
6.56.2.10	reset()	377
6.56.2.11	update()	377
6.56.2.12	updateMetadata()	377
6.56.3	Member Data Documentation	377
6.56.3.1	constants	377

6.56.3.2	data	378
6.56.3.3	data_names	378
6.56.3.4	error_names	378
6.56.3.5	meta_data	378
6.56.3.6	results	378
6.56.3.7	run_id	378
6.57	skdaccess.framework.data_class.SeriesWrapper Class Reference	379
6.57.1	Detailed Description	380
6.57.2	Constructor & Destructor Documentation	380
6.57.2.1	__init__()	380
6.57.3	Member Function Documentation	380
6.57.3.1	__len__()	381
6.57.3.2	addResult()	381
6.57.3.3	get()	381
6.57.3.4	getIndices()	382
6.57.3.5	getIterator()	382
6.57.3.6	getLength()	382
6.57.3.7	getResults()	382
6.57.3.8	getRunID()	383
6.57.3.9	info()	383
6.57.3.10	reset()	383
6.57.3.11	update()	383
6.57.3.12	updateMetadata()	384
6.57.4	Member Data Documentation	384
6.57.4.1	constants	384
6.57.4.2	data	384
6.57.4.3	data_names	384
6.57.4.4	error_names	385

6.57.4.5	meta_data	385
6.57.4.6	results	385
6.57.4.7	run_id	385
6.58	skdaccess.utilities.sounding_util.SoundingParser Class Reference	385
6.58.1	Detailed Description	386
6.58.2	Constructor & Destructor Documentation	386
6.58.2.1	__init__()	386
6.58.3	Member Function Documentation	386
6.58.3.1	handle_data()	386
6.58.3.2	handle_endtag()	387
6.58.3.3	handle_starttag()	387
6.58.4	Member Data Documentation	387
6.58.4.1	data_dict	387
6.58.4.2	in_header	388
6.58.4.3	in_pre_tag	388
6.58.4.4	label	388
6.58.4.5	metadata_dict	388
6.58.4.6	read_data	388
6.58.4.7	tmp	388
6.59	skdaccess.utilities.image_util.SplineLatLon Class Reference	389
6.59.1	Detailed Description	389
6.59.2	Constructor & Destructor Documentation	389
6.59.2.1	__init__()	390
6.59.3	Member Function Documentation	390
6.59.3.1	__call__()	390
6.59.4	Member Data Documentation	391
6.59.4.1	lat_func	391
6.59.4.2	lon_func	391

6.59.4.3	x_offset	391
6.59.4.4	y_offset	391
6.60	skdaccess.framework.data_class.TableWrapper Class Reference	392
6.60.1	Detailed Description	393
6.60.2	Constructor & Destructor Documentation	393
6.60.2.1	__init__()	393
6.60.3	Member Function Documentation	394
6.60.3.1	__len__()	394
6.60.3.2	addColumn()	394
6.60.3.3	addResult()	394
6.60.3.4	get()	395
6.60.3.5	getDefaultColumns()	395
6.60.3.6	getDefaultErrorColumns()	395
6.60.3.7	getIterator()	396
6.60.3.8	getLength()	396
6.60.3.9	getResults()	396
6.60.3.10	getRunID()	396
6.60.3.11	info()	397
6.60.3.12	removeFrames()	397
6.60.3.13	reset()	397
6.60.3.14	update()	397
6.60.3.15	updateData()	398
6.60.3.16	updateFrames()	398
6.60.3.17	updateMetadata()	398
6.60.4	Member Data Documentation	399
6.60.4.1	constants	399
6.60.4.2	data	399
6.60.4.3	default_columns	399

6.60.4.4	default_error_columns	399
6.60.4.5	meta_data	399
6.60.4.6	results	400
6.60.4.7	run_id	400
6.61	skdaccess.framework.data_class.XArrayWrapper Class Reference	400
6.61.1	Detailed Description	401
6.61.2	Constructor & Destructor Documentation	401
6.61.2.1	__init__()	401
6.61.3	Member Function Documentation	401
6.61.3.1	__len__()	401
6.61.3.2	addResult()	401
6.61.3.3	get()	402
6.61.3.4	getIterator()	402
6.61.3.5	getResults()	402
6.61.3.6	getRunID()	403
6.61.3.7	info()	403
6.61.3.8	reset()	403
6.61.3.9	update()	403
6.61.3.10	updateMetadata()	404
6.61.4	Member Data Documentation	404
6.61.4.1	constants	404
6.61.4.2	data	404
6.61.4.3	index_list	404
6.61.4.4	meta_data	405
6.61.4.5	results	405
6.61.4.6	run_id	405

7	File Documentation	407
7.1	finance/timeseries/stream.py File Reference	407
7.2	astro/spectra/stream.py File Reference	407
7.3	engineering/la/generic/stream.py File Reference	407
7.4	engineering/la/traffic_counts/stream.py File Reference	408
7.5	engineering/webcam/mit_sailing/stream.py File Reference	408
7.6	framework/data_class.py File Reference	408
7.7	framework/param_class.py File Reference	409
7.8	geo/mahali/rinex/data_wrapper.py File Reference	410
7.9	solar/sdo/data_fetcher.py File Reference	410
7.10	planetary/ode/cache/data_fetcher.py File Reference	410
7.11	geo/grace/mascon/cache/data_fetcher.py File Reference	410
7.12	geo/grace/data_fetcher.py File Reference	411
7.13	geo/mahali/tec/data_fetcher.py File Reference	411
7.14	geo/mahali/rinex/data_fetcher.py File Reference	411
7.15	geo/mahali/temperature/data_fetcher.py File Reference	412
7.16	geo/ngl_gps/data_fetcher.py File Reference	412
7.17	geo/era_interim/cache/data_fetcher.py File Reference	412
7.18	geo/imsdnhs/data_fetcher.py File Reference	412
7.19	geo/gldas/data_fetcher.py File Reference	413
7.20	geo/sentinel_1/cache/data_fetcher.py File Reference	413
7.21	geo/magnetometer/data_fetcher.py File Reference	413
7.22	geo/wyoming_sounding/cache/data_fetcher.py File Reference	414
7.23	geo/wyoming_sounding/stream/data_fetcher.py File Reference	414
7.24	geo/modis/cache/cloud_opacity/data_fetcher.py File Reference	414
7.25	geo/modis/cache/cloud_mask/data_fetcher.py File Reference	414
7.26	geo/modis/cache/reflectance/data_fetcher.py File Reference	415
7.27	geo/modis/cache/data_fetcher.py File Reference	415

7.28	geo/modis/stream/cloud_opacity/data_fetcher.py File Reference	415
7.29	geo/modis/stream/cloud_mask/data_fetcher.py File Reference	416
7.30	geo/modis/stream/reflectance/data_fetcher.py File Reference	416
7.31	geo/modis/stream/data_fetcher.py File Reference	416
7.32	geo/uavsar/cache/data_fetcher.py File Reference	416
7.33	geo/srtm/cache/data_fetcher.py File Reference	417
7.34	geo/groundwater/data_fetcher.py File Reference	417
7.35	geo/pbo/data_fetcher.py File Reference	417
7.36	astro/kepler/data_fetcher.py File Reference	418
7.37	astro/voyager/data_fetcher.py File Reference	418
7.38	utilities/file_browser.py File Reference	418
7.39	utilities/file_util.py File Reference	418
7.40	utilities/grace_util.py File Reference	419
7.41	utilities/gw_util.py File Reference	419
7.42	utilities/image_util.py File Reference	419
7.43	utilities/kepler_util.py File Reference	420
7.44	utilities/mahali_util.py File Reference	420
7.45	utilities/modis_util.py File Reference	421
7.46	utilities/ode_util.py File Reference	421
7.47	utilities/pbo_util.py File Reference	422
7.48	utilities/sentinel_1_util.py File Reference	423
7.49	utilities/sounding_util.py File Reference	423
7.50	utilities/srtm_util.py File Reference	423
7.51	utilities/support.py File Reference	424
7.52	utilities/uavsar_util.py File Reference	424

Chapter 1

Namespace Index

1.1 Packages

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

skdaccess	13
skdaccess.astro	13
skdaccess.astro.kepler	13
skdaccess.astro.kepler.data_fetcher	14
skdaccess.astro.spectra	14
skdaccess.astro.spectra.stream	14
skdaccess.astro.voyager	14
skdaccess.astro.voyager.data_fetcher	14
skdaccess.engineering	14
skdaccess.engineering.la	15
skdaccess.engineering.la.generic	15
skdaccess.engineering.la.generic.stream	15
skdaccess.engineering.la.traffic_counts	15
skdaccess.engineering.la.traffic_counts.stream	15
skdaccess.engineering.webcam	15
skdaccess.engineering.webcam.mit_sailing	16
skdaccess.engineering.webcam.mit_sailing.stream	16
skdaccess.finance	16
skdaccess.finance.timeseries	16
skdaccess.finance.timeseries.stream	16
skdaccess.framework	16
skdaccess.framework.data_class	17
skdaccess.framework.param_class	17
skdaccess.geo	18
skdaccess.geo.era_interim	18
skdaccess.geo.era_interim.cache	18
skdaccess.geo.era_interim.cache.data_fetcher	18
skdaccess.geo.gldas	18
skdaccess.geo.gldas.data_fetcher	19
skdaccess.geo.grace	19
skdaccess.geo.grace.data_fetcher	19

skdaccess.geo.grace.mascon	19
skdaccess.geo.grace.mascon.cache	19
skdaccess.geo.grace.mascon.cache.data_fetcher	19
skdaccess.geo.groundwater	20
skdaccess.geo.groundwater.data_fetcher	20
skdaccess.geo.imsdnhs	20
skdaccess.geo.imsdnhs.data_fetcher	20
skdaccess.geo.magnetometer	20
skdaccess.geo.magnetometer.data_fetcher	20
skdaccess.geo.mahali	21
skdaccess.geo.mahali.rinex	21
skdaccess.geo.mahali.rinex.data_fetcher	21
skdaccess.geo.mahali.rinex.data_wrapper	21
skdaccess.geo.mahali.tec	21
skdaccess.geo.mahali.tec.data_fetcher	21
skdaccess.geo.mahali.temperature	22
skdaccess.geo.mahali.temperature.data_fetcher	22
skdaccess.geo.modis	22
skdaccess.geo.modis.cache	22
skdaccess.geo.modis.cache.cloud_mask	22
skdaccess.geo.modis.cache.cloud_mask.data_fetcher	22
skdaccess.geo.modis.cache.cloud_opacity	23
skdaccess.geo.modis.cache.cloud_opacity.data_fetcher	23
skdaccess.geo.modis.cache.data_fetcher	23
skdaccess.geo.modis.cache.reflectance	23
skdaccess.geo.modis.cache.reflectance.data_fetcher	23
skdaccess.geo.modis.stream	23
skdaccess.geo.modis.stream.cloud_mask	24
skdaccess.geo.modis.stream.cloud_mask.data_fetcher	24
skdaccess.geo.modis.stream.cloud_opacity	24
skdaccess.geo.modis.stream.cloud_opacity.data_fetcher	24
skdaccess.geo.modis.stream.data_fetcher	24
skdaccess.geo.modis.stream.reflectance	24
skdaccess.geo.modis.stream.reflectance.data_fetcher	25
skdaccess.geo.ngl_gps	25
skdaccess.geo.ngl_gps.data_fetcher	25
skdaccess.geo.pbo	25
skdaccess.geo.pbo.data_fetcher	25
skdaccess.geo.sentinel_1	25
skdaccess.geo.sentinel_1.cache	26
skdaccess.geo.sentinel_1.cache.data_fetcher	26
skdaccess.geo.srtm	26
skdaccess.geo.srtm.cache	26
skdaccess.geo.srtm.cache.data_fetcher	26
skdaccess.geo.uavsar	26
skdaccess.geo.uavsar.cache	27
skdaccess.geo.uavsar.cache.data_fetcher	27
skdaccess.geo.wyoming_sounding	27
skdaccess.geo.wyoming_sounding.cache	27
skdaccess.geo.wyoming_sounding.cache.data_fetcher	27
skdaccess.geo.wyoming_sounding.stream	27
skdaccess.geo.wyoming_sounding.stream.data_fetcher	28
skdaccess.planetary	28
skdaccess.planetary.ode	28

skdaccess.planetary.ode.cache	28
skdaccess.planetary.ode.cache.data_fetcher	28
skdaccess.solar	28
skdaccess.solar.sdo	29
skdaccess.solar.sdo.data_fetcher	29
skdaccess.utilities	29
skdaccess.utilities.file_browser	29
skdaccess.utilities.file_util	30
skdaccess.utilities.grace_util	30
skdaccess.utilities.gw_util	33
skdaccess.utilities.image_util	34
skdaccess.utilities.kepler_util	37
skdaccess.utilities.mahali_util	37
skdaccess.utilities.modis_util	38
skdaccess.utilities.ode_util	43
skdaccess.utilities.pbo_util	47
skdaccess.utilities.sentinel_1_util	51
skdaccess.utilities.sounding_util	52
skdaccess.utilities.srtm_util	53
skdaccess.utilities.support	54
skdaccess.utilities.uavsar_util	56

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

skdaccess.framework.param_class.AutoParam	82
skdaccess.framework.param_class.AutoParamList	85
skdaccess.framework.param_class.AutoParamListCycle	88
skdaccess.framework.param_class.AutoParamMinMax	91
GenericDataFetcher	
skdaccess.engineering.la.traffic_counts.stream.DataFetcher	142
MDF	
skdaccess.geo.modis.cache.cloud_mask.DataFetcher	184
skdaccess.geo.modis.cache.cloud_opacity.DataFetcher	176
skdaccess.geo.modis.cache.reflectance.DataFetcher	185
skdaccess.geo.modis.stream.cloud_mask.DataFetcher	198
skdaccess.geo.modis.stream.cloud_opacity.DataFetcher	197
skdaccess.geo.modis.stream.reflectance.DataFetcher	177
object	
skdaccess.framework.data_class.DataFetcherBase	318
skdaccess.framework.data_class.DataFetcherLocal	331
skdaccess.framework.data_class.DataFetcherCache	324
skdaccess.astro.kepler.DataFetcher	94
skdaccess.astro.voyager.DataFetcher	296
skdaccess.geo.era_interim.cache.DataFetcher	252
skdaccess.geo.grace.mascon.cache.DataFetcher	218
skdaccess.geo.mahali.rinex.DataFetcher	281
skdaccess.geo.mahali.tec.DataFetcher	227
skdaccess.geo.modis.cache.DataFetcher	187
skdaccess.geo.sentinel_1.cache.DataFetcher	117
skdaccess.geo.srtm.cache.DataFetcher	158
skdaccess.geo.uavsar.cache.DataFetcher	167
skdaccess.geo.wyoming_sounding.cache.DataFetcher	133
skdaccess.planetary.ode.cache.DataFetcher	200
skdaccess.framework.data_class.DataFetcherStorage	337

skdaccess.geo.gldas.DataFetcher	110
skdaccess.geo.grace.DataFetcher	274
skdaccess.geo.groundwater.DataFetcher	150
skdaccess.geo.imsdnhs.DataFetcher	245
skdaccess.geo.ngl_gps.DataFetcher	260
skdaccess.geo.pbo.DataFetcher	235
skdaccess.framework.data_class.DataFetcherStream	343
skdaccess.astro.spectra.stream.DataFetcher	268
skdaccess.engineering.la.generic.stream.DataFetcher	103
skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher	178
skdaccess.finance.timeseries.stream.DataFetcher	290
skdaccess.geo.magnetometer.DataFetcher	126
skdaccess.geo.mahali.temperature.DataFetcher	307
skdaccess.geo.modis.stream.DataFetcher	210
skdaccess.geo.wyoming_sounding.stream.DataFetcher	143
skdaccess.solar.sdo.DataFetcher	313
skdaccess.framework.data_class.DataWrapperBase	353
skdaccess.framework.data_class.ImageWrapper	360
skdaccess.framework.data_class.SeriesWrapper	379
skdaccess.framework.data_class.SeriesDictionaryWrapper	373
skdaccess.framework.data_class.TableWrapper	392
skdaccess.framework.data_class.XArrayWrapper	400
skdaccess.geo.mahali.rinex.data_wrapper.DataWrapper	348
skdaccess.framework.param_class.AutoList	61
skdaccess.framework.param_class.AutoListCycle	65
skdaccess.framework.param_class.AutoListPermute	70
skdaccess.framework.param_class.AutoListRemove	74
skdaccess.framework.param_class.AutoListSubset	78
skdaccess.utilities.file_browser.FileBrowser	359
skdaccess.utilities.image_util.AffineGlobalCoords	59
skdaccess.utilities.image_util.LinearGeolocation	369
skdaccess.utilities.image_util.SplineLatLon	389
skdaccess.utilities.modis_util.LatLon	366
HTMLParser	
skdaccess.utilities.sounding_util.SoundingParser	385

Chapter 3

Class Index

3.1 Class List

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

skdaccess.utilities.image_util.AffineGlobalCoords	
Convert between projected and pixel coordinates using an affine transformation	59
skdaccess.framework.param_class.AutoList	
Specifies a list for returning selections of lists, as opposed to a single element	61
skdaccess.framework.param_class.AutoListCycle	
An Autolist that cycles through different lists	65
skdaccess.framework.param_class.AutoListPermute	
A perturber that permutes a list	70
skdaccess.framework.param_class.AutoListRemove	
Removes a different single element from the initial list at each perturb call	74
skdaccess.framework.param_class.AutoListSubset	
An AutoList perturber that creates random subsets of a list	78
skdaccess.framework.param_class.AutoParam	
Defines a tunable parameter class inherited by specific subclasses	82
skdaccess.framework.param_class.AutoParamList	
A tunable parameter with a specified list of choices that can be randomly selected via perturb	85
skdaccess.framework.param_class.AutoParamListCycle	
Cycles through a list of paramters	88
skdaccess.framework.param_class.AutoParamMinMax	
A tunable parameter with min and max ranges, perturbs to a random value in range	91
skdaccess.astro.kepler.DataFetcher	
Data Fetcher for Kepler light curve data	94
skdaccess.engineering.la.generic.stream.DataFetcher	
Class for handling data requests to data.lacity.org	103
skdaccess.geo.gldas.DataFetcher	
Data Fetcher for GLDAS data	110
skdaccess.geo.sentinel_1.cache.DataFetcher	
DataFetcher for retrieving Sentinel SLC data	117
skdaccess.geo.magnetometer.DataFetcher	
Data fetcher for USGS geomagnetic observatories	126
skdaccess.geo.wyoming_sounding.cache.DataFetcher	
DataFetcher for retrieving Wyoming Sounding data	133

skdaccess.engineering.la.traffic_counts.stream.DataFetcher	
DataFetcher for retrieving traffic counts from LA	142
skdaccess.geo.wyoming_sounding.stream.DataFetcher	
DataFetcher for retrieving Wyoming Sounding data	143
skdaccess.geo.groundwater.DataFetcher	
Generates Data Wrappers of groundwater measurements taken in the US	150
skdaccess.geo.srtm.cache.DataFetcher	
DataFetcher for retrieving data from the Shuttle Radar Topography Mission	158
skdaccess.geo.uavsar.cache.DataFetcher	
Data Fetcher for UAVSAR data	167
skdaccess.geo.modis.cache.cloud_opacity.DataFetcher	
Data Fetcher for MODIS Cloud Opacity	176
skdaccess.geo.modis.stream.reflectance.DataFetcher	
Data fetcher for the modis surface reflectance product ('09', 1 km resolution)	177
skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher	
Data Fetcher for retrieving webcam images from the MIT Sailing Pavilion	178
skdaccess.geo.modis.cache.cloud_mask.DataFetcher	
Data Fetcher for MODIS Cloud Mask	184
skdaccess.geo.modis.cache.reflectance.DataFetcher	
Data fetcher for the modis surface reflectance product ('09', 1 km resolution)	185
skdaccess.geo.modis.cache.DataFetcher	
Data Fetcher for MODIS data	187
skdaccess.geo.modis.stream.cloud_opacity.DataFetcher	
Data Fetcher for MODIS Cloud Opacity	197
skdaccess.geo.modis.stream.cloud_mask.DataFetcher	
Data Fetcher for MODIS Cloud Mask	198
skdaccess.planetary.ode.cache.DataFetcher	
Data Fetcher from the Orbital Data Explorer (ODE)	200
skdaccess.geo.modis.stream.DataFetcher	
Data Fetcher for MODIS data	210
skdaccess.geo.grace.mascon.cache.DataFetcher	
Data Fetcher for GRACE mascon data	218
skdaccess.geo.mahali.tec.DataFetcher	
Data Fetcher for Mahali Data	227
skdaccess.geo.pbo.DataFetcher	
Data fetcher for PBO GPS data	235
skdaccess.geo.imsdnhs.DataFetcher	
Fetches data for the Interactive Multisensor Snow and Ice Mapping System Daily Northern Hemisphere Snow and Ice Analysis	245
skdaccess.geo.era_interim.cache.DataFetcher	
DataFetcher for retrieving ERA-I data	252
skdaccess.geo.ngl_gps.DataFetcher	
Data fetcher for GPS data from Nevada Geodetic Laboratory	260
skdaccess.astro.spectra.stream.DataFetcher	
Data Fetcher for Sloan Digital Sky Survey spectra	268
skdaccess.geo.grace.DataFetcher	
Data Fetcher for GRACE data	274
skdaccess.geo.mahali.rinex.DataFetcher	
Data Fetcher for Mahali Data	281
skdaccess.finance.timeseries.stream.DataFetcher	
Data Fetcher for retrieving stock data	290
skdaccess.astro.voyager.DataFetcher	
Data Fetcher for Mahali temperature data	296

skdaccess.geo.mahali.temperature.DataFetcher	
Data Fetcher for Mahali temperature data	307
skdaccess.solar.sdo.DataFetcher	
Data Fetcher for the Solar Dynamics Observatory	313
skdaccess.framework.data_class.DataFetcherBase	
Base class for all data fetchers	318
skdaccess.framework.data_class.DataFetcherCache	
Data fetcher base class for downloading data and caching results on hard disk	324
skdaccess.framework.data_class.DataFetcherLocal	
Data fetcher base class for use when storing data locally	331
skdaccess.framework.data_class.DataFetcherStorage	
Data fetcher base class for use when entire data set is downloaded	337
skdaccess.framework.data_class.DataFetcherStream	
Data fetcher base class for downloading data into memory	343
skdaccess.geo.mahali.rinex.data_wrapper.DataWrapper	
Data wrapper for Mahali data	348
skdaccess.framework.data_class.DataWrapperBase	
Base class for wrapping data for use in DiscoveryPipeline	353
skdaccess.utilities.file_browser.FileBrowser	
	359
skdaccess.framework.data_class.ImageWrapper	
Wrapper for image data	360
skdaccess.utilities.modis_util.LatLon	
Calculates Lat/Lon position from y,x pixel coordinate	366
skdaccess.utilities.image_util.LinearGeolocation	
This class provides functions to convert between pixel and geodetic coordinates	369
skdaccess.framework.data_class.SeriesDictionaryWrapper	
Data wrapper for series data using a dictionary of data frames	373
skdaccess.framework.data_class.SeriesWrapper	
Data wrapper for series data using a data panel	379
skdaccess.utilities.sounding_util.SoundingParser	
This class parses Wyoming Sounding data	385
skdaccess.utilities.image_util.SplineLatLon	
Holds a 2d spline for interpolating lat/lon grid	389
skdaccess.framework.data_class.TableWrapper	
Data wrapper for table data using an ordered dictionary	392
skdaccess.framework.data_class.XArrayWrapper	
Wrapper for xarrays	400

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

astro/kepler/ data_fetcher.py	418
astro/spectra/ stream.py	407
astro/voyager/ data_fetcher.py	418
engineering/la/generic/ stream.py	407
engineering/la/traffic_counts/ stream.py	408
engineering/webcam/mit_sailing/ stream.py	408
finance/timeseries/ stream.py	407
framework/ data_class.py	408
framework/ param_class.py	409
geo/era_interim/cache/ data_fetcher.py	412
geo/gldas/ data_fetcher.py	413
geo/grace/ data_fetcher.py	411
geo/grace/mascon/cache/ data_fetcher.py	410
geo/groundwater/ data_fetcher.py	417
geo/imsdnhs/ data_fetcher.py	412
geo/magnetometer/ data_fetcher.py	413
geo/mahali/rinex/ data_fetcher.py	411
geo/mahali/rinex/ data_wrapper.py	410
geo/mahali/tec/ data_fetcher.py	411
geo/mahali/temperature/ data_fetcher.py	412
geo/modis/cache/ data_fetcher.py	415
geo/modis/cache/cloud_mask/ data_fetcher.py	414
geo/modis/cache/cloud_opacity/ data_fetcher.py	414
geo/modis/cache/reflectance/ data_fetcher.py	415
geo/modis/stream/ data_fetcher.py	416
geo/modis/stream/cloud_mask/ data_fetcher.py	416
geo/modis/stream/cloud_opacity/ data_fetcher.py	415
geo/modis/stream/reflectance/ data_fetcher.py	416
geo/ngl_gps/ data_fetcher.py	412
geo/pbo/ data_fetcher.py	417
geo/sentinel_1/cache/ data_fetcher.py	413

geo/srtm/cache/data_fetcher.py	417
geo/uavsar/cache/data_fetcher.py	416
geo/wyoming_sounding/cache/data_fetcher.py	414
geo/wyoming_sounding/stream/data_fetcher.py	414
planetary/ode/cache/data_fetcher.py	410
solar/sdo/data_fetcher.py	410
utilities/file_browser.py	418
utilities/file_util.py	418
utilities/grace_util.py	419
utilities/gw_util.py	419
utilities/image_util.py	419
utilities/kepler_util.py	420
utilities/mahali_util.py	420
utilities/modis_util.py	421
utilities/ode_util.py	421
utilities/pbo_util.py	422
utilities/sentinel_1_util.py	423
utilities/sounding_util.py	423
utilities/srtm_util.py	423
utilities/support.py	424
utilities/uavsar_util.py	424

Chapter 5

Namespace Documentation

5.1 skdaccess Namespace Reference

Namespaces

- [astro](#)
- [engineering](#)
- [finance](#)
- [framework](#)
- [geo](#)
- [planetary](#)
- [solar](#)
- [utilities](#)

5.2 skdaccess.astro Namespace Reference

Namespaces

- [kepler](#)
- [spectra](#)
- [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.spectra Namespace Reference

Namespaces

- [stream](#)

5.6 skdaccess.astro.spectra.stream Namespace Reference

Classes

- class [DataFetcher](#)
Data Fetcher for Sloan Digital Sky Survey spectra.

5.7 skdaccess.astro.voyager Namespace Reference

Namespaces

- [data_fetcher](#)

5.8 skdaccess.astro.voyager.data_fetcher Namespace Reference

Classes

- class [DataFetcher](#)
Data Fetcher for Mahali temperature data.

5.9 skdaccess.engineering Namespace Reference

Namespaces

- [la](#)
- [webcam](#)

5.10 skdaccess.engineering.la Namespace Reference

Namespaces

- [generic](#)
- [traffic_counts](#)

5.11 skdaccess.engineering.la.generic Namespace Reference

Namespaces

- [stream](#)

5.12 skdaccess.engineering.la.generic.stream Namespace Reference

Classes

- class [DataFetcher](#)
Class for handling data requests to data.lacity.org.

5.13 skdaccess.engineering.la.traffic_counts Namespace Reference

Namespaces

- [stream](#)

5.14 skdaccess.engineering.la.traffic_counts.stream Namespace Reference

Classes

- class [DataFetcher](#)
[DataFetcher](#) for retrieving traffic counts from LA.

5.15 skdaccess.engineering.webcam Namespace Reference

Namespaces

- [mit_sailing](#)

5.16 skdaccess.engineering.webcam.mit_sailing Namespace Reference

Namespaces

- [stream](#)

5.17 skdaccess.engineering.webcam.mit_sailing.stream Namespace Reference

Classes

- class [DataFetcher](#)
Data Fetcher for retrieving webcam images from the MIT Sailing Pavilion.

5.18 skdaccess.finance Namespace Reference

Namespaces

- [timeseries](#)

5.19 skdaccess.finance.timeseries Namespace Reference

Namespaces

- [stream](#)

5.20 skdaccess.finance.timeseries.stream Namespace Reference

Classes

- class [DataFetcher](#)
Data Fetcher for retrieving stock data.

5.21 skdaccess.framework Namespace Reference

Namespaces

- [data_class](#)
- [param_class](#)

5.22 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.

5.23 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.

5.24 skdaccess.geo Namespace Reference

Namespaces

- [era_interim](#)
- [gldas](#)
- [grace](#)
- [groundwater](#)
- [imsdnhs](#)
- [magnetometer](#)
- [mahali](#)
- [modis](#)
- [ngl_gps](#)
- [pbo](#)
- [sentinel_1](#)
- [srtm](#)
- [uavsar](#)
- [wyoming_sounding](#)

5.25 skdaccess.geo.era_interim Namespace Reference

Namespaces

- [cache](#)

5.26 skdaccess.geo.era_interim.cache Namespace Reference

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
[DataFetcher](#) for retrieving ERA-I data.

5.28 skdaccess.geo.gldas Namespace Reference

Namespaces

- [data_fetcher](#)

5.29 skdaccess.geo.gldas.data_fetcher Namespace Reference

Classes

- class [DataFetcher](#)
Data Fetcher for GLDAS data.

5.30 skdaccess.geo.grace Namespace Reference

Namespaces

- [data_fetcher](#)
- [mascon](#)

5.31 skdaccess.geo.grace.data_fetcher Namespace Reference

Classes

- class [DataFetcher](#)
Data Fetcher for GRACE data.

5.32 skdaccess.geo.grace.mascon Namespace Reference

Namespaces

- [cache](#)

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

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher for GRACE mascon data.

5.35 skdaccess.geo.groundwater Namespace Reference

Namespaces

- [data_fetcher](#)

5.36 skdaccess.geo.groundwater.data_fetcher Namespace Reference

Classes

- class [DataFetcher](#)
Generates Data Wrappers of groundwater measurements taken in the US.

5.37 skdaccess.geo.imsdnhs Namespace Reference

Namespaces

- [data_fetcher](#)

5.38 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.

5.39 skdaccess.geo.magnetometer Namespace Reference

Namespaces

- [data_fetcher](#)

5.40 skdaccess.geo.magnetometer.data_fetcher Namespace Reference

Classes

- class [DataFetcher](#)
Data fetcher for USGS geomagnetic observatories.

5.41 skdaccess.geo.mahali Namespace Reference

Namespaces

- [rinex](#)
- [tec](#)
- [temperature](#)

5.42 skdaccess.geo.mahali.rinex Namespace Reference

Namespaces

- [data_fetcher](#)
- [data_wrapper](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher for Mahali Data.

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

Classes

- class [DataWrapper](#)
Data wrapper for Mahali data.

5.45 skdaccess.geo.mahali.tec Namespace Reference

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher for Mahali Data.

5.47 skdaccess.geo.mahali.temperature Namespace Reference

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher for Mahali temperature data.

5.49 skdaccess.geo.modis Namespace Reference

Namespaces

- [cache](#)
- [stream](#)

5.50 skdaccess.geo.modis.cache Namespace Reference

Namespaces

- [cloud_mask](#)
- [cloud_opacity](#)
- [data_fetcher](#)
- [reflectance](#)

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

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher for MODIS Cloud Mask.

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

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher for MODIS Cloud Opacity.

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

Classes

- class [DataFetcher](#)
Data Fetcher for MODIS data.

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

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data fetcher for the modis surface reflectance product ('09', 1 km resolution)

5.58 skdaccess.geo.modis.stream Namespace Reference

Namespaces

- [cloud_mask](#)
- [cloud_opacity](#)
- [data_fetcher](#)
- [reflectance](#)

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

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher for MODIS Cloud Mask.

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

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher for MODIS Cloud Opacity.

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

Classes

- class [DataFetcher](#)
Data Fetcher for MODIS data.

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

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)

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

5.66 skdaccess.geo.ngl_gps Namespace Reference

Namespaces

- [data_fetcher](#)

5.67 skdaccess.geo.ngl_gps.data_fetcher Namespace Reference

Classes

- class [DataFetcher](#)

Data fetcher for GPS data from Nevada Geodetic Laboratory.

5.68 skdaccess.geo.pbo Namespace Reference

Namespaces

- [data_fetcher](#)

5.69 skdaccess.geo.pbo.data_fetcher Namespace Reference

Classes

- class [DataFetcher](#)

Data fetcher for PBO GPS data.

5.70 skdaccess.geo.sentinel_1 Namespace Reference

Namespaces

- [cache](#)

5.71 skdaccess.geo.sentinel_1.cache Namespace Reference

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
[DataFetcher](#) for retrieving Sentinel SLC data.

5.73 skdaccess.geo.srtm Namespace Reference

Namespaces

- [cache](#)

5.74 skdaccess.geo.srtm.cache Namespace Reference

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
[DataFetcher](#) for retrieving data from the Shuttle Radar Topography Mission.

5.76 skdaccess.geo.uavsar Namespace Reference

Namespaces

- [cache](#)

5.77 skdaccess.geo.uavsar.cache Namespace Reference

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher for UAVSAR data.

5.79 skdaccess.geo.wyoming_sounding Namespace Reference

Namespaces

- [cache](#)
- [stream](#)

5.80 skdaccess.geo.wyoming_sounding.cache Namespace Reference

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
[DataFetcher](#) for retrieving Wyoming Sounding data.

5.82 skdaccess.geo.wyoming_sounding.stream Namespace Reference

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
[DataFetcher](#) for retrieving Wyoming Sounding data.

5.84 skdaccess.planetary Namespace Reference

Namespaces

- [ode](#)

5.85 skdaccess.planetary.ode Namespace Reference

Namespaces

- [cache](#)

5.86 skdaccess.planetary.ode.cache Namespace Reference

Namespaces

- [data_fetcher](#)

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

Classes

- class [DataFetcher](#)
Data Fetcher from the Orbital Data Explorer (ODE)

5.88 skdaccess.solar Namespace Reference

Namespaces

- [sdo](#)

5.89 skdaccess.solar.sdo Namespace Reference

Namespaces

- [data_fetcher](#)

5.90 skdaccess.solar.sdo.data_fetcher Namespace Reference

Classes

- class [DataFetcher](#)
Data Fetcher for the Solar Dynamics Observatory.

5.91 skdaccess.utilities Namespace Reference

Namespaces

- [file_browser](#)
- [file_util](#)
- [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](#)

5.92 skdaccess.utilities.file_browser Namespace Reference

Classes

- class [FileBrowser](#)

5.93 skdaccess.utilities.file_util Namespace Reference

Functions

- def [openPandasHDFStoreLocking](#) (filename, mode)

Open a pandas HDF store that may be locked:

5.93.1 Function Documentation

5.93.1.1 openPandasHDFStoreLocking()

```
def skdaccess.utilities.file_util.openPandasHDFStoreLocking (
    filename,
    mode )
```

Open a pandas HDF store that may be locked:

Parameters

<i>filename</i>	Name of file
<i>mode</i>	Mode (Such as read only, see Panda's documentation for flags)

Returns

Panda HDF store

5.94 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 [getStartDate](#) (in_data)

5.94.1 Function Documentation

5.94.1.1 averageDates()

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

Compute the average of a pandas series of timestamps.

Parameters

<i>dates</i>	Pandas series of pandas datetime objects
<i>round_nearest_day</i>	Round to the nearest day

Returns

Average of dates

5.94.1.2 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

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

Returns

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

5.94.1.3 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

<i>dates</i>	Iterable container of pandas timestamps
<i>days</i>	Number of days

Returns

true if they are not with 10 days, false otherwise

5.94.1.4 getStartEndDate()

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

5.94.1.5 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

<i>filename</i>	Name of file to read in
<i>lat_lon_list</i>	List of latitude, longitude tuples that are to be read

Parameters

<i>data_label</i>	Label for data
<i>lat_name</i>	Name of latitude data
<i>lon_name</i>	Name of longitude data
<i>data_name</i>	Name of data product
<i>time_name</i>	Name of time data
<i>lat_bounds_name</i>	Name of latitude boundaries
<i>lon_bounds_name</i>	Name of longitude boundaries
<i>uncertainty_name</i>	Name of uncertainty in data set
<i>lat_bounds</i>	Latitude bounds
<i>lon_bounds</i>	Longitude bounds

Returns

dictionary containing data and dictionary containing latitude and longitude

5.95 skdaccess.utilities.gw_util Namespace Reference

Functions

- def [combine_water_heights](#) (in_data)
Combine median and average water heights.

5.95.1 Function Documentation

5.95.1.1 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

<i>in_data</i>	Input water heights data
----------------	--------------------------

5.96 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](#)

5.96.1 Function Documentation

5.96.1.1 convertBinCentersToEdges()

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

Calculate edges of a set of bins from their centers.

Parameters

<i>bin_centers</i>	Array of bin centers
<i>dtype</i>	Data type of array used to store bin edges

Returns

bin_edges

5.96.1.2 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 )
```

5.96.1.3 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 pixel coordinate doesn't depend on the x projected coordinate

Parameters

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

Returns

list containing the 6 affine transformation coordinates

5.96.1.4 SplineGeolocation()

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

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

5.96.2 Variable Documentation

5.96.2.1 lat_spline

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

5.96.2.2 lon_spline

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

5.96.2.3 x_offset

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

5.96.2.4 x_spline

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

5.96.2.5 y_offset

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

5.96.2.6 y_spline

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

5.97 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.

5.97.1 Function Documentation

5.97.1.1 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

<i>in_data</i>	Data to be normalized
<i>column</i>	Name of column to be normalized
<i>group_column</i>	Name of column used to group data

5.98 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 [parselonoFile](#) (in_file, compression='infer')

5.98.1 Function Documentation

5.98.1.1 `convert_date()`

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

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

Parameters

<code>in_date</code>	Input date
----------------------	------------

return pandas data time object

5.98.1.2 `parseIonoFile()`

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

5.99 `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.

5.99.1 Function Documentation

5.99.1.1 `calibrateModis()`

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

This function calibrates input modis data.

Parameters

<i>data</i>	Input modis data
<i>metadata</i>	Metadata associated with modis input data

Returns

calibrated modis data

5.99.1.2 `checkBit()`

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

Get the bit value from a bit flag.

Parameters

<i>data</i>	Integer bit flag
<i>bit</i>	Which bit to select (start indexing at 0)

Returns

value of chosen bit in bit flag

5.99.1.3 `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

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

Returns

image subsection, fraction of valid data

5.99.1.4 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

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

Returns

list of file IDs

5.99.1.5 getFileURLs()

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

Retrieve the ftp location for a list of file IDs.

Parameters

<i>file_ids</i>	List of file IDs
-----------------	------------------

Returns

List of ftp locations

5.99.1.6 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

<i>in_data</i>	Input modis data
----------------	------------------

Returns

type of modis data

5.99.1.7 getModisData()

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

Loads modis data.

Parameters

<i>dataset</i>	netCDF4 dataset
<i>variable_name</i>	Name of variable to extract from dataset

Returns

(modis_data, metadata)

5.99.1.8 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

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

5.99.1.9 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

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

Returns

scaled data

5.100 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.

5.100.1 Function Documentation

5.100.1.1 `correct_CRISM_label()`

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

5.100.1.2 `correct_file_name_case_in_label()`

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

5.100.1.3 `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

<i>label_file_location</i>	Local address of the current label
<i>other_file_locations</i>	Other files that were downloaded with the label file

Returns

Local address of the new label

5.100.1.4 `get_files_urls()`

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

5.100.1.5 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 )
```

5.100.1.6 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

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

Returns

The array

5.100.1.7 get_raster_extent()

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

Get the extent of a raster opened with GDAL.

Parameters

<i>gdal_raster</i>	A raster opened with GDAL
--------------------	---------------------------

Returns

The raster extent

5.100.1.8 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

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

Returns

List of URL locations

5.100.1.9 query_yes_no()

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

5.101 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.

5.101.1 Function Documentation**5.101.1.1 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

<i>pbo_info</i>	PBO Metadata
<i>station_list</i>	List of stations

Returns

list containing two tuples, lat_range and lon_range

5.101.1.2 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

<i>geo_point</i>	The geographic (lat,lon) coordinate of interest
<i>radiusParam</i>	An overloaded radius of interest [km] or latitude and longitude window [deg] around the geo_point
<i>data</i>	Stabilized (or unstabilized) data generated from the data fetcher or out of stab_sys
<i>header</i>	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

5.101.1.3 getStationCoords()

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

Get the station coordinates for a list of stations.

Parameters

<i>pbo_info</i>	PBO Metadata
<i>station_list</i>	List of stations

Returns

list of tuples containing lat, lon coordinates of stations

5.101.1.4 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

<i>allH</i>	a dictionary of all of the headers of all sites loaded from the data directory
<i>allD</i>	a dictionary of all of the panda format data of all of the corresponding sites
<i>timerng</i>	an array with two string elements, describing the starting and ending dates
<i>indx</i>	a list of site 4ID's indicating stations in the relevant geographic location, or 1 for all sites
<i>mdyratio</i>	optional parameter for the minimum required ratio of data to determine if a sitef is kept for further analysis
<i>use_progress_bar</i>	Display a progress bar
<i>index_date_only</i>	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

5.101.1.5 propagateErrors()

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

```

    SC,
    stationCovs )

```

Propagate GPS errors.

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

Parameters

<i>R</i>	Rotation matrix
<i>sc</i>	Scaling value
<i>stationCovs</i>	Station Covariances

5.101.1.6 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

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

Returns

GPS data with the offsets removed

5.101.1.7 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 implementation of the Helmert 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

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

5.102 skdaccess.utilities.sentinel_1_util Namespace Reference

Functions

- def [parseSatelliteData](#) (in_satellite_file)
Parse Sentinel satellite data.

5.102.1 Function Documentation

5.102.1.1 parseSatelliteData()

```

def skdaccess.utilities.sentinel_1_util.parseSatelliteData (
    in_satellite_file )

```

Parse Sentinel satellite data.

Parameters

<i>in_satellite_file</i>	Satellite orbit filename
--------------------------	--------------------------

Returns

DataFrame of orbit information

5.103 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.

5.103.1 Function Documentation

5.103.1.1 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

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

Returns

list of urls containing requested data

5.104 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.

5.104.1 Function Documentation

5.104.1.1 [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

<i>srtmdw</i>	SRTM data wrapper
<i>lat_start</i>	Starting latiude
<i>lat_end</i>	Ending latiude
<i>lon_start</i>	Starting longitude
<i>lon_end</i>	Ending longitude
<i>flip_y</i>	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

5.104.1.2 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

<i>lat_min</i>	Minimum latitude
<i>lat_max</i>	Maximum latitude
<i>lon_min</i>	Minimum longitude
<i>lon_max</i>	Maximum longitude

Returns

(starting_latitude, ending_latitude, starting_longitude, ending_longitude)

5.104.1.3 merge_srtm_tiles()

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

5.105 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)
Progress bar using tqdm.
- def [convertToStr](#) (in_value, zfill=0)
- def [join_string](#) (part1, part2, concatenation_string='AND', seperator=' ')
Join two strings together using a concatenation string.

5.105.1 Function Documentation

5.105.1.1 convertToStr()

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

5.105.1.2 join_string()

```
def skdaccess.utilities.support.join_string (
    part1,
    part2,
    concatenation_string = 'AND',
    seperator = ' ' )
```

Join two strings together using a concatenation string.

Handles the case where either part1 or part2 are an empty string

Parameters

<i>part1</i>	First string
<i>part2</i>	Second string
<i>concatenation_string</i>	String used to join part1 and part2
<i>seperator</i>	Seperator used to between each part and the concatenation string

Returns

A single string that consists of the part1 and part2 joined together using a concatenation string

5.105.1.3 progress_bar()

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

Progress bar using tqdm.

Parameters

<i>in_iterable</i>	Input iterable
<i>total</i>	Total number of elements
<i>enabled</i>	Enable progress bar

5.105.1.4 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

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

Returns

dictionary of dates with data

5.106 skdaccess.utilities.uavsar_util Namespace Reference

Functions

- def [readUAVSARMetadata](#) (in_file)
Parse UAVSAR metadata.

5.106.1 Function Documentation

5.106.1.1 readUAVSARMetadata()

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

Parse UAVSAR metadata.

Parameters

<i>in_file</i>	String of Metadata filename or file object (file should end in .ann)
----------------	--

Returns

OrderedDict of metadata

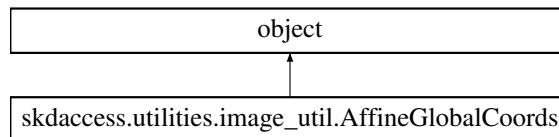
Chapter 6

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

<i>aff_coeffs</i>	Affine coefficients
<i>center_pixels</i>	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

<i>y_proj</i>	Input projected y coordinates
<i>x_proj</i>	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

<code>y_array</code>	Input y pixel coordinates
<code>x_array</code>	Input x pixel coordinates

Returns

projected y coordinates, projected x coordinates

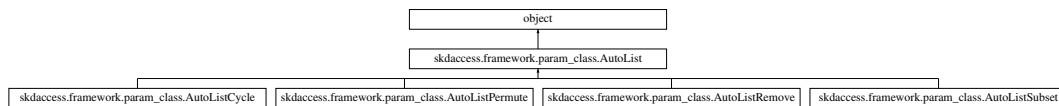
The documentation for this class was generated from the following file:

- [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

<i>val_list</i>	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

<i>ii</i>	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

<i>ii</i>	Index of list to be set
<i>val</i>	Input value

6.2.3.5 `__str__()`

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

String representation of class.

Returns

String containing all parameters 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
```

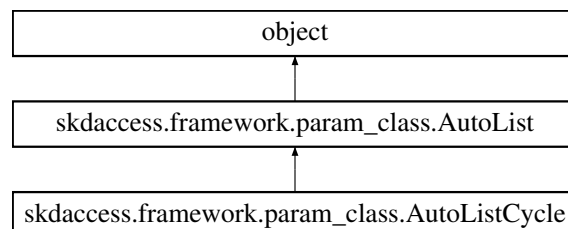
The documentation for this class was generated from the following file:

- 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

<i>list_val_list</i>	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

<i>ii</i>	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

<i>ii</i>	Index of list to be set
<i>val</i>	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`

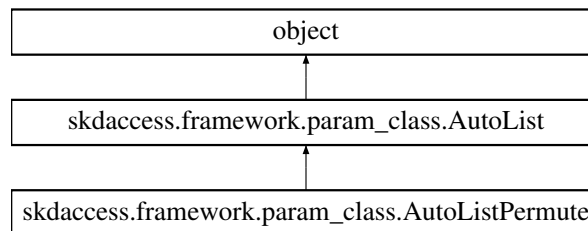
The documentation for this class was generated from the following file:

- [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

<i>ii</i>	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

<i>ii</i>	Index of list to be set
<i>val</i>	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]

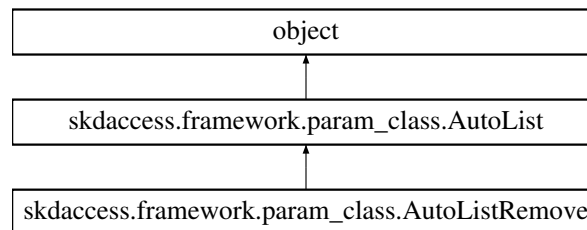
The documentation for this class was generated from the following file:

- [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

<code>val_list</code>	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

<i>ii</i>	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

<i>ii</i>	Index of list to be set
<i>val</i>	Input value

6.5.3.5 `__str__()`

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

String representation of class.

Returns

String containing all parameters 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
```

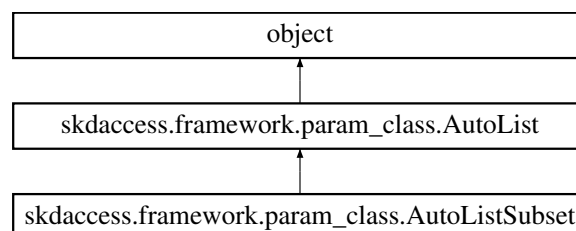
The documentation for this class was generated from the following file:

- 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)
Perturb 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

<i>ii</i>	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

<i>ii</i>	Index of list to be set
<i>val</i>	Input value

6.6.2.5 `__str__()`

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

String representation of class.

Returns

String containing all parameters 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 )
```

Perturb 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
```

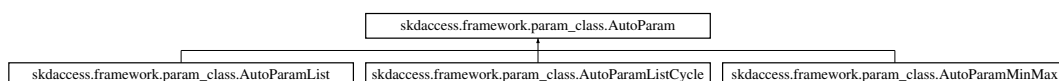
The documentation for this class was generated from the following file:

- [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

<code>val_init</code>	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`

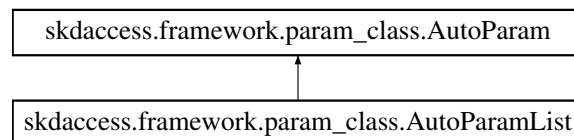
The documentation for this class was generated from the following file:

- [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

<i>val_init</i>	initial value for the parameter
<i>val_list</i>	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`

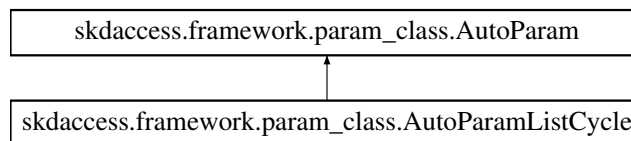
The documentation for this class was generated from the following file:

- [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

<i>val_list</i>	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
```

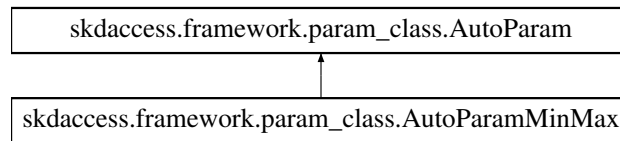
The documentation for this class was generated from the following file:

- 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)
Perturb the parameter 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

<i>val_init</i>	Initial value for parameter
<i>val_min</i>	Minimum value for param
<i>val_max</i>	Maximum value for parameter
<i>decimals</i>	Number of decimals to include in the random number
<i>extreme</i>	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 )
```

Perturb the parameter 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
```

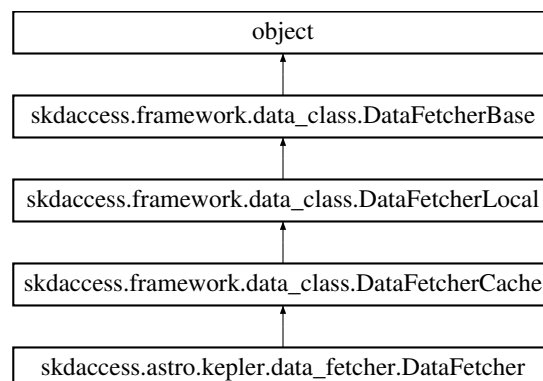
The documentation for this class was generated from the following file:

- [framework/param_class.py](#)

6.11 skdaccess.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 `getHDFSStorage` (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 `getConfigItem` (`section`, `key`)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (`section`, `key`, `value`)
Retrieve skdaccess configuration item.
- 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`

6.11.1 Detailed Description

Data Fetcher for Kepler light curve data.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 `__init__()`

```
def skdaccess.astro.kepler.DataFetcher.__init__ (
    self,
    ap_paramList,
    quarter_list = None )
```

Initialize Kepler Data Fetcher.

Parameters

<i>ap_paramList[kepler_id_list]</i>	List of kepler id's
<i>quarter_list</i>	List of quarters (0-17) (default: all quarters)

6.11.3 Member Function Documentation

6.11.3.1 `__str__()`

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

Generate string description.

6.11.3.2 `cacheData()` [1/2]

```
def skdaccess.astro.kepler.DataFetcher.cacheData (
    self,
    data_specification )
```

Cache Kepler data locally.

Parameters

<i>data_specification</i>	List of kepler IDs
---------------------------	--------------------

6.11.3.3 `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

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

Returns

List of downloaded file locations

6.11.3.4 `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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.11.3.5 downloadKeplerData()

```
def skdaccess.astro.kepler.DataFetcher.downloadKeplerData (
    self,
    kid_list )
```

Download and parse Kepler data for a list of kepler id's.

Parameters

<i>kid_list</i>	List of Kepler ID's to download
-----------------	---------------------------------

Returns

dictionary of kepler data

6.11.3.6 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.11.3.7 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.11.3.8 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.11.3.9 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.11.3.10 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.11.3.11 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

6.11.3.12 output()

```
def skdaccess.astro.kepler.DataFetcher.output (
    self )
```

Output kepler data wrapper.

Returns

DataWrapper

6.11.3.13 perturb()

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

Perturb parameters.

6.11.3.14 reset()

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

Set all parameters to initial value.

6.11.3.15 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.11.3.16 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.11.3.17 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.11.3.18 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.11.4 Member Data Documentation

6.11.4.1 ap_paramList

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

6.11.4.2 quarter_list

```
skdaccess.astro.kepler.DataFetcher.quarter_list
```

6.11.4.3 verbose

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

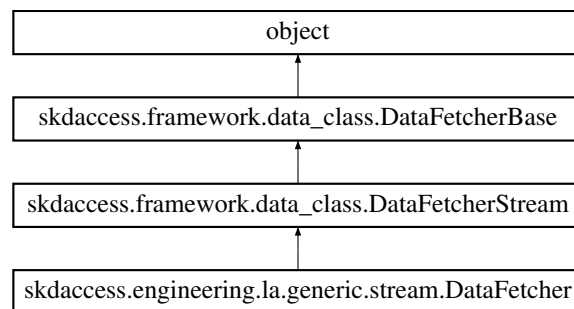
The documentation for this class was generated from the following file:

- [astro/kepler/data_fetcher.py](#)

6.12 skdaccess.engineering.la.generic.stream.DataFetcher Class Reference

Class for handling data requests to data.lacity.org.

Inheritance diagram for `skdaccess.engineering.la.generic.stream.DataFetcher`:



Public Member Functions

- `def __init__ (self, endpoint, parameters, label, verbose=False, app_token=None, date_columns=None, pandas↔
_kwargs)`
Initialize Data Fetcher for accessing data.lacity.org.
- `def output (self)`
Retrieve data from data.lacity.org.
- `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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- def `writeConfig` (conf)
Write config to disk.
- def `verbose_print` (self, args, kwargs)
Print statement if verbose flag is set.

Public Attributes

- `base_url`
- `base_url_and_endpoint`
- `parameters`
- `label`
- `app_token`
- `pandas_kwargs`
- `ap_paramList`
- `verbose`

6.12.1 Detailed Description

Class for handling data requests to data.lacity.org.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 `__init__()`

```
def skdaccess.engineering.la.generic.stream.DataFetcher.__init__ (
    self,
    endpoint,
    parameters,
    label,
    verbose = False,
    app_token = None,
    date_columns = None,
    pandas_kwargs )
```

Initialize Data Fetcher for accessing data.lacity.org.

Parameters

<code>endpoint</code>	Data endpoint string
<code>parameters</code>	Parameters to use when retrieving dta
<code>label</code>	Label of pandas dataframe
<code>verbose</code>	Print out extra information
<code>app_token</code>	Application token to use to avoid throttling issues
<code>pandas_kwargs</code>	Any additional key word arguments are passed to pandas.read_csv

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 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.12.3.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.12.3.4 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.12.3.5 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

6.12.3.6 output()

```
def skdaccess.engineering.la.generic.stream.DataFetcher.output (
    self )
```

Retrieve data from data.lacity.org.

Returns

Table wrapper of containing specified data

6.12.3.7 perturb()

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

Perturb parameters.

6.12.3.8 reset()

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

Set all parameters to initial value.

6.12.3.9 retrieveOnlineData()

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

Method for downloading data into memory.

Parameters

<i>data_specification</i>	Url list of data to be retrieved
---------------------------	----------------------------------

Returns

Retrieved data

6.12.3.10 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.12.3.11 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
```

```
conf ) [inherited]
```

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.12.3.12 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.12.4 Member Data Documentation

6.12.4.1 ap_paramList

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

6.12.4.2 app_token

```
skdaccess.engineering.la.generic.stream.DataFetcher.app_token
```

6.12.4.3 base_url

```
skdaccess.engineering.la.generic.stream.DataFetcher.base_url
```

6.12.4.4 base_url_and_endpoint

```
skdaccess.engineering.la.generic.stream.DataFetcher.base_url_and_endpoint
```

6.12.4.5 label

```
skdaccess.engineering.la.generic.stream.DataFetcher.label
```

6.12.4.6 pandas_kwargs

```
skdaccess.engineering.la.generic.stream.DataFetcher.pandas_kwargs
```

6.12.4.7 parameters

```
skdaccess.engineering.la.generic.stream.DataFetcher.parameters
```

6.12.4.8 verbose

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

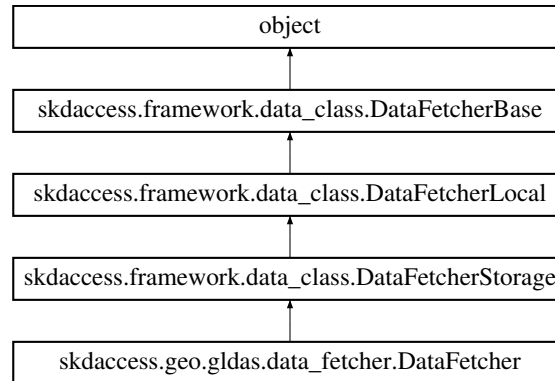
The documentation for this class was generated from the following file:

- [engineering/la/generic/stream.py](#)

6.13 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 getConfigItem (section, key)`
Retrieve skdaccess configuration item.
- `def writeConfigItem (section, key, value)`
Retrieve skdaccess configuration item.
- `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](#)

6.13.1 Detailed Description

Data Fetcher for GLDAS data.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 `__init__()`

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

Construct a GLDAS Data Fetcher.

Parameters

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

6.13.3 Member Function Documentation

6.13.3.1 `__str__()`

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

String representation of data fetcher.

Returns

String listing the name and geopoint of data fetcher

6.13.3.2 downloadFullDataset()

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

Download GLDAS data.

Parameters

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

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 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.13.3.5 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.13.3.6 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

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.gldas.DataFetcher.output (
    self )
```

Create data wrapper of GLDAS data for specified geoint.

Returns

GLDAS 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

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	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

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	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

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.13.3.14 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.13.4 Member Data Documentation**6.13.4.1 `ap_paramList`**

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

6.13.4.2 `end_date`

`skdaccess.geo.gldas.DataFetcher.end_date`

6.13.4.3 `resample`

`skdaccess.geo.gldas.DataFetcher.resample`

6.13.4.4 `start_date`

`skdaccess.geo.gldas.DataFetcher.start_date`

6.13.4.5 `verbose`

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

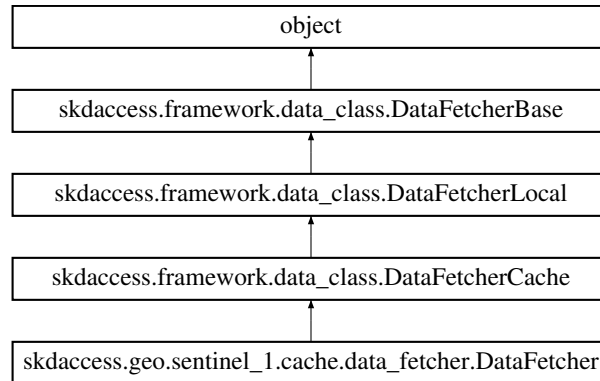
The documentation for this class was generated from the following file:

- [geo/gldas/data_fetcher.py](#)

6.14 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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.14.1 Detailed Description

[DataFetcher](#) for retrieving Sentinel SLC data.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 `__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

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

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

<i>keyname</i>	Name of dataset in configuration file
<i>online_path_list</i>	List of urls to data
<i>username</i>	Username for accessing online resources
<i>password</i>	Password for accessing online resources
<i>authentication_url</i>	The url used for authentication (unused when use_requests=True)
<i>cookiejar</i>	The cookiejar that stores credentials (unused when use_requests=True)
<i>use_requests</i>	Use the requests library instead of the standard library for accessing resources
<i>use_progress_bar</i>	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

<i>in_file_name</i>	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 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.14.3.6 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.14.3.7 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.14.3.8 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.14.3.9 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

6.14.3.10 output()

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

Generate data wrapper.

Returns

Sentinel SLC data in a data wrapper

6.14.3.11 perturb()

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

Perturb parameters.

6.14.3.12 reset()

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

Set all parameters to initial value.

6.14.3.13 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.14.3.14 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.14.3.15 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.14.3.16 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.14.4 Member Data Documentation

6.14.4.1 ap_paramList

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

6.14.4.2 local_paths

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

6.14.4.3 password

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

6.14.4.4 polarization

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

6.14.4.5 satellite_url_list

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

6.14.4.6 swath

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

6.14.4.7 url_list

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

6.14.4.8 username

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

6.14.4.9 verbose

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

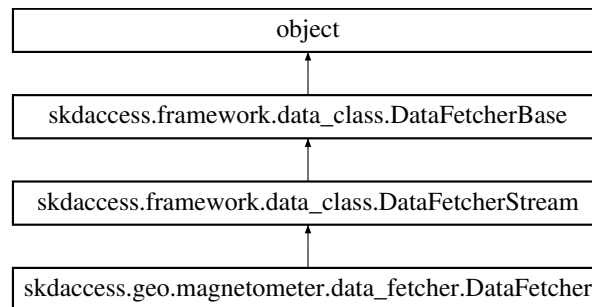
The documentation for this class was generated from the following file:

- `geo/sentinel_1/cache/data_fetcher.py`

6.15 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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.15.1 Detailed Description

Data fetcher for USGS geomagnetic observatories.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 `__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

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

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 `getConfig()`

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.15.3.3 `getConfigItem()`

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.15.3.4 `getDataMetadata()`

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

Get data metadata.

Returns

Pandas dataframe containing station latitude and longitude coordinates

6.15.3.5 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.15.3.6 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

6.15.3.7 output()

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

Generate data wrapper for USGS geomagnetic data.

Returns

geomagnetic data wrapper

6.15.3.8 perturb()

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

Perturb parameters.

6.15.3.9 reset()

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

Set all parameters to initial value.

6.15.3.10 retrieveOnlineData()

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

Method for downloading data into memory.

Parameters

<i>data_specification</i>	Url list of data to be retrieved
---------------------------	----------------------------------

Returns

Retrieved data

6.15.3.11 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.15.3.12 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.15.3.13 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.15.4 Member Data Documentation

6.15.4.1 `ap_paramList`

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

6.15.4.2 `channels`

`skdaccess.geo.magnetometer.DataFetcher.channels`

6.15.4.3 `data_type`

`skdaccess.geo.magnetometer.DataFetcher.data_type`

6.15.4.4 `end_time`

`skdaccess.geo.magnetometer.DataFetcher.end_time`

6.15.4.5 `interval`

`skdaccess.geo.magnetometer.DataFetcher.interval`

6.15.4.6 `start_time`

`skdaccess.geo.magnetometer.DataFetcher.start_time`

6.15.4.7 `verbose`

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

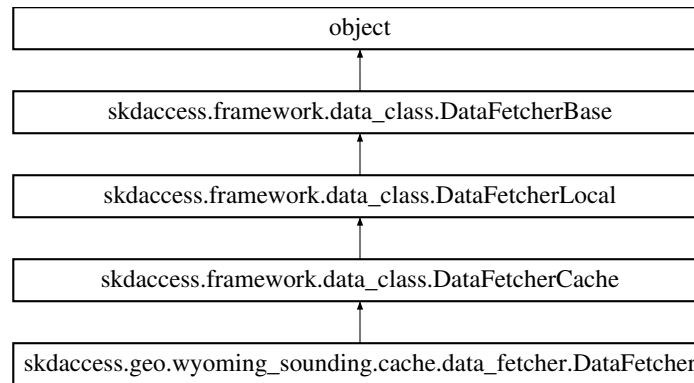
The documentation for this class was generated from the following file:

- `geo/magnetometer/data_fetcher.py`

6.16 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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.16.1 Detailed Description

[DataFetcher](#) for retrieving Wyoming Sounding data.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 `__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

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

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 `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

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

Returns

List of downloaded file locations

6.16.3.3 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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.16.3.4 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.16.3.5 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.16.3.6 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.16.3.7 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.16.3.8 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.16.3.9 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

6.16.3.10 output()

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

Generate data wrapper.

Returns

Wyoming sounding data in a data wrapper

6.16.3.11 perturb()

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

Perturb parameters.

6.16.3.12 reset()

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

Set all parameters to initial value.

6.16.3.13 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.16.3.14 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.16.3.15 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.16.3.16 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.16.4 Member Data Documentation

6.16.4.1 ap_paramList

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

6.16.4.2 day_end

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

6.16.4.3 day_start

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

6.16.4.4 end_hour

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

6.16.4.5 month_list

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

6.16.4.6 start_hour

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

6.16.4.7 station_number

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

6.16.4.8 verbose

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

6.16.4.9 year_list

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

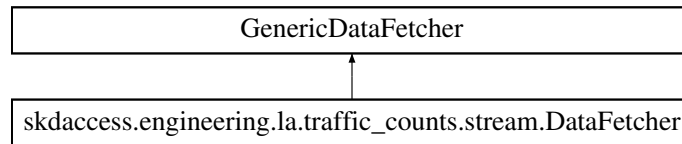
The documentation for this class was generated from the following file:

- `geo/wyoming_sounding/cache/data_fetcher.py`

6.17 skdaccess.engineering.la.traffic_counts.stream.DataFetcher Class Reference

[DataFetcher](#) for retrieving traffic counts from LA.

Inheritance diagram for skdaccess.engineering.la.traffic_counts.stream.DataFetcher:



Public Member Functions

- `def __init__(self, limit=None, start_time=None, end_time=None, app_token=None, verbose=False)`
Initialize Data Fetcher to retrieve traffic couns from LA.

6.17.1 Detailed Description

[DataFetcher](#) for retrieving traffic counts from LA.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 __init__()

```

def skdaccess.engineering.la.traffic_counts.stream.DataFetcher.__init__(
    self,
    limit = None,
    start_time = None,
    end_time = None,
    app_token = None,
    verbose = False )
  
```

Initialize Data Fetcher to retrieve traffic couns from LA.

Parameters

<i>limit</i>	Maximum number of rows
<i>start_time</i>	Starting time
<i>end_time</i>	Ending time
<i>app_token</i>	Application token to avoid throttling
<i>verbose</i>	Print extra information

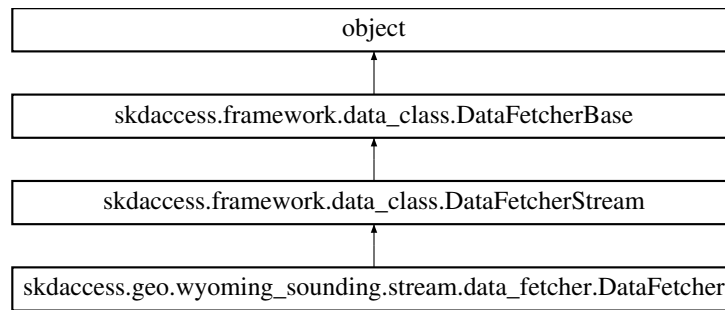
The documentation for this class was generated from the following file:

- [engineering/la/traffic_counts/stream.py](#)

6.18 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 getConfigItem(section, key)`
Retrieve skdaccess configuration item.
- `def writeConfigItem(section, key, value)`
Retrieve skdaccess configuration item.
- `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.stream.DataFetcher.__init__ (
    self,
    station_number,
    year,
    month,
    day_start,
    day_end,
    start_hour = 0,
    end_hour = 12 )
```

Initialize Data Fetcher.

Parameters

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

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.18.3.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.18.3.4 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.18.3.5 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

6.18.3.6 output() [1/2]

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

Output data wrapper.

Returns

Datawrapper

6.18.3.7 output() [2/2]

```
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

6.18.3.8 perturb()

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

Perturb parameters.

6.18.3.9 reset()

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

Set all parameters to initial value.

6.18.3.10 retrieveOnlineData()

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

Method for downloading data into memory.

Parameters

<i>data_specification</i>	Url list of data to be retrieved
---------------------------	----------------------------------

Returns

Retrieved data

6.18.3.11 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.18.3.12 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.18.3.13 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

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.stream.DataFetcher.day_end

6.18.4.3 day_start

skdaccess.geo.wyoming_sounding.stream.DataFetcher.day_start

6.18.4.4 end_hour

skdaccess.geo.wyoming_sounding.stream.DataFetcher.end_hour

6.18.4.5 month_list

skdaccess.geo.wyoming_sounding.stream.DataFetcher.month_list

6.18.4.6 start_hour

skdaccess.geo.wyoming_sounding.stream.DataFetcher.start_hour

6.18.4.7 station_number

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

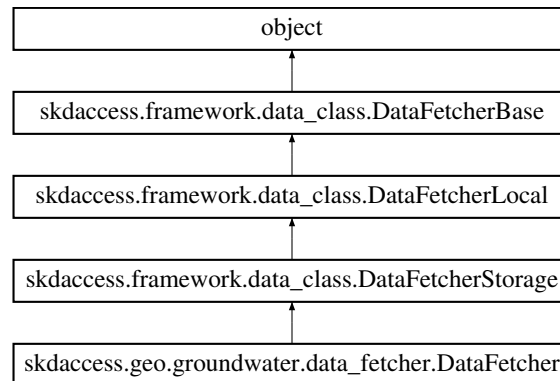
The documentation for this class was generated from the following file:

- [geo/wyoming_sounding/stream/data_fetcher.py](#)

6.19 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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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`

6.19.1 Detailed Description

Generates Data Wrappers of groundwater measurements taken in the US.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 `__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

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

6.19.3 Member Function Documentation

6.19.3.1 `__str__()`

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

String representation of data fetcher.

Returns

string describing data fetcher

6.19.3.2 `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

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

Returns

Absolute path of parsed data

6.19.3.3 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.19.3.4 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.19.3.5 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.19.3.6 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.19.3.7 getStationMetadata()

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

Retrieve metadata on groundwater wells.

Returns

pandas dataframe with groundwater well information

6.19.3.8 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

6.19.3.9 output()

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

Fetch Groundwater Data Wrapper.

Returns

Groundwater Data Wrapper

6.19.3.10 perturb()

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

Perturb parameters.

6.19.3.11 reset()

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

Set all parameters to initial value.

6.19.3.12 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.19.3.13 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.19.3.14 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.19.3.15 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.19.4 Member Data Documentation

6.19.4.1 ap_paramList

`skdaccess.geo.groundwater.DataFetcher.ap_paramList`

6.19.4.2 cutoff

`skdaccess.geo.groundwater.DataFetcher.cutoff`

6.19.4.3 end_date

`skdaccess.geo.groundwater.DataFetcher.end_date`

6.19.4.4 start_date

`skdaccess.geo.groundwater.DataFetcher.start_date`

6.19.4.5 verbose

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

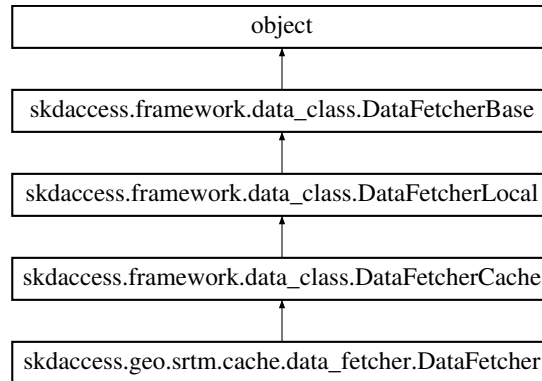
The documentation for this class was generated from the following file:

- `geo/groundwater/data_fetcher.py`

6.20 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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.20.1 Detailed Description

[DataFetcher](#) for retrieving data from the Shuttle Radar Topography Mission.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 `__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

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

6.20.3 Member Function Documentation

6.20.3.1 __str__()

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

Generate string description.

6.20.3.2 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

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

Returns

List of downloaded file locations

6.20.3.3 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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.20.3.4 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.20.3.5 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.20.3.6 `getDataLocation()`

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.20.3.7 `getHDFStorage()`

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.20.3.8 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.20.3.9 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

6.20.3.10 output()

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

Generate SRTM data wrapper.

Returns

SRTM Image Wrapper

6.20.3.11 perturb()

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

Perturb parameters.

6.20.3.12 reset()

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

Set all parameters to initial value.

6.20.3.13 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.20.3.14 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.20.3.15 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.20.3.16 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.20.4 Member Data Documentation

6.20.4.1 ap_paramList

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

6.20.4.2 arcsecond_sampling

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

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

Parameters

<i>in_filename</i>	Input SRTM filename
--------------------	---------------------

Returns

Latitude of southwest corner, Longitude of southwest corner

6.20.4.3 lat_tile_end

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

6.20.4.4 lat_tile_start

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

6.20.4.5 lon_tile_end

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

6.20.4.6 lon_tile_start

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

6.20.4.7 mask_water

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

6.20.4.8 password

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

6.20.4.9 store_geolocation_grids

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

6.20.4.10 username

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

6.20.4.11 verbose

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

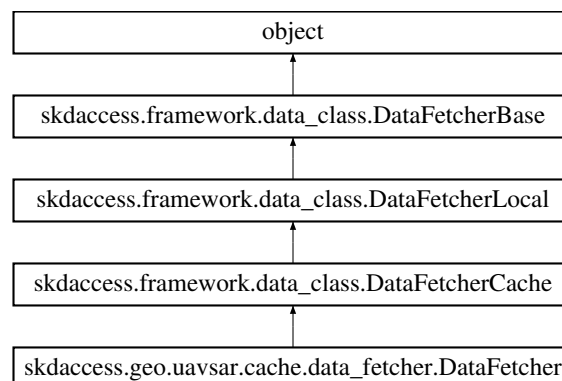
The documentation for this class was generated from the following file:

- [geo/srtm/cache/data_fetcher.py](#)

6.21 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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.21.1 Detailed Description

Data Fetcher for UAVSAR data.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 __init__()

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

Initialize UAVSAR data fetcher.

Parameters

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

6.21.3 Member Function Documentation

6.21.3.1 __str__()

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

Generate string description.

6.21.3.2 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

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

Returns

List of downloaded file locations

6.21.3.3 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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.21.3.4 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.21.3.5 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.21.3.6 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.21.3.7 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.21.3.8 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.21.3.9 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

6.21.3.10 output()

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

Output data as a data wrapper.

Returns

Imagewrapper of data

6.21.3.11 perturb()

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

Perturb parameters.

6.21.3.12 reset()

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

Set all parameters to initial value.

6.21.3.13 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.21.3.14 `verbose_print()`

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.21.3.15 `writeConfig()`

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.21.3.16 `writeConfigItem()`

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.21.4 Member Data Documentation**6.21.4.1 ap_paramList**

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

6.21.4.2 llh_url

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

6.21.4.3 memmap

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

6.21.4.4 metadata_url_list

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

6.21.4.5 slc_url_list

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

6.21.4.6 verbose

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

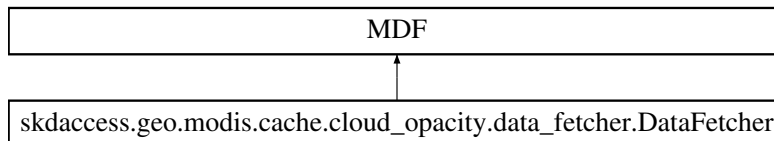
The documentation for this class was generated from the following file:

- `geo/uavsar/cache/data_fetcher.py`

6.22 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.

6.22.1 Detailed Description

Data Fetcher for MODIS Cloud Opacity.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 __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

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

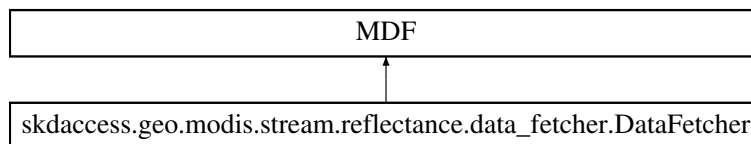
The documentation for this class was generated from the following file:

- [geo/modis/cache/cloud_opacity/data_fetcher.py](#)

6.23 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.

6.23.1 Detailed Description

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

6.23.2 Constructor & Destructor Documentation

6.23.2.1 `__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

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

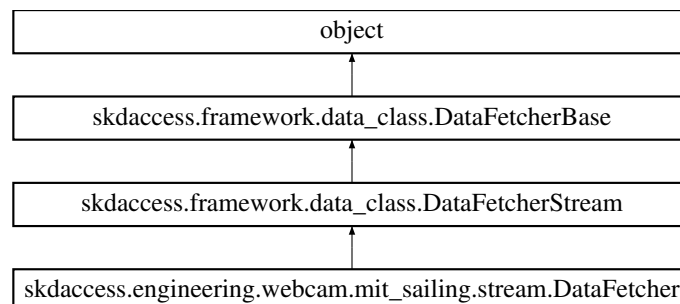
The documentation for this class was generated from the following file:

- [geo/modis/stream/reflectance/data_fetcher.py](#)

6.24 `skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher` Class Reference

Data Fetcher for retrieving webcam images from the MIT Sailing Pavilion.

Inheritance diagram for `skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher`:



Public Member Functions

- def `__init__` (self, `camera_list`=['E', SE, SW, W])
- def `output` (self)
Retrieve data from webcams at the MIT Sailing Pavilion.
- 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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- def `writeConfig` (conf)
Write config to disk.
- def `verbose_print` (self, args, kwargs)
Print statement if verbose flag is set.

Public Attributes

- `camera_list`
- `ap_paramList`
- `verbose`

6.24.1 Detailed Description

Data Fetcher for retrieving webcam images from the MIT Sailing Pavilion.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 `__init__()`

```
def skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher.__init__ (
    self,
    camera_list = ['E',
                  SE,
                  SW,
                  W ]
)
```

Parameters

<i>camera_list</i>	Which camera to retrieve from (List that contains one or more of the following: 'E', 'SE', 'SW', or 'W')
--------------------	--

6.24.3 Member Function Documentation**6.24.3.1 __str__()**

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

Generate string description.

6.24.3.2 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.24.3.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.24.3.4 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.24.3.5 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

6.24.3.6 output()

```
def skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher.output (
    self )
```

Retrieve data from webcams at the MIT Sailing Pavilion.

Returns

Image Wrapper containing the latest images from the webcams

6.24.3.7 perturb()

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

Perturb parameters.

6.24.3.8 reset()

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

Set all parameters to initial value.

6.24.3.9 retrieveOnlineData()

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

Method for downloading data into memory.

Parameters

<i>data_specification</i>	Url list of data to be retrieved
---------------------------	----------------------------------

Returns

Retrieved data

6.24.3.10 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.24.3.11 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.24.3.12 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.24.4 Member Data Documentation

6.24.4.1 `ap_paramList`

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

6.24.4.2 `camera_list`

`skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher.camera_list`

6.24.4.3 `verbose`

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

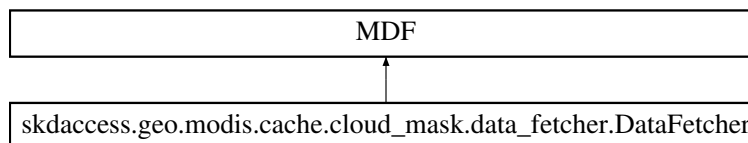
The documentation for this class was generated from the following file:

- [engineering/webcam/mit_sailing/stream.py](#)

6.25 `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.

6.25.1 Detailed Description

Data Fetcher for MODIS Cloud Mask.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 `__init__()`

```
def skdaccess.geo.modis.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

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

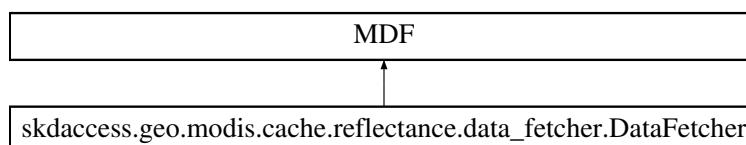
The documentation for this class was generated from the following file:

- [geo/modis/cache/cloud_mask/data_fetcher.py](#)

6.26 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.

6.26.1 Detailed Description

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

6.26.2 Constructor & Destructor Documentation

6.26.2.1 __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

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

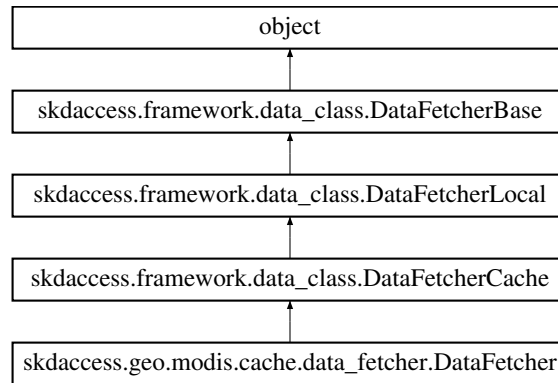
The documentation for this class was generated from the following file:

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

6.27 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 getHDFSStorage (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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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.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

<i>ap_paramList[lat]</i>	Search latitude
<i>ap_paramList[lon]</i>	Search longitude
<i>modis_platform</i>	Platform (Either "Terra" or "Aqua")
<i>modis_id</i>	Product string (e.g. '06_L2')
<i>variable_list</i>	List of variables to fetch
<i>start_date</i>	Starting date
<i>end_date</i>	Ending date
<i>daynightboth</i>	Use daytime data ('D'), nighttime data ('N') or both ('B')
<i>grid</i>	Further divide each image into a multiple grids of size (y,x)
<i>grid_fill</i>	Fill value to use when creating gridded data
<i>use_long_name</i>	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 cacheData() [1/2]

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

Download MODIS data.

Parameters

<i>data_specification</i>	List of file IDs to cache
---------------------------	---------------------------

6.27.3.3 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

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

Returns

List of downloaded file locations

6.27.3.4 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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.27.3.5 find_data()

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

Finds files previously downloaded files associated with fileids.

Parameters

<i>fileid_list</i>	List of file id's
<i>file_object</i>	File object to read from

Returns

Pandas series of file locaitons indexed by file id

6.27.3.6 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.27.3.7 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.27.3.8 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.27.3.9 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.27.3.10 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.27.3.11 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

6.27.3.12 output()

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

Generate data wrapper.

Returns

data wrapper of MODIS data

6.27.3.13 perturb()

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

Perturb parameters.

6.27.3.14 reset()

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

Set all parameters to initial value.

6.27.3.15 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.27.3.16 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.27.3.17 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.27.3.18 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

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.cache.DataFetcher.daynightboth

6.27.4.3 end_date

skdaccess.geo.modis.cache.DataFetcher.end_date

6.27.4.4 grid

skdaccess.geo.modis.cache.DataFetcher.grid

6.27.4.5 grid_fill

skdaccess.geo.modis.cache.DataFetcher.grid_fill

6.27.4.6 modis_id

skdaccess.geo.modis.cache.DataFetcher.modis_id

6.27.4.7 modis_identifier

skdaccess.geo.modis.cache.DataFetcher.modis_identifier

6.27.4.8 modis_platform

`skdaccess.geo.modis.cache.DataFetcher.modis_platform`

6.27.4.9 start_date

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

6.27.4.10 use_long_name

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

6.27.4.11 variable_list

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

6.27.4.12 verbose

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

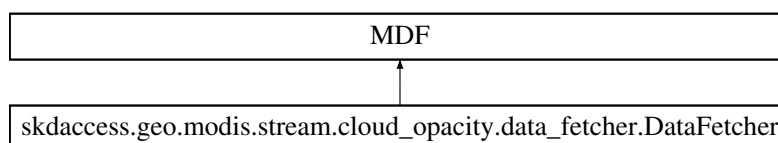
The documentation for this class was generated from the following file:

- [geo/modis/cache/data_fetcher.py](#)

6.28 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.

6.28.1 Detailed Description

Data Fetcher for MODIS Cloud Opacity.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 __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

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

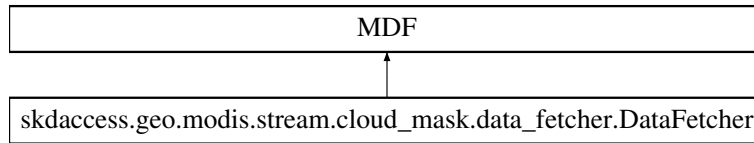
The documentation for this class was generated from the following file:

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

6.29 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.

6.29.1 Detailed Description

Data Fetcher for MODIS Cloud Mask.

6.29.2 Constructor & Destructor Documentation

6.29.2.1 `__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

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

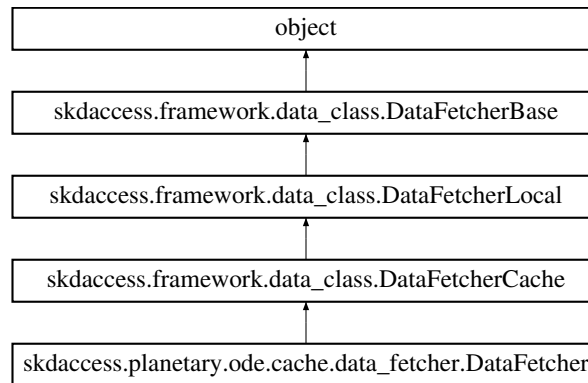
The documentation for this class was generated from the following file:

- [geo/modis/stream/cloud_mask/data_fetcher.py](#)

6.30 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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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`

6.30.1 Detailed Description

Data Fetcher from the Orbital Data Explorer (ODE)

6.30.2 Constructor & Destructor Documentation

6.30.2.1 `__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 )
```

6.30.3 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

<i>keyname</i>	Name of dataset in configuration file
<i>online_path_list</i>	List of urls to data
<i>username</i>	Username for accessing online resources
<i>password</i>	Password for accessing online resources
<i>authentication_url</i>	The url used for authentication (unused when use_requests=True)
<i>cookiejar</i>	The cookiejar that stores credentials (unused when use_requests=True)
<i>use_requests</i>	Use the requests library instead of the standard library for accessing resources
<i>use_progress_bar</i>	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

<i>in_file_name</i>	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 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.30.3.6 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.30.3.7 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.30.3.8 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.30.3.9 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

6.30.3.10 output()

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

Generate data wrapper from ODE data.

6.30.3.11 perturb()

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

Perturb parameters.

6.30.3.12 reset()

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

Set all parameters to initial value.

6.30.3.13 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.30.3.14 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.30.3.15 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.30.3.16 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.30.4 Member Data Documentation

6.30.4.1 `ap_paramList`

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

6.30.4.2 `eastern_lon`

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

6.30.4.3 `file_name`

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

6.30.4.4 `instrument`

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

6.30.4.5 `max_lat`

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

6.30.4.6 `max_ob_time`

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

6.30.4.7 `min_lat`

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

6.30.4.8 min_ob_time

skdaccess.planetary.ode.cache.DataFetcher.min_ob_time

6.30.4.9 mission

skdaccess.planetary.ode.cache.DataFetcher.mission

6.30.4.10 number_product_limit

skdaccess.planetary.ode.cache.DataFetcher.number_product_limit

6.30.4.11 product_id

skdaccess.planetary.ode.cache.DataFetcher.product_id

6.30.4.12 product_type

skdaccess.planetary.ode.cache.DataFetcher.product_type

6.30.4.13 remove_ndv

skdaccess.planetary.ode.cache.DataFetcher.remove_ndv

6.30.4.14 result_offset_number

skdaccess.planetary.ode.cache.DataFetcher.result_offset_number

6.30.4.15 target

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

6.30.4.16 verbose

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

6.30.4.17 western_lon

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

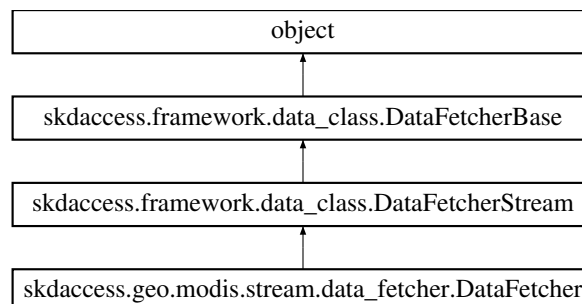
The documentation for this class was generated from the following file:

- [planetary/ode/cache/data_fetcher.py](#)

6.31 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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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`

6.31.1 Detailed Description

Data Fetcher for MODIS data.

6.31.2 Constructor & Destructor Documentation

6.31.2.1 `__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

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

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 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.31.3.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.31.3.4 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.31.3.5 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

6.31.3.6 output()

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

Generate data wrapper.

Returns

data wrapper of MODIS data

6.31.3.7 perturb()

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

Perturb parameters.

6.31.3.8 reset()

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

Set all parameters to initial value.

6.31.3.9 retrieveOnlineData()

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

Method for downloading data into memory.

Parameters

<i>data_specification</i>	Url list of data to be retrieved
---------------------------	----------------------------------

Returns

Retrieved data

6.31.3.10 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.31.3.11 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.31.3.12 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
```

```
key,  
value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.31.4 Member Data Documentation

6.31.4.1 ap_paramList

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

6.31.4.2 daynightboth

```
skdaccess.geo.modis.stream.DataFetcher.daynightboth
```

6.31.4.3 end_date

```
skdaccess.geo.modis.stream.DataFetcher.end_date
```

6.31.4.4 grid

```
skdaccess.geo.modis.stream.DataFetcher.grid
```


6.31.4.5 grid_fill

skdaccess.geo.modis.stream.DataFetcher.grid_fill

6.31.4.6 modis_id

skdaccess.geo.modis.stream.DataFetcher.modis_id

6.31.4.7 modis_identifier

skdaccess.geo.modis.stream.DataFetcher.modis_identifier

6.31.4.8 modis_platform

skdaccess.geo.modis.stream.DataFetcher.modis_platform

6.31.4.9 start_date

skdaccess.geo.modis.stream.DataFetcher.start_date

6.31.4.10 use_long_name

skdaccess.geo.modis.stream.DataFetcher.use_long_name

6.31.4.11 variable_list

skdaccess.geo.modis.stream.DataFetcher.variable_list

6.31.4.12 verbose

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

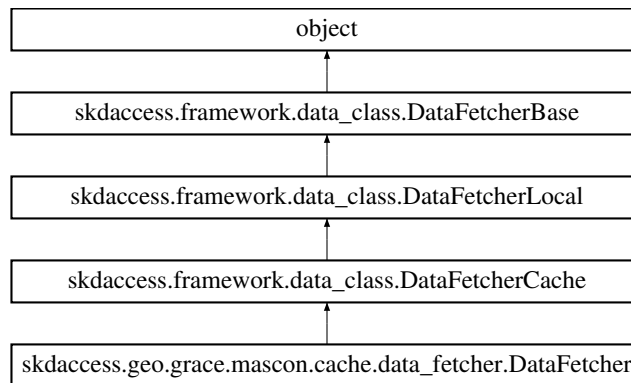
The documentation for this class was generated from the following file:

- [geo/modis/stream/data_fetcher.py](#)

6.32 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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.32.1 Detailed Description

Data Fetcher for GRACE mascon data.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 `__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

<i>ap_paramList[geo_point]</i>	AutoList of geographic location tuples (lat,lon)
<i>start_date</i>	Beginning date
<i>end_date</i>	Ending date

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

<i>keyname</i>	Name of dataset in configuration file
<i>online_path_list</i>	List of urls to data
<i>username</i>	Username for accessing online resources
<i>password</i>	Password for accessing online resources
<i>authentication_url</i>	The url used for authentication (unused when use_requests=True)
<i>cookiejar</i>	The cookiejar that stores credentials (unused when use_requests=True)
<i>use_requests</i>	Use the requests library instead of the standard library for accessing resources
<i>use_progress_bar</i>	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

<i>in_file_name</i>	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 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.32.3.6 `getDataLocation()`

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.32.3.7 `getHDFStorage()`

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.32.3.8 getMasconPlacement()

```
def skdaccess.geo.grace.mascon.cache.DataFetcher.getMasconPlacement (
    self )
```

Retrieve mascon placement data.

Returns

Mascon data, Mascon metadata

6.32.3.9 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.32.3.10 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

6.32.3.11 output()

```
def skdaccess.geo.grace.mascon.cache.DataFetcher.output (
    self )
```

Create a datawrapper containing GRACE mascon data.

Returns

Table Datawrapper containing Mascon GRACE data

6.32.3.12 perturb()

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

Perturb parameters.

6.32.3.13 reset()

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

Set all parameters to initial value.

6.32.3.14 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.32.3.15 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.32.3.16 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.32.3.17 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.32.4 Member Data Documentation

6.32.4.1 `ap_paramList`

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

6.32.4.2 `end_date`

`skdaccess.geo.grace.mascon.cache.DataFetcher.end_date`

6.32.4.3 `mascon_placement_url`

`skdaccess.geo.grace.mascon.cache.DataFetcher.mascon_placement_url`

6.32.4.4 `mascon_url`

`skdaccess.geo.grace.mascon.cache.DataFetcher.mascon_url`

6.32.4.5 `scale_factor_url`

`skdaccess.geo.grace.mascon.cache.DataFetcher.scale_factor_url`

6.32.4.6 `start_date`

`skdaccess.geo.grace.mascon.cache.DataFetcher.start_date`

6.32.4.7 `verbose`

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

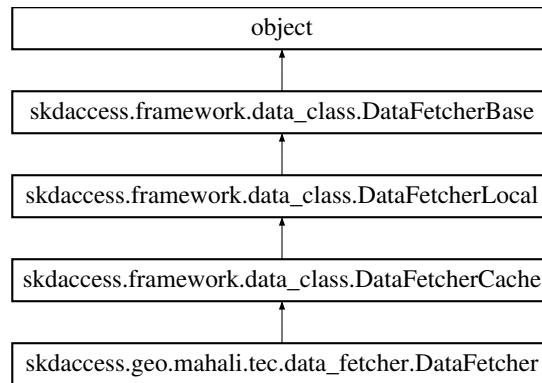
The documentation for this class was generated from the following file:

- `geo/grace/mascon/cache/data_fetcher.py`

6.33 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 `getHDFSStorage` (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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.33.1 Detailed Description

Data Fetcher for Mahali Data.

6.33.2 Constructor & Destructor Documentation

6.33.2.1 `__init__()`

```
def skdaccess.geo.mahali.tec.DataFetcher.__init__ (
    self,
    ap_paramList = [],
    start_date = None,
    end_date = None )
```

Initialize Mahali Data Fetcher.

Parameters

<i>ap_paramList</i> [stations]	Autolist of stations (Defaults to all stations)
<i>start_date</i>	Starting date for seelcting data (Defaults to beginning of available data)
<i>end_date</i>	Ending date for selecting data (Defaults to end of available data)

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()

```
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

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

Returns

List of downloaded file locations

6.33.3.3 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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.33.3.4 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.33.3.5 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.33.3.6 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.33.3.7 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.33.3.8 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.33.3.9 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

6.33.3.10 output()

```
def skdaccess.geo.mahali.tec.DataFetcher.output (
    self )
```

Generate data wrapper for Mahali tec data.

Returns

Mahali data wrapper

6.33.3.11 perturb()

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

Perturb parameters.

6.33.3.12 reset()

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

Set all parameters to initial value.

6.33.3.13 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.33.3.14 `verbose_print()`

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.33.3.15 `writeConfig()`

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.33.3.16 `writeConfigItem()`

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.33.4 Member Data Documentation**6.33.4.1 `ap_paramList`**

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

6.33.4.2 `date_range`

`skdaccess.geo.mahali.tec.DataFetcher.date_range`

6.33.4.3 `end_date`

`skdaccess.geo.mahali.tec.DataFetcher.end_date`

6.33.4.4 `start_date`

`skdaccess.geo.mahali.tec.DataFetcher.start_date`

6.33.4.5 `verbose`

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

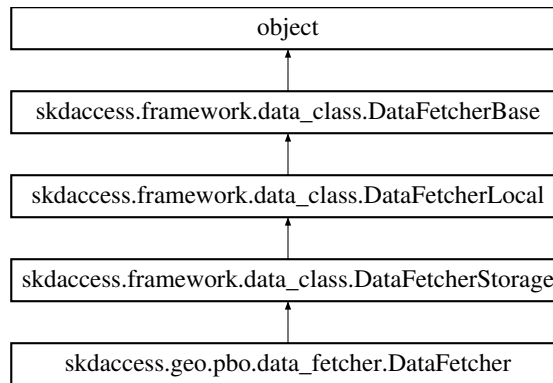
The documentation for this class was generated from the following file:

- `geo/mahali/tec/data_fetcher.py`

6.34 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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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

<i>start_time</i>	String of starting date in the form of "2005-01-01"
<i>end_time</i>	String of ending date in the form of "2014-12-31"
<i>ap_paramList[lat_range]</i>	AutoList, Latitude range used to select stabilization sites
<i>ap_paramList[lon_range]</i>	AutoList, Longitude range used to select stabilization sites
<i>mdyratio</i>	Only keep stations that have mdyratio of data in the specified time range
<i>default_columns</i>	Default columns to process
<i>default_error_columns</i>	Default error columns to process
<i>use_progress_bar</i>	Use a progress bar when loading data
<i>index_date_only</i>	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

<i>out_file</i>	Output filename for parsed data
<i>use_file</i>	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 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.34.3.6 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.34.3.7 getInfo()

```
def skdaccess.geo.pbo.DataFetcher.getInfo (
    self )
```

Get information about the stations and geo_point.

Returns

tuple containing station list and geo_point

6.34.3.8 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.34.3.9 getStationMetadata()

```
def skdaccess.geo.pbo.DataFetcher.getStationMetadata (
    data_frame = False )
```

Read in the metadata and convert to dictionary.

Returns

dictionary of PBO metadata

6.34.3.10 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

6.34.3.11 output()

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

Generate PBO Data Wrapper.

Returns

PBO Data Wrapper

6.34.3.12 perturb()

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

Perturb parameters.

6.34.3.13 reset()

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

Set all parameters to initial value.

6.34.3.14 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.34.3.15 setStationList()

```
def skdaccess.geo.pbo.DataFetcher.setStationList (
    self,
    station_list )
```

Set the list of stations to use.

Parameters

<i>station_list</i>	List of stations to fetch
---------------------	---------------------------

6.34.3.16 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
    self,
```

```

    args,
    kwargs ) [inherited]

```

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.34.3.17 writeConfig()

```

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

```

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.34.3.18 writeConfigItem()

```

def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]

```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

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]
```

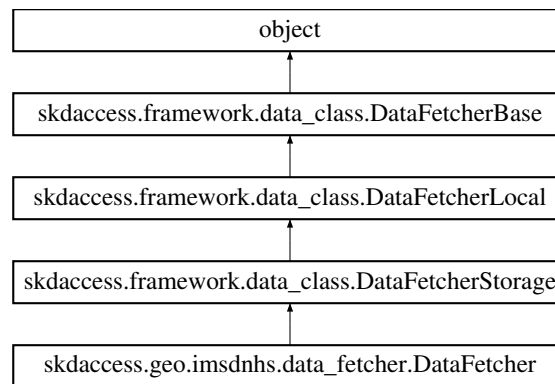
The documentation for this class was generated from the following file:

- [geo/pbo/data_fetcher.py](#)

6.35 skdaccess.geo.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)`
Initializes 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 `getConfigItem` (section, key)
- *Retrieve skdaccess configuration item.*
- def `writeConfigItem` (section, key, value)
- *Retrieve skdaccess configuration item.*
- 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`

6.35.1 Detailed Description

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

6.35.2 Constructor & Destructor Documentation

6.35.2.1 `__init__()`

```
def skdaccess.geo.imsdnhs.DataFetcher.__init__ (
    self,
    coordinate_dict,
    start_date,
    end_date )
```

Initializes the Data Fetcher.

Parameters

<i>coordinate_dict</i>	Dictionary of locations where the names are the keys and the items are lists containing the latitude and longitude are the values
<i>start_date</i>	Starting date
<i>end_date</i>	Ending date

6.35.3 Member Function Documentation

6.35.3.1 __str__()

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

Generate string description.

6.35.3.2 downloadFullDataset()

```
def skdaccess.framework.data_class.DataFetcherStorage.downloadFullDataset (
    cls,
    out_file,
    use_file = None ) [inherited]
```

Abstract function used to download full data set.

Parameters

<i>out_file</i>	output file name
<i>use_file</i>	Use previously downloaded data

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 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.35.3.5 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.35.3.6 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.35.3.7 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

6.35.3.8 output()

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

Fetch snow coverage data for coordinates.

Returns

Data wrapper for snow coverage

6.35.3.9 perturb()

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

Perturb parameters.

6.35.3.10 reset()

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

Set all parameters to initial value.

6.35.3.11 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.35.3.12 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.35.3.13 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.35.3.14 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.35.4 Member Data Documentation**6.35.4.1 ap_paramList**

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

6.35.4.2 coordinate_dict

`skdaccess.geo.imsdnhs.DataFetcher.coordinate_dict`

6.35.4.3 end_date

`skdaccess.geo.imsdnhs.DataFetcher.end_date`

6.35.4.4 start_date

`skdaccess.geo.imsdnhs.DataFetcher.start_date`

6.35.4.5 verbose

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

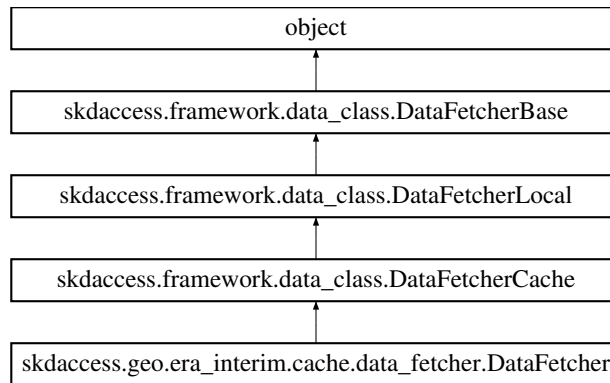
The documentation for this class was generated from the following file:

- [geo/imsdnhs/data_fetcher.py](#)

6.36 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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.36.1 Detailed Description

[DataFetcher](#) for retrieving ERA-I data.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 `__init__()`

```
def skdaccess.geo.era_interim.cache.DataFetcher.__init__ (
    self,
    date_list,
    data_names,
    username,
    password )
```

Initialize Data Fetcher.

Parameters

<i>date_list</i>	list of dates
<i>data_names</i>	list of data names
<i>username</i>	UCAR username
<i>password</i>	UCAR password

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()`

```
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

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

Returns

List of downloaded file locations

6.36.3.3 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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.36.3.4 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.36.3.5 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.36.3.6 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	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

<i>keyname</i>	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.era_interim.cache.DataFetcher.output (
    self )
```

Generate data wrapper.

Returns

Era-I weather in a 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

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	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

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	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

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.36.3.16 `writeConfigItem()`

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.36.4 Member Data Documentation

6.36.4.1 ap_paramList

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

6.36.4.2 data_names

`skdaccess.geo.era_interim.cache.DataFetcher.data_names`

6.36.4.3 date_list

`skdaccess.geo.era_interim.cache.DataFetcher.date_list`

6.36.4.4 password

`skdaccess.geo.era_interim.cache.DataFetcher.password`

6.36.4.5 username

`skdaccess.geo.era_interim.cache.DataFetcher.username`

6.36.4.6 verbose

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

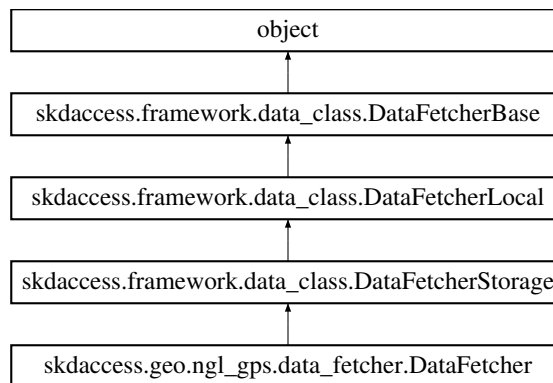
The documentation for this class was generated from the following file:

- [geo/era_interim/cache/data_fetcher.py](#)

6.37 skdaccess.geo.ngl_gps.DataFetcher Class Reference

Data fetcher for GPS data from Nevada 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')`
Construct 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 [getConfigItem](#) (section, key)
- Retrieve skdaccess configuration item.*
 - def [writeConfigItem](#) (section, key, value)
- Retrieve skdaccess configuration item.*
 - 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](#)

6.37.1 Detailed Description

Data fetcher for GPS data from Nevada Geodetic Laboratory.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 [__init__\(\)](#)

```
def skdaccess.geo.ngl_gps.DataFetcher.__init__ (
    self,
    start_date,
    end_date,
    lat_range,
    lon_range,
    mdyratio = 0.7,
    data_type = 'ngl_gps' )
```

Construct NGL data fetcher.

Parameters

<i>start_date</i>	Starting date (string: '2002-01-01')
<i>end_date</i>	Ending date (string: '2015-01-01')
<i>lat_range</i>	Tuple containing latitude range
<i>lon_range</i>	Tuple containing longitude range
<i>mdyratio</i>	Choose stations whose ratio of valid/total is greater than mdyratio
<i>data_type</i>	Either 24 hour product (' ngl_gps ') or 5 minute product ('ngl_5min')

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 `downloadFullDataset()`

```
def skdaccess.framework.data_class.DataFetcherStorage.downloadFullDataset (
    cls,
    out_file,
    use_file = None ) [inherited]
```

Abstract function used to download full data set.

Parameters

<i>out_file</i>	output file name
<i>use_file</i>	Use previously downloaded data

Returns

Absolute path of parsed data

6.37.3.3 getAntennaLogs()

```
def skdaccess.geo.ngl_gps.DataFetcher.getAntennaLogs ( )
```

Retrieve information about antenna changes.

Returns

dictionary of antenna changes

6.37.3.4 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.37.3.5 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.37.3.6 `getDataLocation()`

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.37.3.7 `getMetadata()`

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.37.3.8 `getStationMetadata()`

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

Get station metadata.

Returns

data frame of station metadata

6.37.3.9 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

6.37.3.10 output()

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

Construct NGL GPS data wrapper.

Returns

NGL GPS data wrapper

6.37.3.11 perturb()

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

Perturb parameters.

6.37.3.12 reset()

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

Set all parameters to initial value.

6.37.3.13 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.37.3.14 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.37.3.15 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.37.3.16 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.37.4 Member Data Documentation

6.37.4.1 ap_paramList

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

6.37.4.2 data_type

`skdaccess.geo.ngl_gps.DataFetcher.data_type`

6.37.4.3 end_date

`skdaccess.geo.ngl_gps.DataFetcher.end_date`

6.37.4.4 lat_range

`skdaccess.geo.ngl_gps.DataFetcher.lat_range`

6.37.4.5 lon_range

`skdaccess.geo.ngl_gps.DataFetcher.lon_range`

6.37.4.6 mdyratio

`skdaccess.geo.ngl_gps.DataFetcher.mdyratio`

6.37.4.7 start_date

`skdaccess.geo.ngl_gps.DataFetcher.start_date`

6.37.4.8 verbose

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

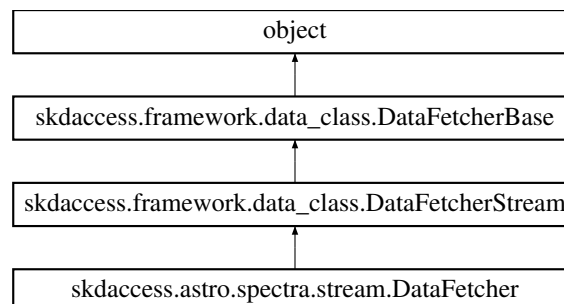
The documentation for this class was generated from the following file:

- [geo/ngl_gps/data_fetcher.py](#)

6.38 skdaccess.astro.spectra.stream.DataFetcher Class Reference

Data Fetcher for Sloan Digital Sky Survey spectra.

Inheritance diagram for `skdaccess.astro.spectra.stream.DataFetcher`:



Public Member Functions

- def `__init__` (self, `ap_paramList`)
Initialize SDSS spectra Data Fetcher.
- 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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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`

6.38.1 Detailed Description

Data Fetcher for Sloan Digital Sky Survey spectra.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 `__init__()`

```
def skdaccess.astro.spectra.stream.DataFetcher.__init__ (
    self,
    ap_paramList )
```

Initialize SDSS spectra Data Fetcher.

Parameters

<i>ap_paramList[url_list]</i>	Autolist of URLs to access
-------------------------------	----------------------------

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 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.38.3.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.38.3.4 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.38.3.5 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

6.38.3.6 output()

```
def skdaccess.astro.spectra.stream.DataFetcher.output (
    self )
```

Generate data wrapper.

Returns

Table wrapper of SDSS spectra data

6.38.3.7 perturb()

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

Perturb parameters.

6.38.3.8 reset()

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

Set all parameters to initial value.

6.38.3.9 retrieveOnlineData()

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

Method for downloading data into memory.

Parameters

<i>data_specification</i>	Url list of data to be retrieved
---------------------------	----------------------------------

Returns

Retrieved data

6.38.3.10 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.38.3.11 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.38.3.12 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.38.4 Member Data Documentation

6.38.4.1 `ap_paramList`

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

6.38.4.2 `verbose`

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

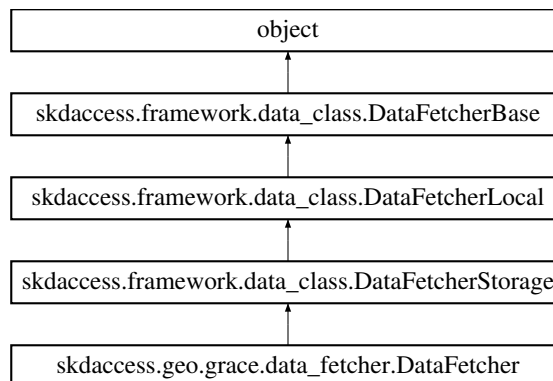
The documentation for this class was generated from the following file:

- [astro/spectra/stream.py](#)

6.39 `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 geopotents.
- `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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.39.1 Detailed Description

Data Fetcher for GRACE data.

6.39.2 Constructor & Destructor Documentation

6.39.2.1 `__init__()`

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

Construct a Grace Data Fetcher.

Parameters

<i>ap_paramList[geo_point]</i>	AutoList of geographic location tuples (lat,lon)
<i>start_date</i>	Beginning date
<i>end_date</i>	Ending date

6.39.3 Member Function Documentation

6.39.3.1 __str__()

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

String representation of data fetcher.

Returns

String listing the name and geopoint of data fetcher

6.39.3.2 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

<i>out_file</i>	Output filename for parsed data
<i>use_file</i>	Directory of already downloaded data. If None, data will be downloaded.

Returns

Absolute path of parsed data

6.39.3.3 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.39.3.4 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.39.3.5 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.39.3.6 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.39.3.7 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

6.39.3.8 output()

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

Create data wrapper of grace data for specified geopoints.

Returns

Grace Data Wrapper

6.39.3.9 perturb()

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

Perturb parameters.

6.39.3.10 reset()

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

Set all parameters to initial value.

6.39.3.11 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.39.3.12 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.39.3.13 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.39.3.14 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.39.4 Member Data Documentation

6.39.4.1 ap_paramList

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

6.39.4.2 end_date

```
skdaccess.geo.grace.DataFetcher.end_date
```


6.39.4.3 start_date

skdaccess.geo.grace.DataFetcher.start_date

6.39.4.4 verbose

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

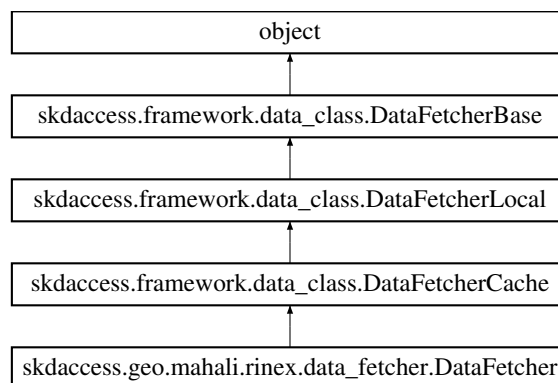
The documentation for this class was generated from the following file:

- [geo/grace/data_fetcher.py](#)

6.40 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 [getHDFSStorage](#) (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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.40.1 Detailed Description

Data Fetcher for Mahali Data.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 `__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

<i>ap_paramList[stations]</i>	Autolist of stations (Defaults to all stations)
<i>start_date</i>	Starting date for seelcting data (Defaults to beginning of available data)
<i>end_date</i>	Ending date for selecting data (Defaults to end of available data)
<i>generate_links</i>	Generate links to data instead of downloading data

6.40.3 Member Function Documentation

6.40.3.1 `__str__()`

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

Generate string description.

6.40.3.2 `cacheData()` [1/2]

```
def skdaccess.geo.mahali.rinex.DataFetcher.cacheData (
    self )
```

Downloads all needed data.

Called by [output\(\)](#).

6.40.3.3 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

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

Returns

List of downloaded file locations

6.40.3.4 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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.40.3.5 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.40.3.6 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.40.3.7 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.40.3.8 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.40.3.9 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.40.3.10 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

6.40.3.11 output()

```
def skdaccess.geo.mahali.rinex.DataFetcher.output (
    self )
```

Generate data wrapper for Mahali data.

Returns

Mahali data wrapper

6.40.3.12 perturb()

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

Perturb parameters.

6.40.3.13 reset()

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

Set all parameters to initial value.

6.40.3.14 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.40.3.15 `verbose_print()`

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.40.3.16 `writeConfig()`

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.40.3.17 `writeConfigItem()`

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.40.4 Member Data Documentation

6.40.4.1 ap_paramList

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

6.40.4.2 date_range

`skdaccess.geo.mahali.rinex.DataFetcher.date_range`

6.40.4.3 end_date

`skdaccess.geo.mahali.rinex.DataFetcher.end_date`

6.40.4.4 generate_links

`skdaccess.geo.mahali.rinex.DataFetcher.generate_links`

6.40.4.5 start_date

`skdaccess.geo.mahali.rinex.DataFetcher.start_date`

6.40.4.6 verbose

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

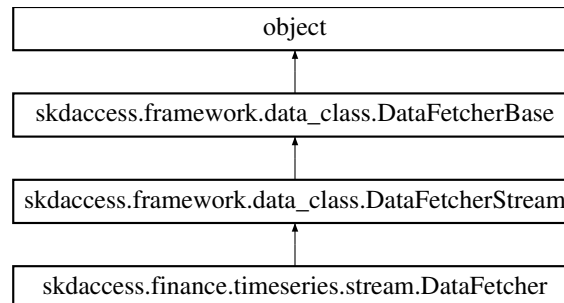
The documentation for this class was generated from the following file:

- [geo/mahali/rinex/data_fetcher.py](#)

6.41 skdaccess.finance.timeseries.stream.DataFetcher Class Reference

Data Fetcher for retrieving stock data.

Inheritance diagram for `skdaccess.finance.timeseries.stream.DataFetcher`:



Public Member Functions

- `def __init__(self, ap_paramList, data_type, start_date=None, end_date=None, interval=None)`
- `def output(self)`
Retrieve stock data.
- `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 getConfigItem(section, key)`
Retrieve skdaccess configuration item.
- `def writeConfigItem(section, key, value)`
Retrieve skdaccess configuration item.
- `def writeConfig(conf)`
Write config to disk.
- `def verbose_print(self, args, kwargs)`
Print statement if verbose flag is set.

Public Attributes

- [data_type](#)
- [start_date](#)
- [end_date](#)
- [interval](#)
- [possible_intervals](#)
- [possible_data_types](#)
- [ap_paramList](#)
- [verbose](#)

6.41.1 Detailed Description

Data Fetcher for retrieving stock data.

6.41.2 Constructor & Destructor Documentation

6.41.2.1 `__init__()`

```
def skdaccess.finance.timeseries.stream.DataFetcher.__init__ (
    self,
    ap_paramList,
    data_type,
    start_date = None,
    end_date = None,
    interval = None )
```

Parameters

<i>ap_paramList</i> [<i>stock_symbol_list</i>]	AutoList of stock symbols
<i>data_type</i>	Type of data to retrieve (daily, daily_adjusted, intraday, monthly, monthly_adjusted, weekly, weekly_adjusted)
<i>start_date</i>	Starting date
<i>end_date</i>	Ending date
<i>interval</i>	Interval for intraday (1min, 5min, 15min, 30min, 60min)

Returns

: Table data wrapper of stock data

6.41.3 Member Function Documentation

6.41.3.1 `__str__()`

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

Generate string description.

6.41.3.2 `getConfig()`

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.41.3.3 `getConfigItem()`

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.41.3.4 `getMetadata()`

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.41.3.5 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

6.41.3.6 output()

```
def skdaccess.finance.timeseries.stream.DataFetcher.output (
    self )
```

Retrieve stock data.

Returns

TableWrapper of stock data

6.41.3.7 perturb()

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

Perturb parameters.

6.41.3.8 reset()

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

Set all parameters to initial value.

6.41.3.9 retrieveOnlineData()

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

Method for downloading data into memory.

Parameters

<i>data_specification</i>	Url list of data to be retrieved
---------------------------	----------------------------------

Returns

Retrieved data

6.41.3.10 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.41.3.11 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.41.3.12 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
```

```
key,  
value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.41.4 Member Data Documentation

6.41.4.1 ap_paramList

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

6.41.4.2 data_type

```
skdaccess.finance.timeseries.stream.DataFetcher.data_type
```

6.41.4.3 end_date

```
skdaccess.finance.timeseries.stream.DataFetcher.end_date
```

6.41.4.4 interval

```
skdaccess.finance.timeseries.stream.DataFetcher.interval
```

6.41.4.5 possible_data_types

`skdaccess.finance.timeseries.stream.DataFetcher.possible_data_types`

6.41.4.6 possible_intervals

`skdaccess.finance.timeseries.stream.DataFetcher.possible_intervals`

6.41.4.7 start_date

`skdaccess.finance.timeseries.stream.DataFetcher.start_date`

6.41.4.8 verbose

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

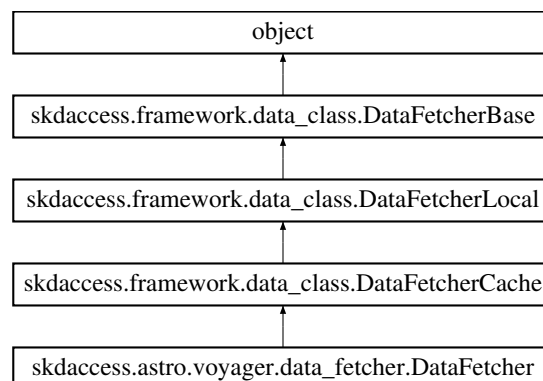
The documentation for this class was generated from the following file:

- [finance/timeseries/stream.py](#)

6.42 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 `getHDFSStorage` (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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.42.1 Detailed Description

Data Fetcher for Mahali temperature data.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 `__init__()`

```
def skdaccess.astro.voyager.DataFetcher.__init__ (
    self,
    start_year,
    end_year,
    spacecraft = 'both' )
```

Initialize Voyager data fetcher.

Parameters

<i>start_year</i>	Starting year
<i>end_year</i>	Ending year
<i>spacecraft</i>	Which spacecraft to use (voyager1, voyager2, or both).

6.42.3 Member Function Documentation

6.42.3.1 `__str__()`

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

Generate string description.

6.42.3.2 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

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

Returns

List of downloaded file locations

6.42.3.3 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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.42.3.4 generateURL()

```
def skdaccess.astro.voyager.DataFetcher.generateURL (
    self,
    spacecraft,
    in_year )
```

Generate url for voyager data.

Parameters

<i>spacecraft</i>	Voyager spacecraft (vy1 or vy2)
<i>in_year</i>	Input year (or 'metadata')

Returns

Url of data location

6.42.3.5 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.42.3.6 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.42.3.7 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.42.3.8 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.42.3.9 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.42.3.10 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

6.42.3.11 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

6.42.3.12 output()

```
def skdaccess.astro.voyager.DataFetcher.output (
    self )
```

Generate data wrapper.

Returns

data wrapper of voyager data

6.42.3.13 parseVoyagerData()

```
def skdaccess.astro.voyager.DataFetcher.parseVoyagerData (
    self,
    spacecraft,
    in_filename )
```

Parse Voyager Data.

Parameters

<i>spacecraft</i>	Voyager spacecraft (vy1 or vy2)
<i>in_filename</i>	Input voyager data filename

Returns

Pandas Dataframe of Voyager data

6.42.3.14 parseVoyagerMetadata()

```
def skdaccess.astro.voyager.DataFetcher.parseVoyagerMetadata (
    self,
    in_file )
```

Parse voyager metadata.

Parameters

<i>in_file</i>	Input filename
----------------	----------------

Returns

Dictionary containing metadata

6.42.3.15 perturb()

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

Perturb parameters.

6.42.3.16 reset()

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

Set all parameters to initial value.

6.42.3.17 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.42.3.18 verbose_print()

```
def skdaccess.framework.data_class.DataFetcherBase.verbose_print (
    self,
```



```

    args,
    kwargs ) [inherited]

```

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.42.3.19 writeConfig()

```

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

```

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.42.3.20 writeConfigItem()

```

def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]

```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.42.4 Member Data Documentation

6.42.4.1 `ap_paramList`

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

6.42.4.2 `base_url`

`skdaccess.astro.voyager.DataFetcher.base_url`

6.42.4.3 `field_names`

`skdaccess.astro.voyager.DataFetcher.field_names`

6.42.4.4 `field_widths`

`skdaccess.astro.voyager.DataFetcher.field_widths`

6.42.4.5 `spacecraft_list`

`skdaccess.astro.voyager.DataFetcher.spacecraft_list`

6.42.4.6 `verbose`

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

6.42.4.7 `year_list`

`skdaccess.astro.voyager.DataFetcher.year_list`

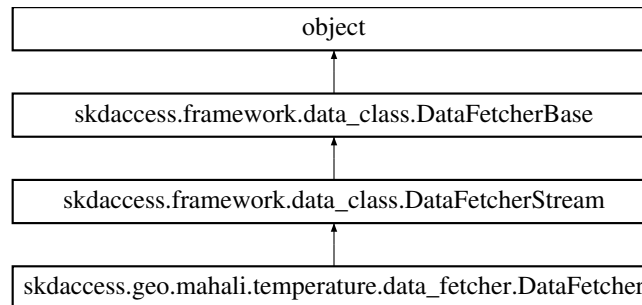
The documentation for this class was generated from the following file:

- `astro/voyager/data_fetcher.py`

6.43 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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.43.1 Detailed Description

Data Fetcher for Mahali temperature data.

6.43.2 Constructor & Destructor Documentation

6.43.2.1 `__init__()`

```
def skdaccess.geo.mahali.temperature.DataFetcher.__init__ (
    self,
    ap_paramList = [],
    start_date = None,
    end_date = None )
```

Initialize Mahali temperature data fetcher.

Parameters

<i>ap_paramList</i> [stations]	Autolist of stations (Defaults to all stations)
<i>start_date</i>	Starting date for seelcting data (Defaults to beginning of available data)
<i>end_date</i>	Ending date for selecting data (Defaults to end of available data)

6.43.3 Member Function Documentation

6.43.3.1 `__str__()`

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

Generate string description.

6.43.3.2 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.43.3.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.43.3.4 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.43.3.5 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

6.43.3.6 output()

```
def skdaccess.geo.mahali.temperature.DataFetcher.output (
    self )
```

Generate data wrapper for Mahali temperatures.

Returns

Mahali temperature data wrapper

6.43.3.7 perturb()

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

Perturb parameters.

6.43.3.8 reset()

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

Set all parameters to initial value.

6.43.3.9 retrieveOnlineData()

```
def skdaccess.geo.mahali.temperature.DataFetcher.retrieveOnlineData (
    self,
    data_specification )
```

Load data in from a remote source.

Parameters

<i>data_specification</i>	Pandas dataframe containing the columns 'station', 'date', and 'filename'
---------------------------	---

Returns

Ordered dictionary for each station (key) which contains a pandas data frame of the temperature

6.43.3.10 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.43.3.11 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.43.3.12 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
```

```
key,  
value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.43.4 Member Data Documentation

6.43.4.1 ap_paramList

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

6.43.4.2 end_date

```
skdaccess.geo.mahali.temperature.DataFetcher.end_date
```

6.43.4.3 start_date

```
skdaccess.geo.mahali.temperature.DataFetcher.start_date
```

6.43.4.4 verbose

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

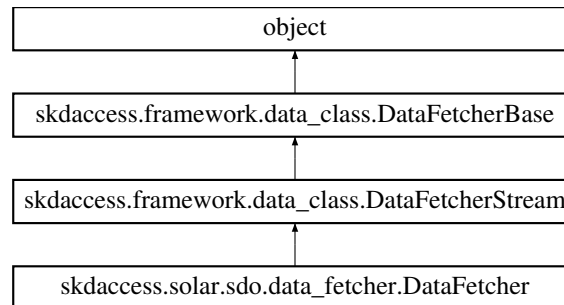
The documentation for this class was generated from the following file:

- geo/mahali/temperature/[data_fetcher.py](#)

6.44 skdaccess.solar.sdo.DataFetcher Class Reference

Data Fetcher for the Solar Dynamics Observatory.

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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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 for the Solar Dynamics Observatory.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 `__init__()`

```
def skdaccess.solar.sdo.DataFetcher.__init__ (
    self,
    ap_paramList )
```

Initialize Solar Dynamics Observatory.

Parameters

<code>ap_paramList[url_list]</code>	Autolist of URLs to access
-------------------------------------	----------------------------

6.44.3 Member Function Documentation

6.44.3.1 `__str__()`

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

Generate string description.

6.44.3.2 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.44.3.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.44.3.4 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.44.3.5 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

6.44.3.6 output()

```
def skdaccess.solar.sdo.DataFetcher.output (
    self )
```

Generate data wrapper.

Returns

data wrapper of SDO data

6.44.3.7 perturb()

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

Perturb parameters.

6.44.3.8 reset()

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

Set all parameters to initial value.

6.44.3.9 retrieveOnlineData()

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

Method for downloading data into memory.

Parameters

<i>data_specification</i>	Url list of data to be retrieved
---------------------------	----------------------------------

Returns

Retrieved data

6.44.3.10 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.44.3.11 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.44.3.12 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
```

```

    key,
    value ) [inherited]

```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.44.4 Member Data Documentation

6.44.4.1 ap_paramList

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

6.44.4.2 verbose

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

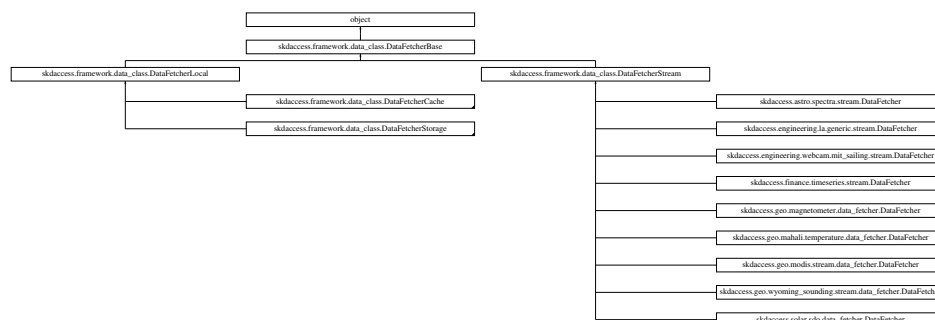
The documentation for this class was generated from the following file:

- [solar/sdo/data_fetcher.py](#)

6.45 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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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`

6.45.1 Detailed Description

Base class for all data fetchers.

6.45.2 Constructor & Destructor Documentation

6.45.2.1 `__init__`()

```
def skdaccess.framework.data_class.DataFetcherBase.__init__ (
    self,
    ap_paramList = [],
    verbose = False )
```

Initialize data fetcher with parameter list.

Parameters

<i>ap_paramList</i>	List of parameters
<i>verbose</i>	Output extra information

6.45.3 Member Function Documentation**6.45.3.1 __str__()**

```
def skdaccess.framework.data_class.DataFetcherBase.__str__ (
    self )
```

Generate string description.

6.45.3.2 getConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfig ( )
```

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.45.3.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key )
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.45.3.4 getMetadata()

```
def skdaccess.framework.data_class.DataFetcherBase.getMetadata (
    self )
```

Return metadata about Data Fetcher.

Returns

metadata of object.

6.45.3.5 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

6.45.3.6 output()

```
def skdaccess.framework.data_class.DataFetcherBase.output (
    self )
```

Output data wrapper.

Returns

Datawrapper

6.45.3.7 perturb()

```
def skdaccess.framework.data_class.DataFetcherBase.perturb (
    self )
```

Perturb parameters.

6.45.3.8 reset()

```
def skdaccess.framework.data_class.DataFetcherBase.reset (
    self )
```

Set all parameters to initial value.

6.45.3.9 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.45.3.10 writeConfig()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfig (
    conf )
```

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.45.3.11 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value )
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.45.4 Member Data Documentation

6.45.4.1 ap_paramList

```
skdaccess.framework.data_class.DataFetcherBase.ap_paramList
```

6.45.4.2 verbose

```
skdaccess.framework.data_class.DataFetcherBase.verbose
```

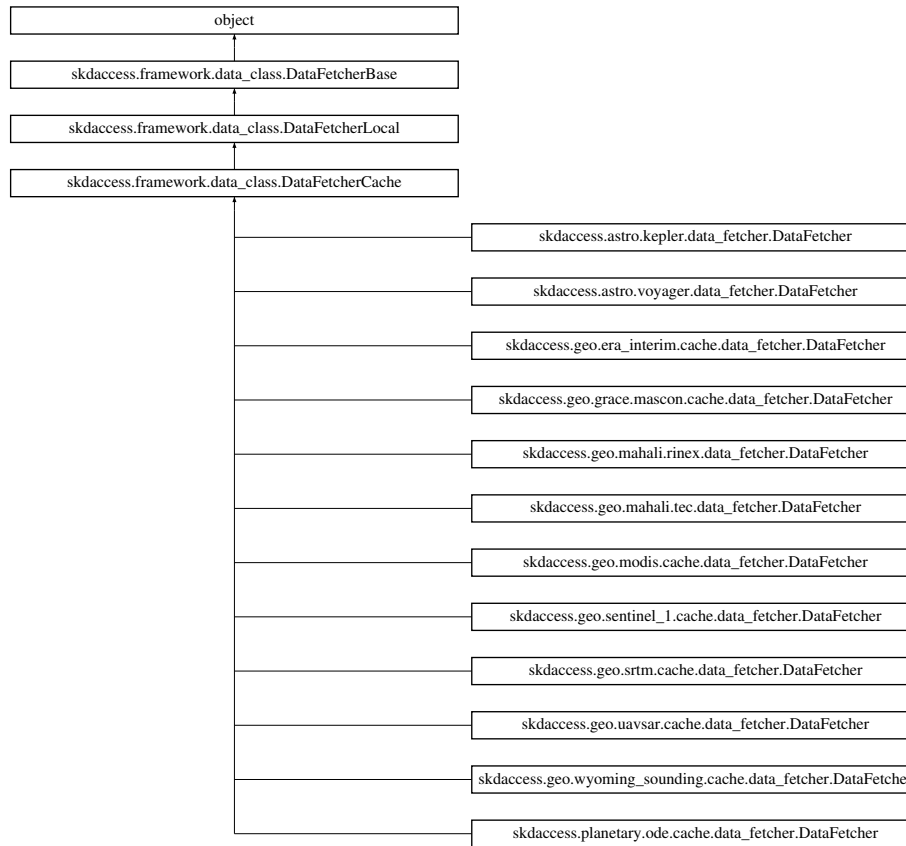
The documentation for this class was generated from the following file:

- framework/[data_class.py](#)

6.46 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 `getConfigItem` (section, key)
Retrieve skdaccess configuration item.
- def `writeConfigItem` (section, key, value)
Retrieve skdaccess configuration item.
- 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`

6.46.1 Detailed Description

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

6.46.2 Member Function Documentation

6.46.2.1 `__str__()`

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

Generate string description.

6.46.2.2 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

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

Returns

List of downloaded file locations

6.46.2.3 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

<i>in_file_name</i>	Input filename to test
---------------------	------------------------

Returns

True if data exists and False otherwise

6.46.2.4 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.46.2.5 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.46.2.6 getDataLocation()

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.46.2.7 getHDFStorage()

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

Retrieve a Pandas HDF Store for a dataset.

Parameters

<i>keyname</i>	Key name of HDF store
----------------	-----------------------

Returns

Pandas HDF Store

6.46.2.8 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.46.2.9 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

6.46.2.10 output()

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

Output data wrapper.

Returns

Datawrapper

6.46.2.11 perturb()

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

Perturb parameters.

6.46.2.12 reset()

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

Set all parameters to initial value.

6.46.2.13 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.46.2.14 `verbose_print()`

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.46.2.15 `writeConfig()`

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.46.2.16 `writeConfigItem()`

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.46.3 Member Data Documentation

6.46.3.1 ap_paramList

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

6.46.3.2 verbose

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

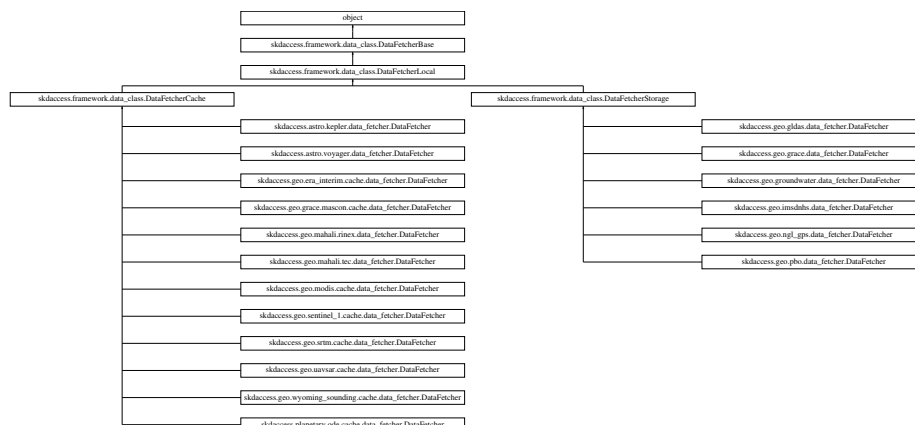
The documentation for this class was generated from the following file:

- [framework/data_class.py](#)

6.47 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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.47.1 Detailed Description

Data fetcher base class for use when storing data locally.

6.47.2 Member Function Documentation

6.47.2.1 __str__()

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

Generate string description.

6.47.2.2 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.47.2.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.47.2.4 getDataLocation()

```
def skdaccess.framework.data_class.DataFetcherLocal.getDataLocation (
    data_name )
```

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.47.2.5 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.47.2.6 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

6.47.2.7 output()

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

Output data wrapper.

Returns

Datawrapper

6.47.2.8 perturb()

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

Perturb parameters.

6.47.2.9 reset()

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

Set all parameters to initial value.

6.47.2.10 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.47.2.11 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.47.2.12 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.47.2.13 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.47.3 Member Data Documentation

6.47.3.1 ap_paramList

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

6.47.3.2 verbose

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

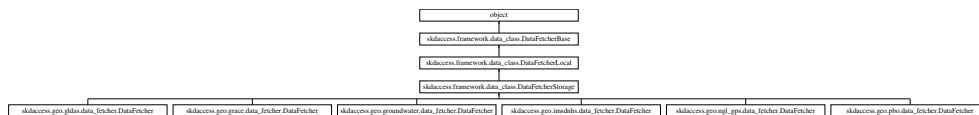
The documentation for this class was generated from the following file:

- framework/[data_class.py](#)

6.48 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 `getConfigItem (section, key)`
Retrieve skdaccess configuration item.
- def `writeConfigItem (section, key, value)`
Retrieve skdaccess configuration item.
- 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`

6.48.1 Detailed Description

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

6.48.2 Member Function Documentation

6.48.2.1 `__str__()`

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

Generate string description.

6.48.2.2 `downloadFullDataset()`

```
def skdaccess.framework.data_class.DataFetcherStorage.downloadFullDataset (
    cls,
    out_file,
    use_file = None )
```

Abstract function used to download full data set.

Parameters

<i>out_file</i>	output file name
<i>use_file</i>	Use previously downloaded data

Returns

Absolute path of parsed data

6.48.2.3 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.48.2.4 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.48.2.5 `getDataLocation()`

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

Get the location of data set.

Parameters

<i>data_name</i>	Name of data set
------------------	------------------

Returns

string of data location, None if not found

6.48.2.6 `getMetadata()`

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.48.2.7 `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

6.48.2.8 output()

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

Output data wrapper.

Returns

Datawrapper

6.48.2.9 perturb()

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

Perturb parameters.

6.48.2.10 reset()

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

Set all parameters to initial value.

6.48.2.11 setDataLocation()

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

Set the location of a data set.

Parameters

<i>data_name</i>	Name of data set
<i>location</i>	Location of data set
<i>key</i>	Key of configuration option

6.48.2.12 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.48.2.13 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.48.2.14 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
    key,
    value ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.48.3 Member Data Documentation

6.48.3.1 ap_paramList

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

6.48.3.2 verbose

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

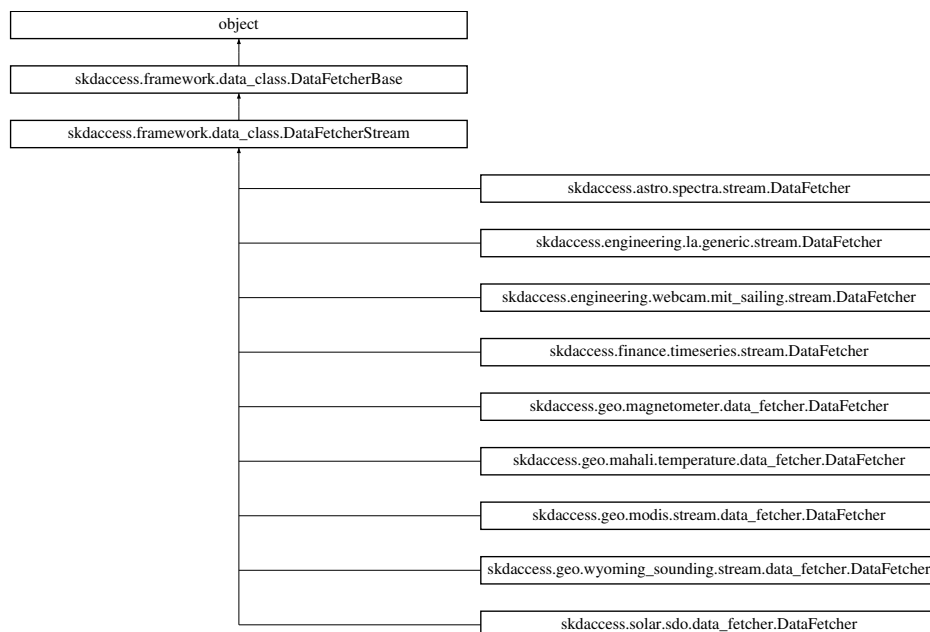
The documentation for this class was generated from the following file:

- framework/[data_class.py](#)

6.49 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 [getConfigItem](#) (section, key)
Retrieve skdaccess configuration item.
- def [writeConfigItem](#) (section, key, value)
Retrieve skdaccess configuration item.
- 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](#)

6.49.1 Detailed Description

Data fetcher base class for downloading data into memory.

6.49.2 Member Function Documentation

6.49.2.1 __str__()

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

Generate string description.

6.49.2.2 getConfig()

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

Retrieve skdaccess configuration.

Returns

configParser.ConfigParser object of configuration

6.49.2.3 getConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.getConfigItem (
    section,
    key ) [inherited]
```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.49.2.4 getMetadata()

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

Return metadata about Data Fetcher.

Returns

metadata of object.

6.49.2.5 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

6.49.2.6 output()

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

Output data wrapper.

Returns

Datawrapper

6.49.2.7 perturb()

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

Perturb parameters.

6.49.2.8 reset()

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

Set all parameters to initial value.

6.49.2.9 retrieveOnlineData()

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

Method for downloading data into memory.

Parameters

<i>data_specification</i>	Url list of data to be retrieved
---------------------------	----------------------------------

Returns

Retrieved data

6.49.2.10 verbose_print()

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

Print statement if verbose flag is set.

Parameters

<i>*args</i>	Arguments to pass to print
<i>**kwargs</i>	Keyword arguments to pass to print

6.49.2.11 writeConfig()

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

Write config to disk.

Parameters

<i>conf</i>	configparser.ConfigParser object
-------------	----------------------------------

6.49.2.12 writeConfigItem()

```
def skdaccess.framework.data_class.DataFetcherBase.writeConfigItem (
    section,
```

```

        key,
        value ) [inherited]

```

Retrieve skdaccess configuration item.

Parameters

<i>section</i>	Section of configuration item
<i>key</i>	Configuration key value

Returns

Requested configuration item or None if it doesn't exist

6.49.3 Member Data Documentation

6.49.3.1 ap_paramList

```

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

```

6.49.3.2 verbose

```

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

```

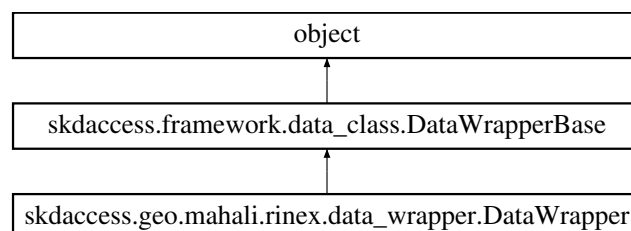
The documentation for this class was generated from the following file:

- framework/[data_class.py](#)

6.50 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](#)

6.50.1 Detailed Description

Data wrapper for Mahali data.

6.50.2 Member Function Documentation

6.50.2.1 `__len__()`

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
    self ) [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.50.2.2 `addResult()`

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
    self,
    rkey,
    rres ) [inherited]
```

Add a result to the data wrapper.

Parameters

<i>rkey</i>	Result key
<i>rres</i>	Result

6.50.2.3 `get()`

```
def skdaccess.framework.data_class.DataWrapperBase.get (
    self ) [inherited]
```

Retrieve stored data.

Returns

Stored data

6.50.2.4 getIterator()

```
def skdaccess.geo.mahali.rinex.data_wrapper.DataWrapper.getIterator (
    self )
```

Get iterator to Mahali data.

Returns

Iterator yielding (site,date,nav,obs)

6.50.2.5 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
    self ) [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.50.2.6 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
    self ) [inherited]
```

Get the Run ID.

Returns

run_id

6.50.2.7 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
    self,
    key = None ) [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.50.2.8 reset()

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

Reset data back to original state.

6.50.2.9 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
    self,
    obj ) [inherited]
```

Updated wrapped data.

Parameters

<i>obj</i>	New data for wrapper
------------	----------------------

6.50.2.10 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
    self,
    new_metadata ) [inherited]
```

Update metadata.

Parameters

<i>new_metadata</i>	New metadata
---------------------	--------------

6.50.3 Member Data Documentation

6.50.3.1 constants

```
skdaccess.framework.data_class.DataWrapperBase.constants [inherited]
```


6.50.3.2 data

`skdaccess.framework.data_class.DataWrapperBase.data` [inherited]

6.50.3.3 meta_data

`skdaccess.framework.data_class.DataWrapperBase.meta_data` [inherited]

6.50.3.4 results

`skdaccess.framework.data_class.DataWrapperBase.results` [inherited]

6.50.3.5 run_id

`skdaccess.framework.data_class.DataWrapperBase.run_id` [inherited]

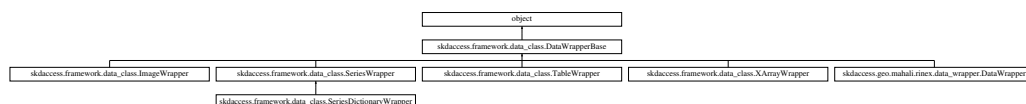
The documentation for this class was generated from the following file:

- [geo/mahali/rinex/data_wrapper.py](#)

6.51 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`

6.51.1 Detailed Description

Base class for wrapping data for use in DiscoveryPipeline.

6.51.2 Constructor & Destructor Documentation

6.51.2.1 `__init__()`

```
def skdaccess.framework.data_class.DataWrapperBase.__init__ (
    self,
    obj_wrap,
    run_id = -1,
    meta_data = None )
```

Construct wrapper from input data.

Parameters

<i>obj_wrap</i>	Data to be wrapped
<i>run_id</i>	ID of the run
<i>meta_data</i>	Metadata to store with data

6.51.3 Member Function Documentation

6.51.3.1 `__len__()`

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
    self )
```

Get length of wrapped data.

Returns

length of wrapped data

6.51.3.2 `addResult()`

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
    self,
    rkey,
    rres )
```

Add a result to the data wrapper.

Parameters

<i>rkey</i>	Result key
<i>rres</i>	Result

6.51.3.3 `get()`

```
def skdaccess.framework.data_class.DataWrapperBase.get (
    self )
```

Retrieve stored data.

Returns

Stored data

6.51.3.4 getIterator()

```
def skdaccess.framework.data_class.DataWrapperBase.getIterator (
    self )
```

Get an iterator to the data.

Returns

iterator to data

6.51.3.5 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
    self )
```

Retrieve accumulated results, if any.

Returns

store results

6.51.3.6 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
    self )
```

Get the Run ID.

Returns

run_id

6.51.3.7 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
    self,
    key = None )
```

Get information about data wrapper.

Returns

The stored metadata

6.51.3.8 reset()

```
def skdaccess.framework.data_class.DataWrapperBase.reset (
    self )
```

Reset data back to original state.

6.51.3.9 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
    self,
    obj )
```

Updated wrapped data.

Parameters

<i>obj</i>	New data for wrapper
------------	----------------------

6.51.3.10 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
    self,
    new_metadata )
```

Update metadata.

Parameters

<i>new_metadata</i>	New metadata
---------------------	--------------

6.51.4 Member Data Documentation**6.51.4.1 constants**

`skdaccess.framework.data_class.DataWrapperBase.constants`

6.51.4.2 data

`skdaccess.framework.data_class.DataWrapperBase.data`

6.51.4.3 meta_data

`skdaccess.framework.data_class.DataWrapperBase.meta_data`

6.51.4.4 results

`skdaccess.framework.data_class.DataWrapperBase.results`

6.51.4.5 run_id

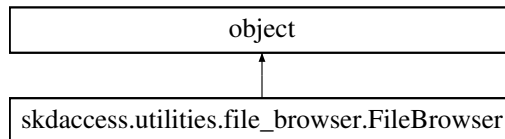
`skdaccess.framework.data_class.DataWrapperBase.run_id`

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

- framework/[data_class.py](#)

6.52 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`

6.52.1 Constructor & Destructor Documentation

6.52.1.1 `__init__()`

```
def skdaccess.utilities.file_browser.FileBrowser.__init__ (  
    self )
```

6.52.2 Member Function Documentation

6.52.2.1 `widget()`

```
def skdaccess.utilities.file_browser.FileBrowser.widget (  
    self )
```

6.52.3 Member Data Documentation

6.52.3.1 dirs

```
skdaccess.utilities.file_browser.FileBrowser.dirs
```

6.52.3.2 files

```
skdaccess.utilities.file_browser.FileBrowser.files
```

6.52.3.3 path

```
skdaccess.utilities.file_browser.FileBrowser.path
```

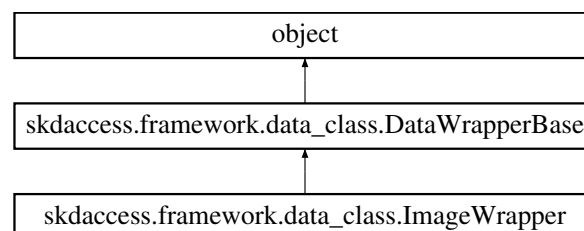
The documentation for this class was generated from the following file:

- [utilities/file_browser.py](#)

6.53 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](#)

6.53.1 Detailed Description

Wrapper for image data.

6.53.2 Member Function Documentation

6.53.2.1 `__len__()`

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
    self ) [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.53.2.2 `addResult()`

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
    self,
    rkey,
    rres ) [inherited]
```

Add a result to the data wrapper.

Parameters

<i>rkey</i>	Result key
<i>rres</i>	Result

6.53.2.3 `deleteData()`

```
def skdaccess.framework.data_class.ImageWrapper.deleteData (
    self,
    label )
```

Delete image.

Parameters

<i>label</i>	Delete image with label
--------------	-------------------------

6.53.2.4 `get()`

```
def skdaccess.framework.data_class.DataWrapperBase.get (
```

```
self ) [inherited]
```

Retrieve stored data.

Returns

Stored data

6.53.2.5 getIterator()

```
def skdaccess.framework.data_class.ImageWrapper.getIterator (
    self )
```

Get an iterator to the data.

Returns

Iterator yielding (label, image_data)

6.53.2.6 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
    self ) [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.53.2.7 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
    self ) [inherited]
```

Get the Run ID.

Returns

run_id

6.53.2.8 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
    self,
    key = None ) [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.53.2.9 reset()

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

Reset data back to original state.

6.53.2.10 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
    self,
    obj ) [inherited]
```

Updated wrapped data.

Parameters

<i>obj</i>	New data for wrapper
------------	----------------------

6.53.2.11 updateData()

```
def skdaccess.framework.data_class.ImageWrapper.updateData (
    self,
    label,
    new_data )
```

Change image.

Parameters

<i>label</i>	Label of data to be changed
<i>new_data</i>	New data to replace old data

6.53.2.12 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
    self,
    new_metadata ) [inherited]
```

Update metadata.

Parameters

<i>new_metadata</i>	New metadata
---------------------	--------------

6.53.3 Member Data Documentation

6.53.3.1 constants

```
skdaccess.framework.data_class.DataWrapperBase.constants [inherited]
```

6.53.3.2 data

```
skdaccess.framework.data_class.DataWrapperBase.data [inherited]
```

6.53.3.3 meta_data

```
skdaccess.framework.data_class.DataWrapperBase.meta_data [inherited]
```

6.53.3.4 results

`skdaccess.framework.data_class.DataWrapperBase.results` [inherited]

6.53.3.5 run_id

`skdaccess.framework.data_class.DataWrapperBase.run_id` [inherited]

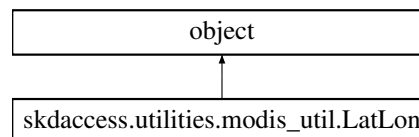
The documentation for this class was generated from the following file:

- framework/[data_class.py](#)

6.54 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](#)

6.54.1 Detailed Description

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

6.54.2 Constructor & Destructor Documentation

6.54.2.1 __init__()

```
def skdaccess.utilities.modis_util.LatLon.__init__ (
    self,
    metadata,
    x_offset = 0,
    y_offset = 0 )
```

Initialize getLatLon object.

Parameters

<i>metadata</i>	Image metadata
<i>x_offset</i>	Pixel offset (used when gridding data)
<i>y_offset</i>	Pixel offset (used when gridding data)

6.54.3 Member Function Documentation

6.54.3.1 __call__()

```
def skdaccess.utilities.modis_util.LatLon.__call__ (
    self,
    y,
    x )
```

Convert pixel coordinates to lat/lon.

Parameters

<i>y</i>	y coordinate
<i>x</i>	x coordinate

Returns

(lat, lon)

6.54.4 Member Data Documentation**6.54.4.1 alat**

`skdaccess.utilities.modis_util.LatLon.alat`

6.54.4.2 alon

`skdaccess.utilities.modis_util.LatLon.alon`

6.54.4.3 lat_data

`skdaccess.utilities.modis_util.LatLon.lat_data`

6.54.4.4 lon_data

`skdaccess.utilities.modis_util.LatLon.lon_data`

6.54.4.5 x_offset

`skdaccess.utilities.modis_util.LatLon.x_offset`

6.54.4.6 y_offset

`skdaccess.utilities.modis_util.LatLon.y_offset`

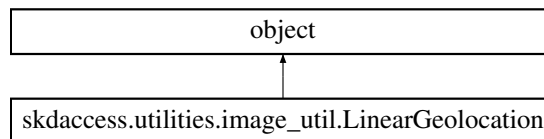
The documentation for this class was generated from the following file:

- [utilities/modis_util.py](#)

6.55 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`

6.55.1 Detailed Description

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

Assumes a linear relationship between pixel and geodetic coordinates

6.55.2 Constructor & Destructor Documentation

6.55.2.1 `__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

<i>data</i>	Numpy 2d data
<i>extents</i>	Latitude and longitude extents
<i>x_offset</i>	Pixel offset in x
<i>y_offset</i>	Pixel offset in y
<i>flip_y</i>	The y axis has been flipped so that increasing y values are decreasing in latitude

6.55.3 Member Function Documentation

6.55.3.1 `getExtents()`

```
def skdaccess.utilities.image_util.LinearGeolocation.getExtents (
    self )
```

Retrieve the extents of the data.

Returns

(minimum_longitude, maximum_longitude, minimum_latitude, maximum_latitude)

6.55.3.2 `getLatLon()`

```
def skdaccess.utilities.image_util.LinearGeolocation.getLatLon (
    self,
    y,
    x )
```

Retrive the latitude and longitude from pixel coordinates.

Parameters

<i>y</i>	The y pixel
<i>x</i>	The x pixel

Returns

(latitude, longitude) of the pixel coordinate

6.55.3.3 getYX()

```
def skdaccess.utilities.image_util.LinearGeolocation.getYX (
    self,
    lat,
    lon )
```

Retrive the pixel coordinates from the latitude and longitude.

Parameters

<i>lat</i>	The Latitude
<i>lon</i>	The Longitude

Returns

(y, x) pixel coordinates of the input latitude and longitude

6.55.4 Member Data Documentation**6.55.4.1 flip_y**

```
skdaccess.utilities.image_util.LinearGeolocation.flip_y
```

6.55.4.2 lat_extents

```
skdaccess.utilities.image_util.LinearGeolocation.lat_extents
```

6.55.4.3 lat_pixel_size

`skdaccess.utilities.image_util.LinearGeolocation.lat_pixel_size`

6.55.4.4 len_x

`skdaccess.utilities.image_util.LinearGeolocation.len_x`

6.55.4.5 len_y

`skdaccess.utilities.image_util.LinearGeolocation.len_y`

6.55.4.6 lon_extents

`skdaccess.utilities.image_util.LinearGeolocation.lon_extents`

6.55.4.7 lon_pixel_size

`skdaccess.utilities.image_util.LinearGeolocation.lon_pixel_size`

6.55.4.8 start_lat

`skdaccess.utilities.image_util.LinearGeolocation.start_lat`

6.55.4.9 start_lon

`skdaccess.utilities.image_util.LinearGeolocation.start_lon`

6.55.4.10 `x_offset`

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

6.55.4.11 `y_offset`

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

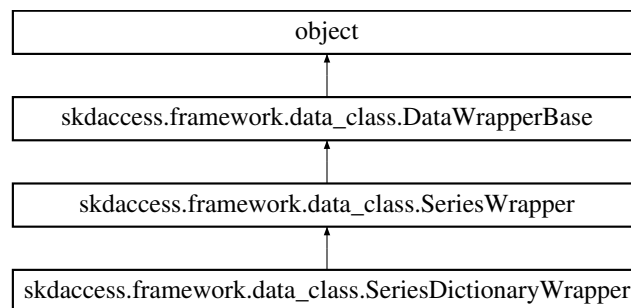
The documentation for this class was generated from the following file:

- [utilities/image_util.py](#)

6.56 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`

6.56.1 Detailed Description

Data wrapper for series data using a dictionary of data frames.

6.56.2 Member Function Documentation

6.56.2.1 `__len__()`

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
    self ) [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.56.2.2 `addResult()`

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
    self,
    rkey,
    rres ) [inherited]
```

Add a result to the data wrapper.

Parameters

<i>rkey</i>	Result key
<i>rres</i>	Result

6.56.2.3 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
    self ) [inherited]
```

Retrieve stored data.

Returns

Stored data

6.56.2.4 getIndices()

```
def skdaccess.framework.data_class.SeriesDictionaryWrapper.getIndices (
    self )
```

Get the indices of the data.

Returns

index of data

6.56.2.5 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

6.56.2.6 `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

6.56.2.7 `getResults()`

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
    self ) [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.56.2.8 `getRunID()`

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
    self ) [inherited]
```

Get the Run ID.

Returns

run_id

6.56.2.9 `info()`

```
def skdaccess.framework.data_class.DataWrapperBase.info (
    self,
    key = None ) [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.56.2.10 reset()

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

Reset data back to original state.

6.56.2.11 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
    self,
    obj ) [inherited]
```

Updated wrapped data.

Parameters

<i>obj</i>	New data for wrapper
------------	----------------------

6.56.2.12 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
    self,
    new_metadata ) [inherited]
```

Update metadata.

Parameters

<i>new_metadata</i>	New metadata
---------------------	--------------

6.56.3 Member Data Documentation

6.56.3.1 constants

```
skdaccess.framework.data_class.DataWrapperBase.constants [inherited]
```

6.56.3.2 data

`skdaccess.framework.data_class.DataWrapperBase.data` [inherited]

6.56.3.3 data_names

`skdaccess.framework.data_class.SeriesWrapper.data_names` [inherited]

6.56.3.4 error_names

`skdaccess.framework.data_class.SeriesWrapper.error_names` [inherited]

6.56.3.5 meta_data

`skdaccess.framework.data_class.DataWrapperBase.meta_data` [inherited]

6.56.3.6 results

`skdaccess.framework.data_class.DataWrapperBase.results` [inherited]

6.56.3.7 run_id

`skdaccess.framework.data_class.DataWrapperBase.run_id` [inherited]

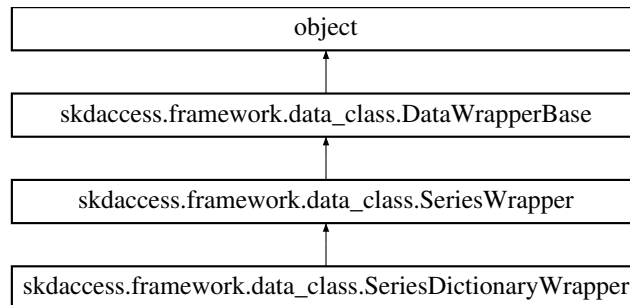
The documentation for this class was generated from the following file:

- framework/[data_class.py](#)

6.57 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 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](#)

6.57.1 Detailed Description

Data wrapper for series data using a data panel.

6.57.2 Constructor & Destructor Documentation

6.57.2.1 `__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

<i>obj_wrap</i>	Pandas data panel to wrap
<i>data_names</i>	List of data column names
<i>error_names</i>	List of error column names
<i>meta_data</i>	Metadata
<i>run_id</i>	ID of run

6.57.3 Member Function Documentation

6.57.3.1 `__len__()`

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
    self ) [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.57.3.2 `addResult()`

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
    self,
    rkey,
    rres ) [inherited]
```

Add a result to the data wrapper.

Parameters

<i>rkey</i>	Result key
<i>rres</i>	Result

6.57.3.3 `get()`

```
def skdaccess.framework.data_class.DataWrapperBase.get (
    self ) [inherited]
```

Retrieve stored data.

Returns

Stored data

6.57.3.4 getIndices()

```
def skdaccess.framework.data_class.SeriesWrapper.getIndices (
    self )
```

Get the indicies of the data.

Returns

index of data

6.57.3.5 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

6.57.3.6 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

6.57.3.7 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
    self ) [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.57.3.8 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
    self ) [inherited]
```

Get the Run ID.

Returns

run_id

6.57.3.9 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
    self,
    key = None ) [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.57.3.10 reset()

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

Reset data back to original state.

6.57.3.11 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
    self,
    obj ) [inherited]
```

Updated wrapped data.

Parameters

<i>obj</i>	New data for wrapper
------------	----------------------

6.57.3.12 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
    self,
    new_metadata ) [inherited]
```

Update metadata.

Parameters

<i>new_metadata</i>	New metadata
---------------------	--------------

6.57.4 Member Data Documentation**6.57.4.1 constants**

```
skdaccess.framework.data_class.DataWrapperBase.constants [inherited]
```

6.57.4.2 data

```
skdaccess.framework.data_class.DataWrapperBase.data [inherited]
```

6.57.4.3 data_names

```
skdaccess.framework.data_class.SeriesWrapper.data_names
```


6.57.4.4 error_names

```
skdaccess.framework.data_class.SeriesWrapper.error_names
```

6.57.4.5 meta_data

```
skdaccess.framework.data_class.DataWrapperBase.meta_data [inherited]
```

6.57.4.6 results

```
skdaccess.framework.data_class.DataWrapperBase.results [inherited]
```

6.57.4.7 run_id

```
skdaccess.framework.data_class.DataWrapperBase.run_id [inherited]
```

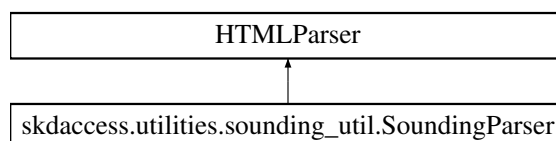
The documentation for this class was generated from the following file:

- framework/[data_class.py](#)

6.58 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`

6.58.1 Detailed Description

This class parses Wyoming Sounding data.

6.58.2 Constructor & Destructor Documentation

6.58.2.1 `__init__()`

```
def skdaccess.utilities.sounding_util.SoundingParser.__init__ (  
    self )
```

Initialize [SoundingParser](#).

6.58.3 Member Function Documentation

6.58.3.1 `handle_data()`

```
def skdaccess.utilities.sounding_util.SoundingParser.handle_data (  
    self,  
    data )
```

Function to parse data between <pre> tags.

Parameters

<i>data</i>	Input data
-------------	------------

6.58.3.2 handle_endtag()

```
def skdaccess.utilities.sounding_util.SoundingParser.handle_endtag (
    self,
    tag )
```

Function called everytime an end tag is encountered.

Parameters

<i>tag</i>	Ending tag
------------	------------

6.58.3.3 handle_starttag()

```
def skdaccess.utilities.sounding_util.SoundingParser.handle_starttag (
    self,
    tag,
    attrs )
```

Function called everytime a start tag is encountered.

Parameters

<i>tag</i>	Starting tag
<i>attrs</i>	Tag attributes

6.58.4 Member Data Documentation

6.58.4.1 data_dict

```
skdaccess.utilities.sounding_util.SoundingParser.data_dict
```

6.58.4.2 in_header

`skdaccess.utilities.sounding_util.SoundingParser.in_header`

6.58.4.3 in_pre_tag

`skdaccess.utilities.sounding_util.SoundingParser.in_pre_tag`

6.58.4.4 label

`skdaccess.utilities.sounding_util.SoundingParser.label`

6.58.4.5 metadata_dict

`skdaccess.utilities.sounding_util.SoundingParser.metadata_dict`

6.58.4.6 read_data

`skdaccess.utilities.sounding_util.SoundingParser.read_data`

6.58.4.7 tmp

`skdaccess.utilities.sounding_util.SoundingParser.tmp`

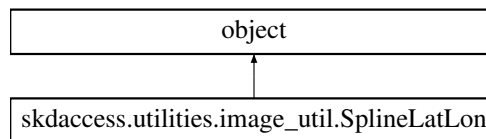
The documentation for this class was generated from the following file:

- [utilities/sounding_util.py](#)

6.59 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`

6.59.1 Detailed Description

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

6.59.2 Constructor & Destructor Documentation

6.59.2.1 `__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

<i>lat_func</i>	Latitude spline function
<i>lon_func</i>	Longitude spline function
<i>lat_grid</i>	Latitude grid
<i>lon_grid</i>	Longitude grid
<i>x_points</i>	1d array of x coordinates
<i>y_points</i>	1d array of y coordinates
<i>lon_extents</i>	Extent of data in longitude
<i>lat_extents</i>	Extent of data in latitude
<i>y_num_pixels</i>	Number of y coordinates
<i>x_num_pixels</i>	Number of x coordinates
<i>x_offset</i>	Offset in the x coordinate
<i>y_offset</i>	Offset in the y coordinate
<i>interp_type</i>	Interpolate type. Currently only 'grid' type is supported

6.59.3 Member Function Documentation

6.59.3.1 `__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)

6.59.4 Member Data Documentation**6.59.4.1 lat_func**

`skdaccess.utilities.image_util.SplineLatLon.lat_func`

6.59.4.2 lon_func

`skdaccess.utilities.image_util.SplineLatLon.lon_func`

6.59.4.3 x_offset

`skdaccess.utilities.image_util.SplineLatLon.x_offset`

6.59.4.4 y_offset

`skdaccess.utilities.image_util.SplineLatLon.y_offset`

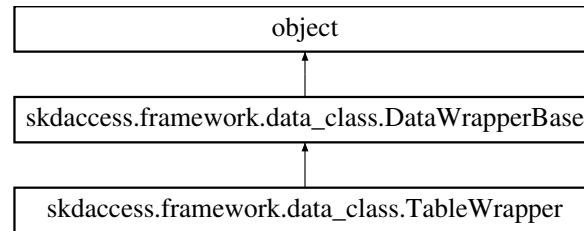
The documentation for this class was generated from the following file:

- [utilities/image_util.py](#)

6.60 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`

6.60.1 Detailed Description

Data wrapper for table data using an ordered dictionary.

6.60.2 Constructor & Destructor Documentation

6.60.2.1 `__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

<code>obj_wrap</code>	Data to be wrapped
<code>run_id</code>	ID of the run
<code>meta_data</code>	Metadata to store with data
<code>default_columns</code>	Default columns for pipeline items
<code>default_error_columns</code>	Default error columns for pipeline items

6.60.3 Member Function Documentation

6.60.3.1 `__len__()`

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
    self ) [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.60.3.2 `addColumn()`

```
def skdaccess.framework.data_class.TableWrapper.addColumn (
    self,
    label,
    column_names,
    new_data )
```

Add new column to data.

Parameters

<i>label</i>	Data label
<i>column_names</i>	Names of columns to update
<i>new_data</i>	New data to add

6.60.3.3 `addResult()`

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
    self,
    rkey,
    rres ) [inherited]
```

Add a result to the data wrapper.

Parameters

<i>rkey</i>	Result key
<i>rres</i>	Result

6.60.3.4 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
    self ) [inherited]
```

Retrieve stored data.

Returns

Stored data

6.60.3.5 getDefaultColumns()

```
def skdaccess.framework.data_class.TableWrapper.getDefaultColumns (
    self )
```

Get the default columns of data.

Returns

List of default columns

6.60.3.6 getDefaultErrorColumns()

```
def skdaccess.framework.data_class.TableWrapper.getDefaultErrorColumns (
    self )
```

Get the default error columns of data.

Returns

List of default error columns

6.60.3.7 `getIterator()`

```
def skdaccess.framework.data_class.TableWrapper.getIterator (
    self )
```

Iterator access to data.

Returns

iterator to (label, data frame) from Dictionary

6.60.3.8 `getLength()`

```
def skdaccess.framework.data_class.TableWrapper.getLength (
    self )
```

Get number of data frames.

Returns

Number of data frames

6.60.3.9 `getResults()`

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
    self ) [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.60.3.10 `getRunID()`

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
    self ) [inherited]
```

Get the Run ID.

Returns

run_id

6.60.3.11 info()

```
def skdaccess.framework.data_class.DataWrapperBase.info (
    self,
    key = None ) [inherited]
```

Get information about data wrapper.

Returns

The stored metadata

6.60.3.12 removeFrames()

```
def skdaccess.framework.data_class.TableWrapper.removeFrames (
    self,
    label_list )
```

Remove Data Frames from wrapper.

Parameters

<i>label_list</i>	List of labels to remove
-------------------	--------------------------

6.60.3.13 reset()

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

Reset data back to original state.

6.60.3.14 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
    self,
    obj ) [inherited]
```

Updated wrapped data.

Parameters

<i>obj</i>	New data for wrapper
------------	----------------------

6.60.3.15 updateData()

```
def skdaccess.framework.data_class.TableWrapper.updateData (
    self,
    label,
    index,
    column_names,
    new_data )
```

Update wrapped data.

Parameters

<i>label</i>	Data label
<i>index</i>	Index of data to update
<i>column_names</i>	Names of columns to update
<i>new_data</i>	Data to replace the old data

6.60.3.16 updateFrames()

```
def skdaccess.framework.data_class.TableWrapper.updateFrames (
    self,
    label_list,
    frame_list )
```

Update data frames.

Parameters

<i>label_list</i>	List of labels to update
<i>frame_list</i>	List of updated frames

6.60.3.17 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
```

```
self,  
new_metadata ) [inherited]
```

Update metadata.

Parameters

<i>new_metadata</i>	New metadata
---------------------	--------------

6.60.4 Member Data Documentation

6.60.4.1 constants

`skdaccess.framework.data_class.DataWrapperBase.constants` [inherited]

6.60.4.2 data

`skdaccess.framework.data_class.DataWrapperBase.data` [inherited]

6.60.4.3 default_columns

`skdaccess.framework.data_class.TableWrapper.default_columns`

6.60.4.4 default_error_columns

`skdaccess.framework.data_class.TableWrapper.default_error_columns`

6.60.4.5 meta_data

`skdaccess.framework.data_class.DataWrapperBase.meta_data` [inherited]

6.60.4.6 results

`skdaccess.framework.data_class.DataWrapperBase.results` [inherited]

6.60.4.7 run_id

`skdaccess.framework.data_class.DataWrapperBase.run_id` [inherited]

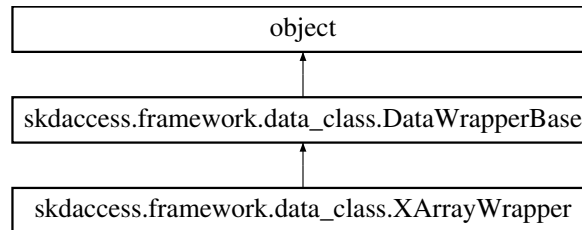
The documentation for this class was generated from the following file:

- framework/[data_class.py](#)

6.61 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 iterates 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](#)

6.61.1 Detailed Description

Wrapper for xarrays.

6.61.2 Constructor & Destructor Documentation

6.61.2.1 `__init__()`

```
def skdaccess.framework.data_class.XArrayWrapper.__init__ (
    self,
    obj_wrap,
    index_list,
    run_id = -1 )
```

6.61.3 Member Function Documentation

6.61.3.1 `__len__()`

```
def skdaccess.framework.data_class.DataWrapperBase.__len__ (
    self ) [inherited]
```

Get length of wrapped data.

Returns

length of wrapped data

6.61.3.2 `addResult()`

```
def skdaccess.framework.data_class.DataWrapperBase.addResult (
    self,
    rkey,
    rres ) [inherited]
```

Add a result to the data wrapper.

Parameters

<i>rkey</i>	Result key
<i>rres</i>	Result

6.61.3.3 get()

```
def skdaccess.framework.data_class.DataWrapperBase.get (
    self ) [inherited]
```

Retrieve stored data.

Returns

Stored data

6.61.3.4 getIterator()

```
def skdaccess.framework.data_class.XArrayWrapper.getIterator (
    self )
```

Get an iterator that iterates over the index.

Returns

iterator to data

6.61.3.5 getResults()

```
def skdaccess.framework.data_class.DataWrapperBase.getResults (
    self ) [inherited]
```

Retrieve accumulated results, if any.

Returns

store results

6.61.3.6 getRunID()

```
def skdaccess.framework.data_class.DataWrapperBase.getRunID (
    self ) [inherited]
```

Get the Run ID.

Returns

run_id

6.61.3.7 info()

```
def skdaccess.framework.data_class.XArrayWrapper.info (
    self,
    key = None )
```

Get information about xarray data wrapper.

Returns

The stored metadata

6.61.3.8 reset()

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

Reset data back to original state.

6.61.3.9 update()

```
def skdaccess.framework.data_class.DataWrapperBase.update (
    self,
    obj ) [inherited]
```

Updated wrapped data.

Parameters

<i>obj</i>	New data for wrapper
------------	----------------------

6.61.3.10 updateMetadata()

```
def skdaccess.framework.data_class.DataWrapperBase.updateMetadata (
    self,
    new_metadata ) [inherited]
```

Update metadata.

Parameters

<i>new_metadata</i>	New metadata
---------------------	--------------

6.61.4 Member Data Documentation**6.61.4.1 constants**

```
skdaccess.framework.data_class.DataWrapperBase.constants [inherited]
```

6.61.4.2 data

```
skdaccess.framework.data_class.DataWrapperBase.data [inherited]
```

6.61.4.3 index_list

```
skdaccess.framework.data_class.XArrayWrapper.index_list
```

6.61.4.4 meta_data

`skdaccess.framework.data_class.DataWrapperBase.meta_data` [inherited]

6.61.4.5 results

`skdaccess.framework.data_class.DataWrapperBase.results` [inherited]

6.61.4.6 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 `finance/timeseries/stream.py` File Reference

Classes

- class `skdaccess.finance.timeseries.stream.DataFetcher`
Data Fetcher for retrieving stock data.

Namespaces

- `skdaccess.finance.timeseries.stream`

7.2 `astro/spectra/stream.py` File Reference

Classes

- class `skdaccess.astro.spectra.stream.DataFetcher`
Data Fetcher for Sloan Digital Sky Survey spectra.

Namespaces

- `skdaccess.astro.spectra.stream`

7.3 `engineering/la/generic/stream.py` File Reference

Classes

- class `skdaccess.engineering.la.generic.stream.DataFetcher`
Class for handling data requests to data.lacity.org.

Namespaces

- [skdaccess.engineering.la.generic.stream](#)

7.4 engineering/la/traffic_counts/stream.py File Reference

Classes

- class [skdaccess.engineering.la.traffic_counts.stream.DataFetcher](#)
DataFetcher for retrieving traffic counts from LA.

Namespaces

- [skdaccess.engineering.la.traffic_counts.stream](#)

7.5 engineering/webcam/mit_sailing/stream.py File Reference

Classes

- class [skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher](#)
Data Fetcher for retrieving webcam images from the MIT Sailing Pavilion.

Namespaces

- [skdaccess.engineering.webcam.mit_sailing.stream](#)

7.6 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](#)

7.7 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](#)

7.8 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](#)

7.9 solar/sdo/data_fetcher.py File Reference

Classes

- class [skdaccess.solar.sdo.DataFetcher](#)
Data Fetcher for the Solar Dynamics Observatory.

Namespaces

- [skdaccess.solar.sdo.data_fetcher](#)

7.10 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](#)

7.11 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](#)

7.12 geo/grace/data_fetcher.py File Reference

Classes

- class [skdaccess.geo.grace.DataFetcher](#)
Data Fetcher for GRACE data.

Namespaces

- [skdaccess.geo.grace.data_fetcher](#)

7.13 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](#)

7.14 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](#)

7.15 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](#)

7.16 geo/ngl_gps/data_fetcher.py File Reference

Classes

- class [skdaccess.geo.ngl_gps.DataFetcher](#)

Data fetcher for GPS data from Nevada Geodetic Laboratory.

Namespaces

- [skdaccess.geo.ngl_gps.data_fetcher](#)

7.17 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](#)

7.18 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](#)

7.19 geo/gldas/data_fetcher.py File Reference

Classes

- class [skdaccess.geo.gldas.DataFetcher](#)
Data Fetcher for GLDAS data.

Namespaces

- [skdaccess.geo.gldas.data_fetcher](#)

7.20 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](#)

7.21 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](#)

7.22 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](#)

7.23 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](#)

7.24 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](#)

7.25 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](#)

7.26 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](#)

7.27 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](#)

7.28 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](#)

7.29 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](#)

7.30 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](#)

7.31 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](#)

7.32 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](#)

7.33 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](#)

7.34 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](#)

7.35 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](#)

7.36 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](#)

7.37 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](#)

7.38 utilities/file_browser.py File Reference

Classes

- class [skdaccess.utilities.file_browser.FileBrowser](#)

Namespaces

- [skdaccess.utilities.file_browser](#)

7.39 utilities/file_util.py File Reference

Namespaces

- [skdaccess.utilities.file_util](#)

Functions

- def [skdaccess.utilities.file_util.openPandasHDFStoreLocking](#) (filename, mode)
Open a pandas HDF store that may be locked:

7.40 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)

7.41 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.

7.42 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](#)

7.43 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.

7.44 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.parseLonoFile](#) (in_file, compression='infer')

7.45 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.

7.46 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.

7.47 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.

7.48 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 satellite data.

7.49 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.

7.50 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.

7.51 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)
Progress bar using tqdm.
- def [skdaccess.utilities.support.convertToStr](#) (in_value, zfill=0)
- def [skdaccess.utilities.support.join_string](#) (part1, part2, concatenation_string='AND', seperator=' ')
Join two strings together using a concatenation string.

7.52 utilities/uavsar_util.py File Reference

Namespaces

- [skdaccess.utilities.uavsar_util](#)

Functions

- def [skdaccess.utilities.uavsar_util.readUAVSARMetadata](#) (in_file)
Parse UAVSAR metadata.

Index

`__call__`

- `skdaccess::framework::param_class::AutoList`, [62](#)
- `skdaccess::framework::param_class::AutoListCycle`, [67](#)
- `skdaccess::framework::param_class::AutoList`↔
`Permute`, [71](#)
- `skdaccess::framework::param_class::AutoList`↔
`Remove`, [75](#)
- `skdaccess::framework::param_class::AutoList`↔
`Subset`, [79](#)
- `skdaccess::framework::param_class::AutoParam`, [84](#)
- `skdaccess::framework::param_class::AutoParamList`, [86](#)
- `skdaccess::framework::param_class::AutoParam`↔
`ListCycle`, [89](#)
- `skdaccess::framework::param_class::AutoParam`↔
`MinMax`, [92](#)
- `skdaccess::utilities::image_util::SplineLatLon`, [390](#)
- `skdaccess::utilities::modis_util::LatLon`, [367](#)

`__getitem__`

- `skdaccess::framework::param_class::AutoList`, [62](#)
- `skdaccess::framework::param_class::AutoListCycle`, [67](#)
- `skdaccess::framework::param_class::AutoList`↔
`Permute`, [71](#)
- `skdaccess::framework::param_class::AutoList`↔
`Remove`, [75](#)
- `skdaccess::framework::param_class::AutoList`↔
`Subset`, [79](#)

`__init__`

- `skdaccess::astro::kepler::data_fetcher::DataFetcher`, [96](#)
- `skdaccess::astro::spectra::stream::DataFetcher`, [269](#)
- `skdaccess::astro::voyager::data_fetcher::Data`↔
`Fetcher`, [298](#)
- `skdaccess::engineering::la::generic::stream::Data`↔
`Fetcher`, [104](#)
- `skdaccess::engineering::la::traffic_counts::stream`↔
`::DataFetcher`, [142](#)
- `skdaccess::engineering::webcam::mit_sailing`↔
`::stream::DataFetcher`, [179](#)
- `skdaccess::finance::timeseries::stream::Data`↔
`Fetcher`, [291](#)
- `skdaccess::framework::data_class::DataFetcher`↔
`Base`, [319](#)

- `skdaccess::framework::data_class::DataWrapper`↔
`Base`, [354](#)
- `skdaccess::framework::data_class::SeriesWrapper`, [380](#)
- `skdaccess::framework::data_class::TableWrapper`, [393](#)
- `skdaccess::framework::data_class::XArrayWrapper`, [401](#)
- `skdaccess::framework::param_class::AutoList`, [62](#)
- `skdaccess::framework::param_class::AutoListCycle`, [66](#)
- `skdaccess::framework::param_class::AutoList`↔
`Remove`, [75](#)
- `skdaccess::framework::param_class::AutoParam`, [83](#)
- `skdaccess::framework::param_class::AutoParamList`, [86](#)
- `skdaccess::framework::param_class::AutoParam`↔
`ListCycle`, [89](#)
- `skdaccess::framework::param_class::AutoParam`↔
`MinMax`, [91](#)
- `skdaccess::geo::era_interim::cache::data_fetcher::`↔
`DataFetcher`, [253](#)
- `skdaccess::geo::gldas::data_fetcher::DataFetcher`, [111](#)
- `skdaccess::geo::grace::data_fetcher::DataFetcher`, [275](#)
- `skdaccess::geo::grace::mascon::cache::data`↔
`fetcher::DataFetcher`, [219](#)
- `skdaccess::geo::groundwater::data_fetcher::Data`↔
`Fetcher`, [151](#)
- `skdaccess::geo::imsdnhs::data_fetcher::Data`↔
`Fetcher`, [246](#)
- `skdaccess::geo::magnetometer::data_fetcher::`↔
`DataFetcher`, [127](#)
- `skdaccess::geo::mahali::rinex::data_fetcher::Data`↔
`Fetcher`, [282](#)
- `skdaccess::geo::mahali::tec::data_fetcher::Data`↔
`Fetcher`, [228](#)
- `skdaccess::geo::mahali::temperature::data_fetcher`↔
`::DataFetcher`, [308](#)
- `skdaccess::geo::modis::cache::cloud_mask::data`↔
`fetcher::DataFetcher`, [185](#)
- `skdaccess::geo::modis::cache::cloud_opacity`↔
`::data_fetcher::DataFetcher`, [176](#)
- `skdaccess::geo::modis::cache::data_fetcher::Data`↔

- Fetcher, [188](#)
 - skdaccess::geo::modis::cache::reflectance::data_↵
fetcher::DataFetcher, [186](#)
 - skdaccess::geo::modis::stream::cloud_mask::data_↵
_fetcher::DataFetcher, [199](#)
 - skdaccess::geo::modis::stream::cloud_opacity_↵
::data_fetcher::DataFetcher, [198](#)
 - skdaccess::geo::modis::stream::data_fetcher::↵
DataFetcher, [212](#)
 - skdaccess::geo::modis::stream::reflectance::data_↵
fetcher::DataFetcher, [177](#)
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
[261](#)
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, [236](#)
 - skdaccess::geo::sentinel_1::cache::data_fetcher::↵
DataFetcher, [118](#)
 - skdaccess::geo::srtm::cache::data_fetcher::Data↵
Fetcher, [159](#)
 - skdaccess::geo::uavsar::cache::data_fetcher::Data↵
Fetcher, [169](#)
 - skdaccess::geo::wyoming_sounding::cache::data_↵
fetcher::DataFetcher, [134](#)
 - skdaccess::geo::wyoming_sounding::stream::data_↵
_fetcher::DataFetcher, [144](#)
 - skdaccess::planetary::ode::cache::data_fetcher::↵
DataFetcher, [201](#)
 - skdaccess::solar::sdo::data_fetcher::DataFetcher,
[314](#)
 - skdaccess::utilities::file_browser::FileBrowser, [359](#)
 - skdaccess::utilities::image_util::AffineGlobalCoords,
[59](#)
 - skdaccess::utilities::image_util::LinearGeolocation,
[370](#)
 - skdaccess::utilities::image_util::SplineLatLon, [389](#)
 - skdaccess::utilities::modis_util::LatLon, [367](#)
 - skdaccess::utilities::sounding_util::SoundingParser,
[386](#)
- __len__
 - skdaccess::framework::data_class::DataWrapper↵
Base, [355](#)
 - skdaccess::framework::data_class::ImageWrapper,
[361](#)
 - skdaccess::framework::data_class::SeriesDictionary↵
Wrapper, [374](#)
 - skdaccess::framework::data_class::SeriesWrapper,
[380](#)
 - skdaccess::framework::data_class::TableWrapper,
[394](#)
 - skdaccess::framework::data_class::XArrayWrapper,
[401](#)
 - skdaccess::framework::param_class::AutoList, [63](#)
 - skdaccess::framework::param_class::AutoListCycle,
[67](#)
 - skdaccess::framework::param_class::AutoList↵
- Permute, [71](#)
 - skdaccess::framework::param_class::AutoList↵
Remove, [76](#)
 - skdaccess::framework::param_class::AutoList↵
Subset, [80](#)
 - skdaccess::geo::mahali::rinex::data_wrapper::Data↵
Wrapper, [349](#)
- __setitem__
 - skdaccess::framework::param_class::AutoList, [63](#)
 - skdaccess::framework::param_class::AutoListCycle,
[68](#)
 - skdaccess::framework::param_class::AutoList↵
Permute, [72](#)
 - skdaccess::framework::param_class::AutoList↵
Remove, [76](#)
 - skdaccess::framework::param_class::AutoList↵
Subset, [80](#)
- __str__
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
[96](#)
 - skdaccess::astro::spectra::stream::DataFetcher, [270](#)
 - skdaccess::astro::voyager::data_fetcher::Data↵
Fetcher, [298](#)
 - skdaccess::engineering::la::generic::stream::Data↵
Fetcher, [105](#)
 - skdaccess::engineering::webcam::mit_sailing↵
::stream::DataFetcher, [180](#)
 - skdaccess::finance::timeseries::stream::Data↵
Fetcher, [291](#)
 - skdaccess::framework::data_class::DataFetcher↵
Base, [320](#)
 - skdaccess::framework::data_class::DataFetcher↵
Cache, [325](#)
 - skdaccess::framework::data_class::DataFetcher↵
Local, [332](#)
 - skdaccess::framework::data_class::DataFetcher↵
Storage, [338](#)
 - skdaccess::framework::data_class::DataFetcher↵
Stream, [344](#)
 - skdaccess::framework::param_class::AutoList, [63](#)
 - skdaccess::framework::param_class::AutoListCycle,
[68](#)
 - skdaccess::framework::param_class::AutoList↵
Permute, [72](#)
 - skdaccess::framework::param_class::AutoList↵
Remove, [76](#)
 - skdaccess::framework::param_class::AutoList↵
Subset, [80](#)
 - skdaccess::framework::param_class::AutoParam, [84](#)
 - skdaccess::framework::param_class::AutoParamList,
[86](#)
 - skdaccess::framework::param_class::AutoParam↵
ListCycle, [89](#)
 - skdaccess::framework::param_class::AutoParam↵

- MinMax, [92](#)
- skdaccess::geo::era_interim::cache::data_fetcher::DataFetcher, [254](#)
- skdaccess::geo::gldas::data_fetcher::DataFetcher, [111](#)
- skdaccess::geo::grace::data_fetcher::DataFetcher, [276](#)
- skdaccess::geo::grace::mascon::cache::data_fetcher::DataFetcher, [220](#)
- skdaccess::geo::groundwater::data_fetcher::DataFetcher, [152](#)
- skdaccess::geo::imsdnhs::data_fetcher::DataFetcher, [247](#)
- skdaccess::geo::magnetometer::data_fetcher::DataFetcher, [127](#)
- skdaccess::geo::mahali::rinex::data_fetcher::DataFetcher, [283](#)
- skdaccess::geo::mahali::tec::data_fetcher::DataFetcher, [229](#)
- skdaccess::geo::mahali::temperature::data_fetcher::DataFetcher, [308](#)
- skdaccess::geo::modis::cache::data_fetcher::DataFetcher, [189](#)
- skdaccess::geo::modis::stream::data_fetcher::DataFetcher, [212](#)
- skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, [262](#)
- skdaccess::geo::pbo::data_fetcher::DataFetcher, [238](#)
- skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, [119](#)
- skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, [160](#)
- skdaccess::geo::uavsar::cache::data_fetcher::DataFetcher, [169](#)
- skdaccess::geo::wyoming_sounding::cache::data_fetcher::DataFetcher, [135](#)
- skdaccess::geo::wyoming_sounding::stream::data_fetcher::DataFetcher, [145](#)
- skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, [202](#)
- skdaccess::solar::sdo::data_fetcher::DataFetcher, [314](#)
- addColumn
 - skdaccess::framework::data_class::TableWrapper, [394](#)
- addResult
 - skdaccess::framework::data_class::DataWrapperBase, [355](#)
 - skdaccess::framework::data_class::ImageWrapper, [362](#)
 - skdaccess::framework::data_class::SeriesDictionaryWrapper, [374](#)
 - skdaccess::framework::data_class::SeriesWrapper, [381](#)
- skdaccess::framework::data_class::TableWrapper, [394](#)
- skdaccess::framework::data_class::XArrayWrapper, [401](#)
- skdaccess::geo::mahali::rinex::data_wrapper::DataWrapper, [350](#)
- alat
 - skdaccess::utilities::modis_util::LatLon, [368](#)
- alon
 - skdaccess::utilities::modis_util::LatLon, [368](#)
- antenna_info
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, [243](#)
- ap_paramList
 - skdaccess::astro::kepler::data_fetcher::DataFetcher, [102](#)
 - skdaccess::astro::spectra::stream::DataFetcher, [273](#)
 - skdaccess::astro::voyager::data_fetcher::DataFetcher, [305](#)
 - skdaccess::engineering::la::generic::stream::DataFetcher, [108](#)
 - skdaccess::engineering::webcam::mit_sailing::stream::DataFetcher, [183](#)
 - skdaccess::finance::timeseries::stream::DataFetcher, [295](#)
 - skdaccess::framework::data_class::DataFetcherBase, [323](#)
 - skdaccess::framework::data_class::DataFetcherCache, [331](#)
 - skdaccess::framework::data_class::DataFetcherLocal, [336](#)
 - skdaccess::framework::data_class::DataFetcherStorage, [343](#)
 - skdaccess::framework::data_class::DataFetcherStream, [348](#)
 - skdaccess::geo::era_interim::cache::data_fetcher::DataFetcher, [259](#)
 - skdaccess::geo::gldas::data_fetcher::DataFetcher, [116](#)
 - skdaccess::geo::grace::data_fetcher::DataFetcher, [280](#)
 - skdaccess::geo::grace::mascon::cache::data_fetcher::DataFetcher, [225](#)
 - skdaccess::geo::groundwater::data_fetcher::DataFetcher, [157](#)
 - skdaccess::geo::imsdnhs::data_fetcher::DataFetcher, [251](#)
 - skdaccess::geo::magnetometer::data_fetcher::DataFetcher, [131](#)
 - skdaccess::geo::mahali::rinex::data_fetcher::DataFetcher, [289](#)
 - skdaccess::geo::mahali::tec::data_fetcher::DataFetcher, [234](#)
 - skdaccess::geo::mahali::temperature::data_fetcher::DataFetcher, [308](#)

- skdaccess::geo::modis::cache::data_fetcher::Data←
Fetcher, 195
 - skdaccess::geo::modis::stream::data_fetcher::←
DataFetcher, 216
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
267
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 244
 - skdaccess::geo::sentinel_1::cache::data_fetcher::←
DataFetcher, 124
 - skdaccess::geo::srtm::cache::data_fetcher::Data←
Fetcher, 165
 - skdaccess::geo::uavsar::cache::data_fetcher::Data←
Fetcher, 175
 - skdaccess::geo::wyoming_sounding::cache::data_←
fetcher::DataFetcher, 140
 - skdaccess::geo::wyoming_sounding::stream::data_←
_fetcher::DataFetcher, 148
 - skdaccess::planetary::ode::cache::data_fetcher::←
DataFetcher, 207
 - skdaccess::solar::sdo::data_fetcher::DataFetcher,
318
- app_token
 - skdaccess::engineering::la::generic::stream::Data←
Fetcher, 108
- arcsecond_sampling
 - skdaccess::geo::srtm::cache::data_fetcher::Data←
Fetcher, 165
- astro/kepler/data_fetcher.py, 418
- astro/spectra/stream.py, 407
- astro/voyager/data_fetcher.py, 418
- averageDates
 - skdaccess::utilities::grace_util, 31
- base_url
 - skdaccess::astro::voyager::data_fetcher::Data←
Fetcher, 306
 - skdaccess::engineering::la::generic::stream::Data←
Fetcher, 108
- base_url_and_endpoint
 - skdaccess::engineering::la::generic::stream::Data←
Fetcher, 109
- cacheData
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
96, 97
 - skdaccess::astro::voyager::data_fetcher::Data←
Fetcher, 298
 - skdaccess::framework::data_class::DataFetcher←
Cache, 325
 - skdaccess::geo::era_interim::cache::data_fetcher::←
DataFetcher, 254
 - skdaccess::geo::grace::mascon::cache::data_←
fetcher::DataFetcher, 220
- skdaccess::geo::mahali::rinex::data_fetcher::Data←
Fetcher, 283
- skdaccess::geo::mahali::tec::data_fetcher::Data←
Fetcher, 229
- skdaccess::geo::modis::cache::data_fetcher::Data←
Fetcher, 189, 190
- skdaccess::geo::sentinel_1::cache::data_fetcher::←
DataFetcher, 119
- skdaccess::geo::srtm::cache::data_fetcher::Data←
Fetcher, 160
- skdaccess::geo::uavsar::cache::data_fetcher::Data←
Fetcher, 169
- skdaccess::geo::wyoming_sounding::cache::data_←
fetcher::DataFetcher, 135
- skdaccess::planetary::ode::cache::data_fetcher::←
DataFetcher, 202
- calibrateModis
 - skdaccess::utilities::modis_util, 39
- camera_list
 - skdaccess::engineering::webcam::mit_sailing←
::stream::DataFetcher, 184
- channels
 - skdaccess::geo::magnetometer::data_fetcher::←
DataFetcher, 132
- checkBit
 - skdaccess::utilities::modis_util, 39
- checkIfDataExists
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
97
 - skdaccess::astro::voyager::data_fetcher::Data←
Fetcher, 299
 - skdaccess::framework::data_class::DataFetcher←
Cache, 326
 - skdaccess::geo::era_interim::cache::data_fetcher::←
DataFetcher, 254
 - skdaccess::geo::grace::mascon::cache::data_←
fetcher::DataFetcher, 221
 - skdaccess::geo::mahali::rinex::data_fetcher::Data←
Fetcher, 284
 - skdaccess::geo::mahali::tec::data_fetcher::Data←
Fetcher, 229
 - skdaccess::geo::modis::cache::data_fetcher::Data←
Fetcher, 190
 - skdaccess::geo::sentinel_1::cache::data_fetcher::←
DataFetcher, 120
 - skdaccess::geo::srtm::cache::data_fetcher::Data←
Fetcher, 161
 - skdaccess::geo::uavsar::cache::data_fetcher::Data←
Fetcher, 170
 - skdaccess::geo::wyoming_sounding::cache::data_←
fetcher::DataFetcher, 136
 - skdaccess::planetary::ode::cache::data_fetcher::←
DataFetcher, 203
- combine_water_heights

- skdaccess::utilities::gw_util, [33](#)
- computeEWD
 - skdaccess::utilities::grace_util, [31](#)
- constants
 - skdaccess::framework::data_class::DataWrapper↔
Base, [358](#)
 - skdaccess::framework::data_class::ImageWrapper,
[365](#)
 - skdaccess::framework::data_class::SeriesDictionary↔
Wrapper, [377](#)
 - skdaccess::framework::data_class::SeriesWrapper,
[384](#)
 - skdaccess::framework::data_class::TableWrapper,
[399](#)
 - skdaccess::framework::data_class::XArrayWrapper,
[404](#)
 - skdaccess::geo::mahali::rinex::data_wrapper::Data↔
Wrapper, [352](#)
- convert_date
 - skdaccess::utilities::mahali_util, [37](#)
- convertBinCentersToEdges
 - skdaccess::utilities::image_util, [34](#)
- convertToStr
 - skdaccess::utilities::support, [55](#)
- coordinate_dict
 - skdaccess::geo::imsdnhs::data_fetcher::Data↔
Fetcher, [251](#)
- correct_CRISM_label
 - skdaccess::utilities::ode_util, [43](#)
- correct_file_name_case_in_label
 - skdaccess::utilities::ode_util, [44](#)
- correct_label_file
 - skdaccess::utilities::ode_util, [44](#)
- createGrid
 - skdaccess::utilities::modis_util, [39](#)
- current_index
 - skdaccess::framework::param_class::AutoParam↔
ListCycle, [90](#)
- cutoff
 - skdaccess::geo::groundwater::data_fetcher::Data↔
Fetcher, [157](#)
- data
 - skdaccess::framework::data_class::DataWrapper↔
Base, [358](#)
 - skdaccess::framework::data_class::ImageWrapper,
[365](#)
 - skdaccess::framework::data_class::SeriesDictionary↔
Wrapper, [377](#)
 - skdaccess::framework::data_class::SeriesWrapper,
[384](#)
 - skdaccess::framework::data_class::TableWrapper,
[399](#)
 - skdaccess::framework::data_class::XArrayWrapper,
[404](#)
 - skdaccess::geo::mahali::rinex::data_wrapper::Data↔
Wrapper, [352](#)
- data_dict
 - skdaccess::utilities::sounding_util::SoundingParser,
[387](#)
- data_names
 - skdaccess::framework::data_class::SeriesDictionary↔
Wrapper, [378](#)
 - skdaccess::framework::data_class::SeriesWrapper,
[384](#)
 - skdaccess::geo::era_interim::cache::data_fetcher::↔
DataFetcher, [259](#)
- data_type
 - skdaccess::finance::timeseries::stream::Data↔
Fetcher, [295](#)
 - skdaccess::geo::magnetometer::data_fetcher::↔
DataFetcher, [132](#)
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
[267](#)
- date_list
 - skdaccess::geo::era_interim::cache::data_fetcher::↔
DataFetcher, [259](#)
- date_range
 - skdaccess::geo::mahali::rinex::data_fetcher::Data↔
Fetcher, [289](#)
 - skdaccess::geo::mahali::tec::data_fetcher::Data↔
Fetcher, [234](#)
- dateMismatch
 - skdaccess::utilities::grace_util, [31](#)
- day_end
 - skdaccess::geo::wyoming_sounding::cache::data_↔
fetcher::DataFetcher, [140](#)
 - skdaccess::geo::wyoming_sounding::stream::data_↔
_fetcher::DataFetcher, [149](#)
- day_start
 - skdaccess::geo::wyoming_sounding::cache::data_↔
fetcher::DataFetcher, [140](#)
 - skdaccess::geo::wyoming_sounding::stream::data_↔
_fetcher::DataFetcher, [149](#)
- daynightboth
 - skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, [196](#)
 - skdaccess::geo::modis::stream::data_fetcher::↔
DataFetcher, [216](#)
- decimals
 - skdaccess::framework::param_class::AutoParam↔
MinMax, [93](#)
- default_columns
 - skdaccess::framework::data_class::TableWrapper,
[399](#)
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, [244](#)
- default_error_columns

- skdaccess::framework::data_class::TableWrapper, 399
- skdaccess::geo::pbo::data_fetcher::DataFetcher, 244
- deleteData
 - skdaccess::framework::data_class::ImageWrapper, 362
- dirs
 - skdaccess::utilities::file_browser::FileBrowser, 360
- downloadFullDataset
 - skdaccess::framework::data_class::DataFetcher↔Storage, 338
 - skdaccess::geo::gldas::data_fetcher::DataFetcher, 112
 - skdaccess::geo::grace::data_fetcher::DataFetcher, 276
 - skdaccess::geo::groundwater::data_fetcher::Data↔Fetcher, 152
 - skdaccess::geo::imsdnhs::data_fetcher::Data↔Fetcher, 247
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, 262
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 238
- downloadKeplerData
 - skdaccess::astro::kepler::data_fetcher::DataFetcher, 98
- eastern_lon
 - skdaccess::planetary::ode::cache::data_fetcher::↔DataFetcher, 208
- end_date
 - skdaccess::finance::timeseries::stream::Data↔Fetcher, 295
 - skdaccess::geo::gldas::data_fetcher::DataFetcher, 116
 - skdaccess::geo::grace::data_fetcher::DataFetcher, 280
 - skdaccess::geo::grace::mascon::cache::data_↔fetcher::DataFetcher, 226
 - skdaccess::geo::groundwater::data_fetcher::Data↔Fetcher, 157
 - skdaccess::geo::imsdnhs::data_fetcher::Data↔Fetcher, 251
 - skdaccess::geo::mahali::rinex::data_fetcher::Data↔Fetcher, 289
 - skdaccess::geo::mahali::tec::data_fetcher::Data↔Fetcher, 234
 - skdaccess::geo::mahali::temperature::data_fetcher↔::DataFetcher, 312
 - skdaccess::geo::modis::cache::data_fetcher::Data↔Fetcher, 196
 - skdaccess::geo::modis::stream::data_fetcher::↔DataFetcher, 216
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, 267
- end_hour
 - skdaccess::geo::wyoming_sounding::cache::data_↔fetcher::DataFetcher, 141
 - skdaccess::geo::wyoming_sounding::stream::data_↔fetcher::DataFetcher, 149
- end_time
 - skdaccess::geo::magnetometer::data_fetcher::↔DataFetcher, 132
- engineering/la/generic/stream.py, 407
- engineering/la/traffic_counts/stream.py, 408
- engineering/webcam/mit_sailing/stream.py, 408
- error_names
 - skdaccess::framework::data_class::SeriesDictionary↔Wrapper, 378
 - skdaccess::framework::data_class::SeriesWrapper, 384
- field_names
 - skdaccess::astro::voyager::data_fetcher::Data↔Fetcher, 306
- field_widths
 - skdaccess::astro::voyager::data_fetcher::Data↔Fetcher, 306
- file_name
 - skdaccess::planetary::ode::cache::data_fetcher::↔DataFetcher, 208
- files
 - skdaccess::utilities::file_browser::FileBrowser, 360
- finance/timeseries/stream.py, 407
- find_data
 - skdaccess::geo::modis::cache::data_fetcher::Data↔Fetcher, 191
- flip_y
 - skdaccess::utilities::image_util::LinearGeolocation, 371
- framework/data_class.py, 408
- framework/param_class.py, 409
- generate_links
 - skdaccess::geo::mahali::rinex::data_fetcher::Data↔Fetcher, 289
- generateQueries
 - skdaccess::utilities::sounding_util, 52
- generateURL
 - skdaccess::astro::voyager::data_fetcher::Data↔Fetcher, 300
- geo/era_interim/cache/data_fetcher.py, 412
- geo/gldas/data_fetcher.py, 413
- geo/grace/data_fetcher.py, 411
- geo/grace/mascon/cache/data_fetcher.py, 410
- geo/groundwater/data_fetcher.py, 417
- geo/imsdnhs/data_fetcher.py, 412
- geo/magnetometer/data_fetcher.py, 413
- geo/mahali/rinex/data_fetcher.py, 411
- geo/mahali/rinex/data_wrapper.py, 410

- geo/mahali/tec/data_fetcher.py, 411
- geo/mahali/temperature/data_fetcher.py, 412
- geo/modis/cache/cloud_mask/data_fetcher.py, 414
- geo/modis/cache/cloud_opacity/data_fetcher.py, 414
- geo/modis/cache/data_fetcher.py, 415
- geo/modis/cache/reflectance/data_fetcher.py, 415
- geo/modis/stream/cloud_mask/data_fetcher.py, 416
- geo/modis/stream/cloud_opacity/data_fetcher.py, 415
- geo/modis/stream/data_fetcher.py, 416
- geo/modis/stream/reflectance/data_fetcher.py, 416
- geo/ngl_gps/data_fetcher.py, 412
- geo/pbo/data_fetcher.py, 417
- geo/sentinel_1/cache/data_fetcher.py, 413
- geo/srtm/cache/data_fetcher.py, 417
- geo/uavsar/cache/data_fetcher.py, 416
- geo/wyoming_sounding/cache/data_fetcher.py, 414
- geo/wyoming_sounding/stream/data_fetcher.py, 414
- get
 - skdaccess::framework::data_class::DataWrapper←
Base, 355
 - skdaccess::framework::data_class::ImageWrapper,
362
 - skdaccess::framework::data_class::SeriesDictionary←
Wrapper, 375
 - skdaccess::framework::data_class::SeriesWrapper,
381
 - skdaccess::framework::data_class::TableWrapper,
395
 - skdaccess::framework::data_class::XArrayWrapper,
402
 - skdaccess::geo::mahali::rinex::data_wrapper::Data←
Wrapper, 350
- get_files_urls
 - skdaccess::utilities::ode_util, 44
- get_query_url
 - skdaccess::utilities::ode_util, 44
- get_raster_array
 - skdaccess::utilities::ode_util, 45
- get_raster_extent
 - skdaccess::utilities::ode_util, 45
- getAllOptions
 - skdaccess::framework::param_class::AutoList, 64
 - skdaccess::framework::param_class::AutoListCycle,
68
 - skdaccess::framework::param_class::AutoList←
Permute, 72
 - skdaccess::framework::param_class::AutoList←
Remove, 77
 - skdaccess::framework::param_class::AutoList←
Subset, 81
- getAntennaLogs
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
262
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 239
- getConfig
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
98
 - skdaccess::astro::spectra::stream::DataFetcher, 270
 - skdaccess::astro::voyager::data_fetcher::Data←
Fetcher, 300
 - skdaccess::engineering::la::generic::stream::Data←
Fetcher, 105
 - skdaccess::engineering::webcam::mit_sailing←
::stream::DataFetcher, 180
 - skdaccess::finance::timeseries::stream::Data←
Fetcher, 292
 - skdaccess::framework::data_class::DataFetcher←
Base, 320
 - skdaccess::framework::data_class::DataFetcher←
Cache, 327
 - skdaccess::framework::data_class::DataFetcher←
Local, 333
 - skdaccess::framework::data_class::DataFetcher←
Storage, 339
 - skdaccess::framework::data_class::DataFetcher←
Stream, 345
 - skdaccess::geo::era_interim::cache::data_fetcher::←
DataFetcher, 255
 - skdaccess::geo::gldas::data_fetcher::DataFetcher,
112
 - skdaccess::geo::grace::data_fetcher::DataFetcher,
276
 - skdaccess::geo::grace::mascon::cache::data_←
fetcher::DataFetcher, 221
 - skdaccess::geo::groundwater::data_fetcher::Data←
Fetcher, 153
 - skdaccess::geo::imsdnhs::data_fetcher::Data←
Fetcher, 247
 - skdaccess::geo::magnetometer::data_fetcher::←
DataFetcher, 128
 - skdaccess::geo::mahali::rinex::data_fetcher::Data←
Fetcher, 285
 - skdaccess::geo::mahali::tec::data_fetcher::Data←
Fetcher, 230
 - skdaccess::geo::mahali::temperature::data_fetcher←
::DataFetcher, 308
 - skdaccess::geo::modis::cache::data_fetcher::Data←
Fetcher, 191
 - skdaccess::geo::modis::stream::data_fetcher::←
DataFetcher, 212
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
263
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 239
 - skdaccess::geo::sentinel_1::cache::data_fetcher::←
DataFetcher, 120
 - skdaccess::geo::srtm::cache::data_fetcher::Data←
Fetcher, 161
 - skdaccess::geo::uavsar::cache::data_fetcher::Data←

- Fetcher, [171](#)
 - skdaccess::geo::wyoming_sounding::cache::data_↵
fetcher::DataFetcher, [136](#)
 - skdaccess::geo::wyoming_sounding::stream::data_↵
_fetcher::DataFetcher, [145](#)
 - skdaccess::planetary::ode::cache::data_fetcher::↵
DataFetcher, [203](#)
 - skdaccess::solar::sdo::data_fetcher::DataFetcher,
[314](#)
- getConfigItem
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
[98](#)
 - skdaccess::astro::spectra::stream::DataFetcher, [270](#)
 - skdaccess::astro::voyager::data_fetcher::Data↵
Fetcher, [300](#)
 - skdaccess::engineering::la::generic::stream::Data↵
Fetcher, [105](#)
 - skdaccess::engineering::webcam::mit_sailing↵
::stream::DataFetcher, [180](#)
 - skdaccess::finance::timeseries::stream::Data↵
Fetcher, [292](#)
 - skdaccess::framework::data_class::DataFetcher↵
Base, [320](#)
 - skdaccess::framework::data_class::DataFetcher↵
Cache, [327](#)
 - skdaccess::framework::data_class::DataFetcher↵
Local, [333](#)
 - skdaccess::framework::data_class::DataFetcher↵
Storage, [339](#)
 - skdaccess::framework::data_class::DataFetcher↵
Stream, [345](#)
 - skdaccess::geo::era_interim::cache::data_fetcher::↵
DataFetcher, [255](#)
 - skdaccess::geo::gldas::data_fetcher::DataFetcher,
[112](#)
 - skdaccess::geo::grace::data_fetcher::DataFetcher,
[277](#)
 - skdaccess::geo::grace::mascon::cache::data_↵
fetcher::DataFetcher, [221](#)
 - skdaccess::geo::groundwater::data_fetcher::Data↵
Fetcher, [153](#)
 - skdaccess::geo::imsdnhs::data_fetcher::Data↵
Fetcher, [248](#)
 - skdaccess::geo::magnetometer::data_fetcher::↵
DataFetcher, [128](#)
 - skdaccess::geo::mahali::rinex::data_fetcher::Data↵
Fetcher, [285](#)
 - skdaccess::geo::mahali::tec::data_fetcher::Data↵
Fetcher, [230](#)
 - skdaccess::geo::mahali::temperature::data_fetcher↵
::DataFetcher, [309](#)
 - skdaccess::geo::modis::cache::data_fetcher::Data↵
Fetcher, [191](#)
 - skdaccess::geo::modis::stream::data_fetcher::↵
DataFetcher, [213](#)
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
[263](#)
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, [239](#)
 - skdaccess::geo::sentinel_1::cache::data_fetcher::↵
DataFetcher, [120](#)
 - skdaccess::geo::srtm::cache::data_fetcher::Data↵
Fetcher, [161](#)
 - skdaccess::geo::uavsar::cache::data_fetcher::Data↵
Fetcher, [171](#)
 - skdaccess::geo::wyoming_sounding::cache::data_↵
fetcher::DataFetcher, [136](#)
 - skdaccess::geo::wyoming_sounding::stream::data_↵
_fetcher::DataFetcher, [145](#)
 - skdaccess::planetary::ode::cache::data_fetcher::↵
DataFetcher, [203](#)
 - skdaccess::solar::sdo::data_fetcher::DataFetcher,
[315](#)
- getDataLocation
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
[99](#)
 - skdaccess::astro::voyager::data_fetcher::Data↵
Fetcher, [301](#)
 - skdaccess::framework::data_class::DataFetcher↵
Cache, [327](#)
 - skdaccess::framework::data_class::DataFetcher↵
Local, [333](#)
 - skdaccess::framework::data_class::DataFetcher↵
Storage, [339](#)
 - skdaccess::geo::era_interim::cache::data_fetcher::↵
DataFetcher, [256](#)
 - skdaccess::geo::gldas::data_fetcher::DataFetcher,
[113](#)
 - skdaccess::geo::grace::data_fetcher::DataFetcher,
[277](#)
 - skdaccess::geo::grace::mascon::cache::data_↵
fetcher::DataFetcher, [222](#)
 - skdaccess::geo::groundwater::data_fetcher::Data↵
Fetcher, [153](#)
 - skdaccess::geo::imsdnhs::data_fetcher::Data↵
Fetcher, [248](#)
 - skdaccess::geo::mahali::rinex::data_fetcher::Data↵
Fetcher, [285](#)
 - skdaccess::geo::mahali::tec::data_fetcher::Data↵
Fetcher, [231](#)
 - skdaccess::geo::modis::cache::data_fetcher::Data↵
Fetcher, [192](#)
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
[263](#)
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, [240](#)
 - skdaccess::geo::sentinel_1::cache::data_fetcher::↵
DataFetcher, [121](#)
 - skdaccess::geo::srtm::cache::data_fetcher::Data↵
Fetcher, [162](#)

- skdaccess::geo::uavsar::cache::data_fetcher::Data↔
Fetcher, 171
- skdaccess::geo::wyoming_sounding::cache::data_↔
fetcher::DataFetcher, 137
- skdaccess::planetary::ode::cache::data_fetcher::↔
DataFetcher, 204
- getDataMetadata
 - skdaccess::geo::magnetometer::data_fetcher::↔
DataFetcher, 128
- getDefaultColumns
 - skdaccess::framework::data_class::TableWrapper,
395
- getDefaultErrorColumns
 - skdaccess::framework::data_class::TableWrapper,
395
- getExtents
 - skdaccess::utilities::image_util::LinearGeolocation,
370
- getExtentsFromCentersPlateCarree
 - skdaccess::utilities::image_util, 35
- getFileIDs
 - skdaccess::utilities::modis_util, 40
- getFileURLs
 - skdaccess::utilities::modis_util, 41
- getGeoTransform
 - skdaccess::utilities::image_util, 35
- getHDFStorage
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
99
 - skdaccess::astro::voyager::data_fetcher::Data↔
Fetcher, 301
 - skdaccess::framework::data_class::DataFetcher↔
Cache, 328
 - skdaccess::geo::era_interim::cache::data_fetcher::↔
DataFetcher, 256
 - skdaccess::geo::grace::mascon::cache::data_↔
fetcher::DataFetcher, 222
 - skdaccess::geo::mahali::rinex::data_fetcher::Data↔
Fetcher, 286
 - skdaccess::geo::mahali::tec::data_fetcher::Data↔
Fetcher, 231
 - skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, 192
 - skdaccess::geo::sentinel_1::cache::data_fetcher::↔
DataFetcher, 121
 - skdaccess::geo::srtm::cache::data_fetcher::Data↔
Fetcher, 162
 - skdaccess::geo::uavsar::cache::data_fetcher::Data↔
Fetcher, 172
 - skdaccess::geo::wyoming_sounding::cache::data_↔
fetcher::DataFetcher, 137
 - skdaccess::planetary::ode::cache::data_fetcher::↔
DataFetcher, 204
- getImageType
 - skdaccess::utilities::modis_util, 41
- getIndices
 - skdaccess::framework::data_class::SeriesDictionary↔
Wrapper, 375
 - skdaccess::framework::data_class::SeriesWrapper,
381
- getInfo
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 240
- getIterator
 - skdaccess::framework::data_class::DataWrapper↔
Base, 356
 - skdaccess::framework::data_class::ImageWrapper,
363
 - skdaccess::framework::data_class::SeriesDictionary↔
Wrapper, 375
 - skdaccess::framework::data_class::SeriesWrapper,
382
 - skdaccess::framework::data_class::TableWrapper,
395
 - skdaccess::framework::data_class::XArrayWrapper,
402
 - skdaccess::geo::mahali::rinex::data_wrapper::Data↔
Wrapper, 350
- getLatLon
 - skdaccess::utilities::image_util::LinearGeolocation,
370
- getLatLonRange
 - skdaccess::utilities::pbo_util, 47
- getLength
 - skdaccess::framework::data_class::SeriesDictionary↔
Wrapper, 375
 - skdaccess::framework::data_class::SeriesWrapper,
382
 - skdaccess::framework::data_class::TableWrapper,
396
- getMasconPlacement
 - skdaccess::geo::grace::mascon::cache::data_↔
fetcher::DataFetcher, 222
- getMetadata
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
99
 - skdaccess::astro::spectra::stream::DataFetcher, 271
 - skdaccess::astro::voyager::data_fetcher::Data↔
Fetcher, 301
 - skdaccess::engineering::la::generic::stream::Data↔
Fetcher, 105
 - skdaccess::engineering::webcam::mit_sailing↔
::stream::DataFetcher, 181
 - skdaccess::finance::timeseries::stream::Data↔
Fetcher, 292
 - skdaccess::framework::data_class::DataFetcher↔
Base, 321
 - skdaccess::framework::data_class::DataFetcher↔
Cache, 328

- skdaccess::framework::data_class::DataFetcher↔
 - Local, [334](#)
- skdaccess::framework::data_class::DataFetcher↔
 - Storage, [340](#)
- skdaccess::framework::data_class::DataFetcher↔
 - Stream, [345](#)
- skdaccess::geo::era_interim::cache::data_fetcher::↔
 - DataFetcher, [256](#)
- skdaccess::geo::gldas::data_fetcher::DataFetcher, [113](#)
- skdaccess::geo::grace::data_fetcher::DataFetcher, [278](#)
- skdaccess::geo::grace::mascon::cache::data_↔
 - fetcher::DataFetcher, [223](#)
- skdaccess::geo::groundwater::data_fetcher::Data↔
 - Fetcher, [154](#)
- skdaccess::geo::imsdnh::data_fetcher::Data↔
 - Fetcher, [248](#)
- skdaccess::geo::magnetometer::data_fetcher::↔
 - DataFetcher, [129](#)
- skdaccess::geo::mahali::rinex::data_fetcher::Data↔
 - Fetcher, [286](#)
- skdaccess::geo::mahali::tec::data_fetcher::Data↔
 - Fetcher, [231](#)
- skdaccess::geo::mahali::temperature::data_fetcher↔
 - ::DataFetcher, [309](#)
- skdaccess::geo::modis::cache::data_fetcher::Data↔
 - Fetcher, [193](#)
- skdaccess::geo::modis::stream::data_fetcher::↔
 - DataFetcher, [213](#)
- skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, [264](#)
- skdaccess::geo::pbo::data_fetcher::DataFetcher, [240](#)
- skdaccess::geo::sentinel_1::cache::data_fetcher::↔
 - DataFetcher, [121](#)
- skdaccess::geo::srtm::cache::data_fetcher::Data↔
 - Fetcher, [162](#)
- skdaccess::geo::uavsar::cache::data_fetcher::Data↔
 - Fetcher, [172](#)
- skdaccess::geo::wyoming_sounding::cache::data_↔
 - fetcher::DataFetcher, [137](#)
- skdaccess::geo::wyoming_sounding::stream::data↔
 - _fetcher::DataFetcher, [145](#)
- skdaccess::planetary::ode::cache::data_fetcher::↔
 - DataFetcher, [205](#)
- skdaccess::solar::sdo::data_fetcher::DataFetcher, [315](#)
- getMetadataFiles
 - skdaccess::astro::voyager::data_fetcher::Data↔
 - Fetcher, [302](#)
- getModisData
 - skdaccess::utilities::modis_util, [42](#)
- getPixelYX
 - skdaccess::utilities::image_util::AffineGlobalCoords, [60](#)
- getProjectedYX
 - skdaccess::utilities::image_util::AffineGlobalCoords, [60](#)
- getROIstations
 - skdaccess::utilities::pbo_util, [48](#)
- getResults
 - skdaccess::framework::data_class::DataWrapper↔
 - Base, [356](#)
 - skdaccess::framework::data_class::ImageWrapper, [363](#)
 - skdaccess::framework::data_class::SeriesDictionary↔
 - Wrapper, [376](#)
 - skdaccess::framework::data_class::SeriesWrapper, [382](#)
 - skdaccess::framework::data_class::TableWrapper, [396](#)
 - skdaccess::framework::data_class::XArrayWrapper, [402](#)
 - skdaccess::geo::mahali::rinex::data_wrapper::Data↔
 - Wrapper, [351](#)
- getRunID
 - skdaccess::framework::data_class::DataWrapper↔
 - Base, [356](#)
 - skdaccess::framework::data_class::ImageWrapper, [363](#)
 - skdaccess::framework::data_class::SeriesDictionary↔
 - Wrapper, [376](#)
 - skdaccess::framework::data_class::SeriesWrapper, [382](#)
 - skdaccess::framework::data_class::TableWrapper, [396](#)
 - skdaccess::framework::data_class::XArrayWrapper, [402](#)
 - skdaccess::geo::mahali::rinex::data_wrapper::Data↔
 - Wrapper, [351](#)
- getSRTMData
 - skdaccess::utilities::srtm_util, [53](#)
- getSRTMLatLon
 - skdaccess::utilities::srtm_util, [53](#)
- getStartDate
 - skdaccess::utilities::grace_util, [32](#)
- getStationCoords
 - skdaccess::utilities::pbo_util, [48](#)
- getStationMetadata
 - skdaccess::geo::groundwater::data_fetcher::Data↔
 - Fetcher, [154](#)
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, [264](#)
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, [240](#)
- getYX
 - skdaccess::utilities::image_util::LinearGeolocation, [371](#)
- grid

- skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, [196](#)
- skdaccess::geo::modis::stream::data_fetcher::↔
DataFetcher, [216](#)
- grid_fill
 - skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, [196](#)
 - skdaccess::geo::modis::stream::data_fetcher::↔
DataFetcher, [216](#)
- handle_data
 - skdaccess::utilities::sounding_util::SoundingParser,
[386](#)
- handle_endtag
 - skdaccess::utilities::sounding_util::SoundingParser,
[387](#)
- handle_starttag
 - skdaccess::utilities::sounding_util::SoundingParser,
[387](#)
- in_header
 - skdaccess::utilities::sounding_util::SoundingParser,
[387](#)
- in_pre_tag
 - skdaccess::utilities::sounding_util::SoundingParser,
[388](#)
- index
 - skdaccess::framework::param_class::AutoListCycle,
[69](#)
- index_date_only
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, [244](#)
- index_list
 - skdaccess::framework::data_class::XArrayWrapper,
[404](#)
- info
 - skdaccess::framework::data_class::DataWrapper↔
Base, [356](#)
 - skdaccess::framework::data_class::ImageWrapper,
[363](#)
 - skdaccess::framework::data_class::SeriesDictionary↔
Wrapper, [376](#)
 - skdaccess::framework::data_class::SeriesWrapper,
[383](#)
 - skdaccess::framework::data_class::TableWrapper,
[396](#)
 - skdaccess::framework::data_class::XArrayWrapper,
[403](#)
 - skdaccess::geo::mahali::rinex::data_wrapper::Data↔
Wrapper, [351](#)
- instrument
 - skdaccess::planetary::ode::cache::data_fetcher::↔
DataFetcher, [208](#)
- interval
 - skdaccess::finance::timeseries::stream::Data↔
Fetcher, [295](#)
- skdaccess::geo::magnetometer::data_fetcher::↔
DataFetcher, [132](#)
- join_string
 - skdaccess::utilities::support, [55](#)
- label
 - skdaccess::engineering::la::generic::stream::Data↔
Fetcher, [109](#)
 - skdaccess::utilities::sounding_util::SoundingParser,
[388](#)
- lat_data
 - skdaccess::utilities::modis_util::LatLon, [368](#)
- lat_extents
 - skdaccess::utilities::image_util::LinearGeolocation,
[371](#)
- lat_func
 - skdaccess::utilities::image_util::SplineLatLon, [391](#)
- lat_pixel_size
 - skdaccess::utilities::image_util::LinearGeolocation,
[371](#)
- lat_range
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
[267](#)
- lat_spline
 - skdaccess::utilities::image_util, [36](#)
- lat_tile_end
 - skdaccess::geo::srtm::cache::data_fetcher::Data↔
Fetcher, [166](#)
- lat_tile_start
 - skdaccess::geo::srtm::cache::data_fetcher::Data↔
Fetcher, [166](#)
- len_x
 - skdaccess::utilities::image_util::LinearGeolocation,
[372](#)
- len_y
 - skdaccess::utilities::image_util::LinearGeolocation,
[372](#)
- list_val_list
 - skdaccess::framework::param_class::AutoListCycle,
[69](#)
- llh_url
 - skdaccess::geo::uavsar::cache::data_fetcher::Data↔
Fetcher, [175](#)
- local_paths
 - skdaccess::geo::sentinel_1::cache::data_fetcher::↔
DataFetcher, [124](#)
- lon_data
 - skdaccess::utilities::modis_util::LatLon, [368](#)
- lon_extents
 - skdaccess::utilities::image_util::LinearGeolocation,
[372](#)
- lon_func
 - skdaccess::utilities::image_util::SplineLatLon, [391](#)
- lon_pixel_size

- skdaccess::utilities::image_util::LinearGeolocation, 372
- lon_range
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, 267
- lon_spline
 - skdaccess::utilities::image_util, 36
- lon_tile_end
 - skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, 166
- lon_tile_start
 - skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, 166
- mascon_placement_url
 - skdaccess::geo::grace::mascon::cache::data_fetcher::DataFetcher, 226
- mascon_url
 - skdaccess::geo::grace::mascon::cache::data_fetcher::DataFetcher, 226
- mask_water
 - skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, 166
- max_lat
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, 208
- max_ob_time
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, 208
- mdyratio
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, 267
- memmap
 - skdaccess::geo::uavsar::cache::data_fetcher::DataFetcher, 175
- merge_srtm_tiles
 - skdaccess::utilities::srtm_util, 54
- meta_data
 - skdaccess::framework::data_class::DataWrapperBase, 358
 - skdaccess::framework::data_class::ImageWrapper, 365
 - skdaccess::framework::data_class::SeriesDictionaryWrapper, 378
 - skdaccess::framework::data_class::SeriesWrapper, 385
 - skdaccess::framework::data_class::TableWrapper, 399
 - skdaccess::framework::data_class::XArrayWrapper, 404
 - skdaccess::geo::mahali::rinex::data_wrapper::DataWrapper, 353
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 244
- metadata_dict
 - skdaccess::utilities::sounding_util::SoundingParser, 388
- metadata_url_list
 - skdaccess::geo::uavsar::cache::data_fetcher::DataFetcher, 175
- min_lat
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, 208
- min_ob_time
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, 208
- mission
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, 209
- modis_id
 - skdaccess::geo::modis::cache::data_fetcher::DataFetcher, 196
 - skdaccess::geo::modis::stream::data_fetcher::DataFetcher, 217
- modis_identifier
 - skdaccess::geo::modis::cache::data_fetcher::DataFetcher, 196
 - skdaccess::geo::modis::stream::data_fetcher::DataFetcher, 217
- modis_platform
 - skdaccess::geo::modis::cache::data_fetcher::DataFetcher, 196
 - skdaccess::geo::modis::stream::data_fetcher::DataFetcher, 217
- month_list
 - skdaccess::geo::wyoming_sounding::cache::data_fetcher::DataFetcher, 141
 - skdaccess::geo::wyoming_sounding::stream::data_fetcher::DataFetcher, 149
- multirun_enabled
 - skdaccess::astro::kepler::data_fetcher::DataFetcher, 100
 - skdaccess::astro::spectra::stream::DataFetcher, 271
 - skdaccess::astro::voyager::data_fetcher::DataFetcher, 302
 - skdaccess::engineering::la::generic::stream::DataFetcher, 106
 - skdaccess::engineering::webcam::mit_sailing::stream::DataFetcher, 181
 - skdaccess::finance::timeseries::stream::DataFetcher, 293
 - skdaccess::framework::data_class::DataFetcherBase, 321
 - skdaccess::framework::data_class::DataFetcherCache, 328
 - skdaccess::framework::data_class::DataFetcherLocal, 334
 - skdaccess::framework::data_class::DataFetcherStorage, 340

- skdaccess::framework::data_class::DataFetcher↔
Stream, [346](#)
- skdaccess::geo::era_interim::cache::data_fetcher::↔
DataFetcher, [257](#)
- skdaccess::geo::gldas::data_fetcher::DataFetcher,
[113](#)
- skdaccess::geo::grace::data_fetcher::DataFetcher,
[278](#)
- skdaccess::geo::grace::mascon::cache::data_↔
fetcher::DataFetcher, [223](#)
- skdaccess::geo::groundwater::data_fetcher::Data↔
Fetcher, [154](#)
- skdaccess::geo::imsdnhs::data_fetcher::Data↔
Fetcher, [249](#)
- skdaccess::geo::magnetometer::data_fetcher::↔
DataFetcher, [129](#)
- skdaccess::geo::mahali::rinex::data_fetcher::Data↔
Fetcher, [286](#)
- skdaccess::geo::mahali::tec::data_fetcher::Data↔
Fetcher, [232](#)
- skdaccess::geo::mahali::temperature::data_fetcher↔
::DataFetcher, [309](#)
- skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, [193](#)
- skdaccess::geo::modis::stream::data_fetcher::↔
DataFetcher, [213](#)
- skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
[264](#)
- skdaccess::geo::pbo::data_fetcher::DataFetcher, [241](#)
- skdaccess::geo::sentinel_1::cache::data_fetcher::↔
DataFetcher, [122](#)
- skdaccess::geo::srtm::cache::data_fetcher::Data↔
Fetcher, [163](#)
- skdaccess::geo::uavsar::cache::data_fetcher::Data↔
Fetcher, [172](#)
- skdaccess::geo::wyoming_sounding::cache::data_↔
fetcher::DataFetcher, [138](#)
- skdaccess::geo::wyoming_sounding::stream::data↔
_fetcher::DataFetcher, [146](#)
- skdaccess::planetary::ode::cache::data_fetcher::↔
DataFetcher, [205](#)
- skdaccess::solar::sdo::data_fetcher::DataFetcher,
[315](#)
- n
 - skdaccess::framework::param_class::AutoList↔
Remove, [78](#)
 - skdaccess::framework::param_class::AutoParam↔
MinMax, [93](#)
- n_max
 - skdaccess::framework::param_class::AutoParam↔
MinMax, [93](#)
- normalize
 - skdaccess::utilities::kepler_util, [37](#)
- nostab_sys
 - skdaccess::utilities::pbo_util, [49](#)
- number_product_limit
 - skdaccess::planetary::ode::cache::data_fetcher::↔
DataFetcher, [209](#)
- openPandasHDFStoreLocking
 - skdaccess::utilities::file_util, [30](#)
- output
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
[100](#)
 - skdaccess::astro::spectra::stream::DataFetcher, [271](#)
 - skdaccess::astro::voyager::data_fetcher::Data↔
Fetcher, [302](#)
 - skdaccess::engineering::la::generic::stream::Data↔
Fetcher, [106](#)
 - skdaccess::engineering::webcam::mit_sailing↔
::stream::DataFetcher, [181](#)
 - skdaccess::finance::timeseries::stream::Data↔
Fetcher, [293](#)
 - skdaccess::framework::data_class::DataFetcher↔
Base, [321](#)
 - skdaccess::framework::data_class::DataFetcher↔
Cache, [329](#)
 - skdaccess::framework::data_class::DataFetcher↔
Local, [334](#)
 - skdaccess::framework::data_class::DataFetcher↔
Storage, [340](#)
 - skdaccess::framework::data_class::DataFetcher↔
Stream, [346](#)
 - skdaccess::geo::era_interim::cache::data_fetcher::↔
DataFetcher, [257](#)
 - skdaccess::geo::gldas::data_fetcher::DataFetcher,
[114](#)
 - skdaccess::geo::grace::data_fetcher::DataFetcher,
[278](#)
 - skdaccess::geo::grace::mascon::cache::data_↔
fetcher::DataFetcher, [223](#)
 - skdaccess::geo::groundwater::data_fetcher::Data↔
Fetcher, [154](#)
 - skdaccess::geo::imsdnhs::data_fetcher::Data↔
Fetcher, [249](#)
 - skdaccess::geo::magnetometer::data_fetcher::↔
DataFetcher, [129](#)
 - skdaccess::geo::mahali::rinex::data_fetcher::Data↔
Fetcher, [287](#)
 - skdaccess::geo::mahali::tec::data_fetcher::Data↔
Fetcher, [232](#)
 - skdaccess::geo::mahali::temperature::data_fetcher↔
::DataFetcher, [310](#)
 - skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, [193](#)
 - skdaccess::geo::modis::stream::data_fetcher::↔
DataFetcher, [214](#)

- skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, 265
- skdaccess::geo::pbo::data_fetcher::DataFetcher, 241
- skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, 122
- skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, 163
- skdaccess::geo::uavsar::cache::data_fetcher::DataFetcher, 173
- skdaccess::geo::wyoming_sounding::cache::data_fetcher::DataFetcher, 138
- skdaccess::geo::wyoming_sounding::stream::data_fetcher::DataFetcher, 146
- skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, 205
- skdaccess::solar::sdo::data_fetcher::DataFetcher, 316
- pandas_kwargs
 - skdaccess::engineering::la::generic::stream::DataFetcher, 109
- parameters
 - skdaccess::engineering::la::generic::stream::DataFetcher, 109
- parselonoFile
 - skdaccess::utilities::mahali_util, 38
- parseSatelliteData
 - skdaccess::utilities::sentinel_1_util, 51
- parseVoyagerData
 - skdaccess::astro::voyager::data_fetcher::DataFetcher, 303
- parseVoyagerMetadata
 - skdaccess::astro::voyager::data_fetcher::DataFetcher, 303
- password
 - skdaccess::geo::era_interim::cache::data_fetcher::DataFetcher, 259
 - skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, 124
 - skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, 166
- path
 - skdaccess::utilities::file_browser::FileBrowser, 360
- perturb
 - skdaccess::astro::kepler::data_fetcher::DataFetcher, 100
 - skdaccess::astro::spectra::stream::DataFetcher, 271
 - skdaccess::astro::voyager::data_fetcher::DataFetcher, 304
 - skdaccess::engineering::la::generic::stream::DataFetcher, 106
 - skdaccess::engineering::webcam::mit_sailing::stream::DataFetcher, 181
 - skdaccess::finance::timeseries::stream::DataFetcher, 293
 - skdaccess::framework::data_class::DataFetcherBase, 321
 - skdaccess::framework::data_class::DataFetcherCache, 329
 - skdaccess::framework::data_class::DataFetcherLocal, 334
 - skdaccess::framework::data_class::DataFetcherStorage, 341
 - skdaccess::framework::data_class::DataFetcherStream, 346
 - skdaccess::framework::param_class::AutoList, 64
 - skdaccess::framework::param_class::AutoListCycle, 68
 - skdaccess::framework::param_class::AutoListPermute, 73
 - skdaccess::framework::param_class::AutoListRemove, 77
 - skdaccess::framework::param_class::AutoListSubset, 81
 - skdaccess::framework::param_class::AutoParam, 84
 - skdaccess::framework::param_class::AutoParamList, 87
 - skdaccess::framework::param_class::AutoParamListCycle, 89
 - skdaccess::framework::param_class::AutoParamMinMax, 92
 - skdaccess::geo::era_interim::cache::data_fetcher::DataFetcher, 257
 - skdaccess::geo::gldas::data_fetcher::DataFetcher, 114
 - skdaccess::geo::grace::data_fetcher::DataFetcher, 278
 - skdaccess::geo::grace::mascon::cache::data_fetcher::DataFetcher, 223
 - skdaccess::geo::groundwater::data_fetcher::DataFetcher, 155
 - skdaccess::geo::imsdnhs::data_fetcher::DataFetcher, 249
 - skdaccess::geo::magnetometer::data_fetcher::DataFetcher, 129
 - skdaccess::geo::mahali::rinex::data_fetcher::DataFetcher, 287
 - skdaccess::geo::mahali::tec::data_fetcher::DataFetcher, 232
 - skdaccess::geo::mahali::temperature::data_fetcher::DataFetcher, 310
 - skdaccess::geo::modis::cache::data_fetcher::DataFetcher, 193
 - skdaccess::geo::modis::stream::data_fetcher::DataFetcher, 214
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, 265

- skdaccess::geo::pbo::data_fetcher::DataFetcher, [241](#)
- skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, [122](#)
- skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, [163](#)
- skdaccess::geo::uavsar::cache::data_fetcher::DataFetcher, [173](#)
- skdaccess::geo::wyoming_sounding::cache::data_fetcher::DataFetcher, [138](#)
- skdaccess::geo::wyoming_sounding::stream::data_fetcher::DataFetcher, [146](#)
- skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, [205](#)
- skdaccess::solar::sdo::data_fetcher::DataFetcher, [316](#)
- planetary/ode/cache/data_fetcher.py, [410](#)
- polarization
 - skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, [125](#)
- possible_data_types
 - skdaccess::finance::timeseries::stream::DataFetcher, [295](#)
- possible_intervals
 - skdaccess::finance::timeseries::stream::DataFetcher, [296](#)
- product_id
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, [209](#)
- product_type
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, [209](#)
- progress_bar
 - skdaccess::utilities::support, [55](#)
- propagateErrors
 - skdaccess::utilities::pbo_util, [49](#)
- quarter_list
 - skdaccess::astro::kepler::data_fetcher::DataFetcher, [102](#)
- query_files_urls
 - skdaccess::utilities::ode_util, [46](#)
- query_yes_no
 - skdaccess::utilities::ode_util, [47](#)
- read_data
 - skdaccess::utilities::sounding_util::SoundingParser, [388](#)
- readMODISData
 - skdaccess::utilities::modis_util, [42](#)
- readTellusData
 - skdaccess::utilities::grace_util, [32](#)
- readUAVSARMetadata
 - skdaccess::utilities::uavsar_util, [56](#)
- remove_ndv
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, [209](#)
- removeAntennaOffset
 - skdaccess::utilities::pbo_util, [50](#)
- removeFrames
 - skdaccess::framework::data_class::TableWrapper, [397](#)
- resample
 - skdaccess::geo::gldas::data_fetcher::DataFetcher, [116](#)
- rescale
 - skdaccess::utilities::modis_util, [42](#)
- reset
 - skdaccess::astro::kepler::data_fetcher::DataFetcher, [100](#)
 - skdaccess::astro::spectra::stream::DataFetcher, [272](#)
 - skdaccess::astro::voyager::data_fetcher::DataFetcher, [304](#)
 - skdaccess::engineering::la::generic::stream::DataFetcher, [106](#)
 - skdaccess::engineering::webcam::mit_sailing::stream::DataFetcher, [182](#)
 - skdaccess::finance::timeseries::stream::DataFetcher, [293](#)
 - skdaccess::framework::data_class::DataFetcherBase, [322](#)
 - skdaccess::framework::data_class::DataFetcherCache, [329](#)
 - skdaccess::framework::data_class::DataFetcherLocal, [335](#)
 - skdaccess::framework::data_class::DataFetcherStorage, [341](#)
 - skdaccess::framework::data_class::DataFetcherStream, [346](#)
 - skdaccess::framework::data_class::DataWrapperBase, [357](#)
 - skdaccess::framework::data_class::ImageWrapper, [364](#)
 - skdaccess::framework::data_class::SeriesDictionaryWrapper, [376](#)
 - skdaccess::framework::data_class::SeriesWrapper, [383](#)
 - skdaccess::framework::data_class::TableWrapper, [397](#)
 - skdaccess::framework::data_class::XArrayWrapper, [403](#)
 - skdaccess::framework::param_class::AutoList, [64](#)
 - skdaccess::framework::param_class::AutoListCycle, [69](#)
 - skdaccess::framework::param_class::AutoListPermute, [73](#)
 - skdaccess::framework::param_class::AutoListRemove, [77](#)
 - skdaccess::framework::param_class::AutoList

- Subset, [81](#)
- skdaccess::framework::param_class::AutoParam, [84](#)
- skdaccess::framework::param_class::AutoParamList, [87](#)
- skdaccess::framework::param_class::AutoParam↔ListCycle, [90](#)
- skdaccess::framework::param_class::AutoParam↔MinMax, [93](#)
- skdaccess::geo::era_interim::cache::data_fetcher::↔DataFetcher, [257](#)
- skdaccess::geo::gldas::data_fetcher::DataFetcher, [114](#)
- skdaccess::geo::grace::data_fetcher::DataFetcher, [278](#)
- skdaccess::geo::grace::mascon::cache::data_↔fetcher::DataFetcher, [224](#)
- skdaccess::geo::groundwater::data_fetcher::Data↔Fetcher, [155](#)
- skdaccess::geo::imsdnhs::data_fetcher::Data↔Fetcher, [249](#)
- skdaccess::geo::magnetometer::data_fetcher::↔DataFetcher, [130](#)
- skdaccess::geo::mahali::rinex::data_fetcher::Data↔Fetcher, [287](#)
- skdaccess::geo::mahali::rinex::data_wrapper::Data↔Wrapper, [351](#)
- skdaccess::geo::mahali::tec::data_fetcher::Data↔Fetcher, [232](#)
- skdaccess::geo::mahali::temperature::data_fetcher↔::DataFetcher, [310](#)
- skdaccess::geo::modis::cache::data_fetcher::Data↔Fetcher, [194](#)
- skdaccess::geo::modis::stream::data_fetcher::↔DataFetcher, [214](#)
- skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, [265](#)
- skdaccess::geo::pbo::data_fetcher::DataFetcher, [241](#)
- skdaccess::geo::sentinel_1::cache::data_fetcher::↔DataFetcher, [122](#)
- skdaccess::geo::srtm::cache::data_fetcher::Data↔Fetcher, [163](#)
- skdaccess::geo::uavsar::cache::data_fetcher::Data↔Fetcher, [173](#)
- skdaccess::geo::wyoming_sounding::cache::data_↔fetcher::DataFetcher, [138](#)
- skdaccess::geo::wyoming_sounding::stream::data_↔fetcher::DataFetcher, [147](#)
- skdaccess::planetary::ode::cache::data_fetcher::↔DataFetcher, [206](#)
- skdaccess::solar::sdo::data_fetcher::DataFetcher, [316](#)
- result_offset_number
 - skdaccess::planetary::ode::cache::data_fetcher::↔DataFetcher, [209](#)
- results
 - skdaccess::framework::data_class::DataWrapper↔Base, [358](#)
 - skdaccess::framework::data_class::ImageWrapper, [365](#)
 - skdaccess::framework::data_class::SeriesDictionary↔Wrapper, [378](#)
 - skdaccess::framework::data_class::SeriesWrapper, [385](#)
 - skdaccess::framework::data_class::TableWrapper, [399](#)
 - skdaccess::framework::data_class::XArrayWrapper, [405](#)
 - skdaccess::geo::mahali::rinex::data_wrapper::Data↔Wrapper, [353](#)
- retrieveCommonDatesHDF
 - skdaccess::utilities::support, [56](#)
- retrieveOnlineData
 - skdaccess::astro::spectra::stream::DataFetcher, [272](#)
 - skdaccess::engineering::la::generic::stream::Data↔Fetcher, [107](#)
 - skdaccess::engineering::webcam::mit_sailing↔::stream::DataFetcher, [182](#)
 - skdaccess::finance::timeseries::stream::Data↔Fetcher, [293](#)
 - skdaccess::framework::data_class::DataFetcher↔Stream, [346](#)
 - skdaccess::geo::magnetometer::data_fetcher::↔DataFetcher, [130](#)
 - skdaccess::geo::mahali::temperature::data_fetcher↔::DataFetcher, [310](#)
 - skdaccess::geo::modis::stream::data_fetcher::↔DataFetcher, [214](#)
 - skdaccess::geo::wyoming_sounding::stream::data_↔fetcher::DataFetcher, [147](#)
 - skdaccess::solar::sdo::data_fetcher::DataFetcher, [316](#)
- run_id
 - skdaccess::framework::data_class::DataWrapper↔Base, [358](#)
 - skdaccess::framework::data_class::ImageWrapper, [366](#)
 - skdaccess::framework::data_class::SeriesDictionary↔Wrapper, [378](#)
 - skdaccess::framework::data_class::SeriesWrapper, [385](#)
 - skdaccess::framework::data_class::TableWrapper, [400](#)
 - skdaccess::framework::data_class::XArrayWrapper, [405](#)
 - skdaccess::geo::mahali::rinex::data_wrapper::Data↔Wrapper, [353](#)
- satellite_url_list

- skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, 125
- scale_factor_url
 - skdaccess::geo::grace::mascon::cache::data_fetcher::DataFetcher, 226
- setDataLocation
 - skdaccess::astro::kepler::data_fetcher::DataFetcher, 101
 - skdaccess::astro::voyager::data_fetcher::DataFetcher, 304
 - skdaccess::framework::data_class::DataFetcherCache, 329
 - skdaccess::framework::data_class::DataFetcherLocal, 335
 - skdaccess::framework::data_class::DataFetcherStorage, 341
 - skdaccess::geo::era_interim::cache::data_fetcher::DataFetcher, 257
 - skdaccess::geo::gldas::data_fetcher::DataFetcher, 114
 - skdaccess::geo::grace::data_fetcher::DataFetcher, 279
 - skdaccess::geo::grace::mascon::cache::data_fetcher::DataFetcher, 224
 - skdaccess::geo::groundwater::data_fetcher::DataFetcher, 155
 - skdaccess::geo::imsdnhs::data_fetcher::DataFetcher, 249
 - skdaccess::geo::mahali::rinex::data_fetcher::DataFetcher, 287
 - skdaccess::geo::mahali::tec::data_fetcher::DataFetcher, 232
 - skdaccess::geo::modis::cache::data_fetcher::DataFetcher, 194
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher, 265
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 242
 - skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, 123
 - skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, 164
 - skdaccess::geo::uavsar::cache::data_fetcher::DataFetcher, 173
 - skdaccess::geo::wyoming_sounding::cache::data_fetcher::DataFetcher, 139
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, 206
- setStationList
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 242
- skdaccess, 13
 - skdaccess.astro, 13
 - skdaccess.astro.kepler, 13
 - skdaccess.astro.kepler.data_fetcher, 14
 - skdaccess.astro.kepler.DataFetcher, 94
 - skdaccess.astro.spectra, 14
 - skdaccess.astro.spectra.stream, 14
 - skdaccess.astro.spectra.stream.DataFetcher, 268
 - skdaccess.astro.voyager, 14
 - skdaccess.astro.voyager.data_fetcher, 14
 - skdaccess.astro.voyager.DataFetcher, 296
 - skdaccess.engineering, 14
 - skdaccess.engineering.la, 15
 - skdaccess.engineering.la.generic, 15
 - skdaccess.engineering.la.generic.stream, 15
 - skdaccess.engineering.la.generic.stream.DataFetcher, 103
 - skdaccess.engineering.la.traffic_counts, 15
 - skdaccess.engineering.la.traffic_counts.stream, 15
 - skdaccess.engineering.la.traffic_counts.stream.DataFetcher, 142
 - skdaccess.engineering.webcam, 15
 - skdaccess.engineering.webcam.mit_sailing, 16
 - skdaccess.engineering.webcam.mit_sailing.stream, 16
 - skdaccess.engineering.webcam.mit_sailing.stream.DataFetcher, 178
 - skdaccess.finance, 16
 - skdaccess.finance.timeseries, 16
 - skdaccess.finance.timeseries.stream, 16
 - skdaccess.finance.timeseries.stream.DataFetcher, 290
 - skdaccess.framework, 16
 - skdaccess.framework.data_class, 17
 - skdaccess.framework.data_class.DataFetcherBase, 318
 - skdaccess.framework.data_class.DataFetcherCache, 324
 - skdaccess.framework.data_class.DataFetcherLocal, 331
 - skdaccess.framework.data_class.DataFetcherStorage, 337
 - skdaccess.framework.data_class.DataFetcherStream, 343
 - skdaccess.framework.data_class.DataWrapperBase, 353
 - skdaccess.framework.data_class.ImageWrapper, 360
 - skdaccess.framework.data_class.SeriesDictionaryWrapper, 373
 - skdaccess.framework.data_class.SeriesWrapper, 379
 - skdaccess.framework.data_class.TableWrapper, 392
 - skdaccess.framework.data_class.XArrayWrapper, 400
 - skdaccess.framework.param_class, 17
 - skdaccess.framework.param_class.AutoList, 61
 - skdaccess.framework.param_class.AutoListCycle, 65
 - skdaccess.framework.param_class.AutoListPermute, 70
 - skdaccess.framework.param_class.AutoListRemove, 74
 - skdaccess.framework.param_class.AutoListSubset, 78
 - skdaccess.framework.param_class.AutoParam, 82
 - skdaccess.framework.param_class.AutoParamList, 85
 - skdaccess.framework.param_class.AutoParamListCycle, 88
 - skdaccess.framework.param_class.AutoParamMinMax, 91
 - skdaccess.geo, 18

- skdaccess.geo.era_interim, 18
- skdaccess.geo.era_interim.cache, 18
- skdaccess.geo.era_interim.cache.data_fetcher, 18
- skdaccess.geo.era_interim.cache.DataFetcher, 252
- skdaccess.geo.gldas, 18
- skdaccess.geo.gldas.data_fetcher, 19
- skdaccess.geo.gldas.DataFetcher, 110
- skdaccess.geo.grace, 19
- skdaccess.geo.grace.data_fetcher, 19
- skdaccess.geo.grace.DataFetcher, 274
- skdaccess.geo.grace.mascon, 19
- skdaccess.geo.grace.mascon.cache, 19
- skdaccess.geo.grace.mascon.cache.data_fetcher, 19
- skdaccess.geo.grace.mascon.cache.DataFetcher, 218
- skdaccess.geo.groundwater, 20
- skdaccess.geo.groundwater.data_fetcher, 20
- skdaccess.geo.groundwater.DataFetcher, 150
- skdaccess.geo.imsdnhs, 20
- skdaccess.geo.imsdnhs.data_fetcher, 20
- skdaccess.geo.imsdnhs.DataFetcher, 245
- skdaccess.geo.magnetometer, 20
- skdaccess.geo.magnetometer.data_fetcher, 20
- skdaccess.geo.magnetometer.DataFetcher, 126
- skdaccess.geo.mahali, 21
- skdaccess.geo.mahali.rinex, 21
- skdaccess.geo.mahali.rinex.data_fetcher, 21
- skdaccess.geo.mahali.rinex.data_wrapper, 21
- skdaccess.geo.mahali.rinex.data_wrapper.DataWrapper, 348
- skdaccess.geo.mahali.rinex.DataFetcher, 281
- skdaccess.geo.mahali.tec, 21
- skdaccess.geo.mahali.tec.data_fetcher, 21
- skdaccess.geo.mahali.tec.DataFetcher, 227
- skdaccess.geo.mahali.temperature, 22
- skdaccess.geo.mahali.temperature.data_fetcher, 22
- skdaccess.geo.mahali.temperature.DataFetcher, 307
- skdaccess.geo.modis, 22
- skdaccess.geo.modis.cache, 22
- skdaccess.geo.modis.cache.cloud_mask, 22
- skdaccess.geo.modis.cache.cloud_mask.data_fetcher, 22
- skdaccess.geo.modis.cache.cloud_mask.DataFetcher, 184
- skdaccess.geo.modis.cache.cloud_opacity, 23
- skdaccess.geo.modis.cache.cloud_opacity.data_fetcher, 23
- skdaccess.geo.modis.cache.cloud_opacity.DataFetcher, 176
- skdaccess.geo.modis.cache.data_fetcher, 23
- skdaccess.geo.modis.cache.DataFetcher, 187
- skdaccess.geo.modis.cache.reflectance, 23
- skdaccess.geo.modis.cache.reflectance.data_fetcher, 23
- skdaccess.geo.modis.cache.reflectance.DataFetcher, 185
- skdaccess.geo.modis.stream, 23
- skdaccess.geo.modis.stream.cloud_mask, 24
- skdaccess.geo.modis.stream.cloud_mask.data_fetcher, 24
- skdaccess.geo.modis.stream.cloud_mask.DataFetcher, 198
- skdaccess.geo.modis.stream.cloud_opacity, 24
- skdaccess.geo.modis.stream.cloud_opacity.data_fetcher, 24
- skdaccess.geo.modis.stream.cloud_opacity.DataFetcher, 197
- skdaccess.geo.modis.stream.data_fetcher, 24
- skdaccess.geo.modis.stream.DataFetcher, 210
- skdaccess.geo.modis.stream.reflectance, 24
- skdaccess.geo.modis.stream.reflectance.data_fetcher, 25
- skdaccess.geo.modis.stream.reflectance.DataFetcher, 177
- skdaccess.geo.ngl_gps, 25
- skdaccess.geo.ngl_gps.data_fetcher, 25
- skdaccess.geo.ngl_gps.DataFetcher, 260
- skdaccess.geo.pbo, 25
- skdaccess.geo.pbo.data_fetcher, 25
- skdaccess.geo.pbo.DataFetcher, 235
- skdaccess.geo.sentinel_1, 25
- skdaccess.geo.sentinel_1.cache, 26
- skdaccess.geo.sentinel_1.cache.data_fetcher, 26
- skdaccess.geo.sentinel_1.cache.DataFetcher, 117
- skdaccess.geo.srtm, 26
- skdaccess.geo.srtm.cache, 26
- skdaccess.geo.srtm.cache.data_fetcher, 26
- skdaccess.geo.srtm.cache.DataFetcher, 158
- skdaccess.geo.uavsar, 26
- skdaccess.geo.uavsar.cache, 27
- skdaccess.geo.uavsar.cache.data_fetcher, 27
- skdaccess.geo.uavsar.cache.DataFetcher, 167
- skdaccess.geo.wyoming_sounding, 27
- skdaccess.geo.wyoming_sounding.cache, 27
- skdaccess.geo.wyoming_sounding.cache.data_fetcher, 27
- skdaccess.geo.wyoming_sounding.cache.DataFetcher, 133
- skdaccess.geo.wyoming_sounding.stream, 27
- skdaccess.geo.wyoming_sounding.stream.data_fetcher, 28
- skdaccess.geo.wyoming_sounding.stream.DataFetcher, 143
- skdaccess.planetary, 28
- skdaccess.planetary.ode, 28
- skdaccess.planetary.ode.cache, 28
- skdaccess.planetary.ode.cache.data_fetcher, 28
- skdaccess.planetary.ode.cache.DataFetcher, 200
- skdaccess.solar, 28
- skdaccess.solar.sdo, 29
- skdaccess.solar.sdo.data_fetcher, 29
- skdaccess.solar.sdo.DataFetcher, 313
- skdaccess.utilities, 29

- skdaccess.utilities.file_browser, 29
- skdaccess.utilities.file_browser.FileBrowser, 359
- skdaccess.utilities.file_util, 30
- skdaccess.utilities.grace_util, 30
- skdaccess.utilities.gw_util, 33
- skdaccess.utilities.image_util, 34
- skdaccess.utilities.image_util.AffineGlobalCoords, 59
- skdaccess.utilities.image_util.LinearGeolocation, 369
- skdaccess.utilities.image_util.SplineLatLon, 389
- skdaccess.utilities.kepler_util, 37
- skdaccess.utilities.mahali_util, 37
- skdaccess.utilities.modis_util, 38
- skdaccess.utilities.modis_util.LatLon, 366
- skdaccess.utilities.ode_util, 43
- skdaccess.utilities.pbo_util, 47
- skdaccess.utilities.sentinel_1_util, 51
- skdaccess.utilities.sounding_util, 52
- skdaccess.utilities.sounding_util.SoundingParser, 385
- skdaccess.utilities.srtm_util, 53
- skdaccess.utilities.support, 54
- skdaccess.utilities.uavsar_util, 56
- skdaccess::astro::kepler::data_fetcher::DataFetcher
 - __init__, 96
 - __str__, 96
 - ap_paramList, 102
 - cacheData, 96, 97
 - checkIfDataExists, 97
 - downloadKeplerData, 98
 - getConfig, 98
 - getConfigItem, 98
 - getDataLocation, 99
 - getHDFStorage, 99
 - getMetadata, 99
 - multirun_enabled, 100
 - output, 100
 - perturb, 100
 - quarter_list, 102
 - reset, 100
 - setDataLocation, 101
 - verbose, 102
 - verbose_print, 101
 - writeConfig, 101
 - writeConfigItem, 102
- skdaccess::astro::spectra::stream::DataFetcher
 - __init__, 269
 - __str__, 270
 - ap_paramList, 273
 - getConfig, 270
 - getConfigItem, 270
 - getMetadata, 271
 - multirun_enabled, 271
 - output, 271
 - perturb, 271
 - reset, 272
 - retrieveOnlineData, 272
 - verbose, 274
 - verbose_print, 272
 - writeConfig, 273
 - writeConfigItem, 273
- skdaccess::astro::voyager::data_fetcher::DataFetcher
 - __init__, 298
 - __str__, 298
 - ap_paramList, 305
 - base_url, 306
 - cacheData, 298
 - checkIfDataExists, 299
 - field_names, 306
 - field_widths, 306
 - generateURL, 300
 - getConfig, 300
 - getConfigItem, 300
 - getDataLocation, 301
 - getHDFStorage, 301
 - getMetadata, 301
 - getMetadataFiles, 302
 - multirun_enabled, 302
 - output, 302
 - parseVoyagerData, 303
 - parseVoyagerMetadata, 303
 - perturb, 304
 - reset, 304
 - setDataLocation, 304
 - spacecraft_list, 306
 - verbose, 306
 - verbose_print, 304
 - writeConfig, 305
 - writeConfigItem, 305
 - year_list, 306
- skdaccess::engineering::la::generic::stream::DataFetcher
 - __init__, 104
 - __str__, 105
 - ap_paramList, 108
 - app_token, 108
 - base_url, 108
 - base_url_and_endpoint, 109
 - getConfig, 105
 - getConfigItem, 105
 - getMetadata, 105
 - label, 109
 - multirun_enabled, 106
 - output, 106
 - pandas_kwargs, 109
 - parameters, 109
 - perturb, 106
 - reset, 106
 - retrieveOnlineData, 107
 - verbose, 109
 - verbose_print, 107

- writeConfig, 107
- writeConfigItem, 108
- skdaccess::engineering::la::traffic_counts::stream::DataFetcher
 - __init__, 142
- skdaccess::engineering::webcam::mit_sailing::stream::DataFetcher
 - __init__, 179
 - __str__, 180
 - ap_paramList, 183
 - camera_list, 184
 - getConfig, 180
 - getConfigItem, 180
 - getMetadata, 181
 - multirun_enabled, 181
 - output, 181
 - perturb, 181
 - reset, 182
 - retrieveOnlineData, 182
 - verbose, 184
 - verbose_print, 182
 - writeConfig, 183
 - writeConfigItem, 183
- skdaccess::finance::timeseries::stream::DataFetcher
 - __init__, 291
 - __str__, 291
 - ap_paramList, 295
 - data_type, 295
 - end_date, 295
 - getConfig, 292
 - getConfigItem, 292
 - getMetadata, 292
 - interval, 295
 - multirun_enabled, 293
 - output, 293
 - perturb, 293
 - possible_data_types, 295
 - possible_intervals, 296
 - reset, 293
 - retrieveOnlineData, 293
 - start_date, 296
 - verbose, 296
 - verbose_print, 294
 - writeConfig, 294
 - writeConfigItem, 294
- skdaccess::framework::data_class::DataFetcherBase
 - __init__, 319
 - __str__, 320
 - ap_paramList, 323
 - getConfig, 320
 - getConfigItem, 320
 - getMetadata, 321
 - multirun_enabled, 321
 - output, 321
 - perturb, 321
 - reset, 322
 - verbose, 323
 - verbose_print, 322
 - writeConfig, 322
 - writeConfigItem, 323
- skdaccess::framework::data_class::DataFetcherCache
 - __str__, 325
 - ap_paramList, 331
 - cacheData, 325
 - checkIfDataExists, 326
 - getConfig, 327
 - getConfigItem, 327
 - getDataLocation, 327
 - getHDFStorage, 328
 - getMetadata, 328
 - multirun_enabled, 328
 - output, 329
 - perturb, 329
 - reset, 329
 - setDataLocation, 329
 - verbose, 331
 - verbose_print, 330
 - writeConfig, 330
 - writeConfigItem, 330
- skdaccess::framework::data_class::DataFetcherLocal
 - __str__, 332
 - ap_paramList, 336
 - getConfig, 333
 - getConfigItem, 333
 - getDataLocation, 333
 - getMetadata, 334
 - multirun_enabled, 334
 - output, 334
 - perturb, 334
 - reset, 335
 - setDataLocation, 335
 - verbose, 337
 - verbose_print, 335
 - writeConfig, 336
 - writeConfigItem, 336
- skdaccess::framework::data_class::DataFetcherStorage
 - __str__, 338
 - ap_paramList, 343
 - downloadFullDataset, 338
 - getConfig, 339
 - getConfigItem, 339
 - getDataLocation, 339
 - getMetadata, 340
 - multirun_enabled, 340
 - output, 340
 - perturb, 341
 - reset, 341
 - setDataLocation, 341

- verbose, [343](#)
- verbose_print, [342](#)
- writeConfig, [342](#)
- writeConfigItem, [342](#)
- skdaccess::framework::data_class::DataFetcherStream
 - __str__, [344](#)
 - ap_paramList, [348](#)
 - getConfig, [345](#)
 - getConfigItem, [345](#)
 - getMetadata, [345](#)
 - multirun_enabled, [346](#)
 - output, [346](#)
 - perturb, [346](#)
 - reset, [346](#)
 - retrieveOnlineData, [346](#)
 - verbose, [348](#)
 - verbose_print, [347](#)
 - writeConfig, [347](#)
 - writeConfigItem, [347](#)
- skdaccess::framework::data_class::DataWrapperBase
 - __init__, [354](#)
 - __len__, [355](#)
 - addResult, [355](#)
 - constants, [358](#)
 - data, [358](#)
 - get, [355](#)
 - getIterator, [356](#)
 - getResults, [356](#)
 - getRunID, [356](#)
 - info, [356](#)
 - meta_data, [358](#)
 - reset, [357](#)
 - results, [358](#)
 - run_id, [358](#)
 - update, [357](#)
 - updateMetadata, [357](#)
- skdaccess::framework::data_class::ImageWrapper
 - __len__, [361](#)
 - addResult, [362](#)
 - constants, [365](#)
 - data, [365](#)
 - deleteData, [362](#)
 - get, [362](#)
 - getIterator, [363](#)
 - getResults, [363](#)
 - getRunID, [363](#)
 - info, [363](#)
 - meta_data, [365](#)
 - reset, [364](#)
 - results, [365](#)
 - run_id, [366](#)
 - update, [364](#)
 - updateData, [364](#)
 - updateMetadata, [365](#)
- skdaccess::framework::data_class::SeriesDictionary←
 - Wrapper
 - __len__, [374](#)
 - addResult, [374](#)
 - constants, [377](#)
 - data, [377](#)
 - data_names, [378](#)
 - error_names, [378](#)
 - get, [375](#)
 - getIndices, [375](#)
 - getIterator, [375](#)
 - getLength, [375](#)
 - getResults, [376](#)
 - getRunID, [376](#)
 - info, [376](#)
 - meta_data, [378](#)
 - reset, [376](#)
 - results, [378](#)
 - run_id, [378](#)
 - update, [377](#)
 - updateMetadata, [377](#)
- skdaccess::framework::data_class::SeriesWrapper
 - __init__, [380](#)
 - __len__, [380](#)
 - addResult, [381](#)
 - constants, [384](#)
 - data, [384](#)
 - data_names, [384](#)
 - error_names, [384](#)
 - get, [381](#)
 - getIndices, [381](#)
 - getIterator, [382](#)
 - getLength, [382](#)
 - getResults, [382](#)
 - getRunID, [382](#)
 - info, [383](#)
 - meta_data, [385](#)
 - reset, [383](#)
 - results, [385](#)
 - run_id, [385](#)
 - update, [383](#)
 - updateMetadata, [384](#)
- skdaccess::framework::data_class::TableWrapper
 - __init__, [393](#)
 - __len__, [394](#)
 - addColumn, [394](#)
 - addResult, [394](#)
 - constants, [399](#)
 - data, [399](#)
 - default_columns, [399](#)
 - default_error_columns, [399](#)
 - get, [395](#)
 - getDefaultColumns, [395](#)
 - getDefaultErrorColumns, [395](#)

- getIterator, 395
- getLength, 396
- getResults, 396
- getRunID, 396
- info, 396
- meta_data, 399
- removeFrames, 397
- reset, 397
- results, 399
- run_id, 400
- update, 397
- updateData, 398
- updateFrames, 398
- updateMetadata, 398
- skdaccess::framework::data_class::XArrayWrapper
 - __init__, 401
 - __len__, 401
 - addResult, 401
 - constants, 404
 - data, 404
 - get, 402
 - getIterator, 402
 - getResults, 402
 - getRunID, 402
 - index_list, 404
 - info, 403
 - meta_data, 404
 - reset, 403
 - results, 405
 - run_id, 405
 - update, 403
 - updateMetadata, 404
- skdaccess::framework::param_class::AutoList
 - __call__, 62
 - __getitem__, 62
 - __init__, 62
 - __len__, 63
 - __setitem__, 63
 - __str__, 63
 - getAllOptions, 64
 - perturb, 64
 - reset, 64
 - val, 64
 - val_init, 65
 - val_list, 65
- skdaccess::framework::param_class::AutoListCycle
 - __call__, 67
 - __getitem__, 67
 - __init__, 66
 - __len__, 67
 - __setitem__, 68
 - __str__, 68
 - getAllOptions, 68
 - index, 69
- list_val_list, 69
- perturb, 68
- reset, 69
- val, 69
- val_init, 69
- val_list, 70
- skdaccess::framework::param_class::AutoListPermute
 - __call__, 71
 - __getitem__, 71
 - __len__, 71
 - __setitem__, 72
 - __str__, 72
 - getAllOptions, 72
 - perturb, 73
 - reset, 73
 - val, 73
 - val_init, 73
 - val_list, 74
- skdaccess::framework::param_class::AutoListRemove
 - __call__, 75
 - __getitem__, 75
 - __init__, 75
 - __len__, 76
 - __setitem__, 76
 - __str__, 76
 - getAllOptions, 77
 - n, 78
 - perturb, 77
 - reset, 77
 - val, 77
 - val_init, 78
 - val_list, 78
- skdaccess::framework::param_class::AutoListSubset
 - __call__, 79
 - __getitem__, 79
 - __len__, 80
 - __setitem__, 80
 - __str__, 80
 - getAllOptions, 81
 - perturb, 81
 - reset, 81
 - val, 81
 - val_init, 82
 - val_list, 82
- skdaccess::framework::param_class::AutoParam
 - __call__, 84
 - __init__, 83
 - __str__, 84
 - perturb, 84
 - reset, 84
 - val, 85
 - val_init, 85
- skdaccess::framework::param_class::AutoParamList
 - __call__, 86

- `__init__`, 86
- `__str__`, 86
- `perturb`, 87
- `reset`, 87
- `val`, 87
- `val_init`, 87
- `val_list`, 87
- `skdaccess::framework::param_class::AutoParamListCycle`
 - `__call__`, 89
 - `__init__`, 89
 - `__str__`, 89
 - `current_index`, 90
 - `perturb`, 89
 - `reset`, 90
 - `val`, 90
 - `val_init`, 90
 - `val_list`, 90
- `skdaccess::framework::param_class::AutoParamMinMax`
 - `__call__`, 92
 - `__init__`, 91
 - `__str__`, 92
 - `decimals`, 93
 - `n`, 93
 - `n_max`, 93
 - `perturb`, 92
 - `reset`, 93
 - `val`, 93
 - `val_init`, 93
 - `val_max`, 94
 - `val_min`, 94
- `skdaccess::geo::era_interim::cache::data_fetcher::DataFetcher`
 - `__init__`, 253
 - `__str__`, 254
 - `ap_paramList`, 259
 - `cacheData`, 254
 - `checkIfDataExists`, 254
 - `data_names`, 259
 - `date_list`, 259
 - `getConfig`, 255
 - `getConfigItem`, 255
 - `getDataLocation`, 256
 - `getHDFStorage`, 256
 - `getMetadata`, 256
 - `multirun_enabled`, 257
 - `output`, 257
 - `password`, 259
 - `perturb`, 257
 - `reset`, 257
 - `setDataLocation`, 257
 - `username`, 259
 - `verbose`, 259
 - `verbose_print`, 258
 - `writeConfig`, 258
 - `writeConfigItem`, 258
- `skdaccess::geo::gldas::data_fetcher::DataFetcher`
 - `__init__`, 111
 - `__str__`, 111
 - `ap_paramList`, 116
 - `downloadFullDataset`, 112
 - `end_date`, 116
 - `getConfig`, 112
 - `getConfigItem`, 112
 - `getDataLocation`, 113
 - `getMetadata`, 113
 - `multirun_enabled`, 113
 - `output`, 114
 - `perturb`, 114
 - `resample`, 116
 - `reset`, 114
 - `setDataLocation`, 114
 - `start_date`, 116
 - `verbose`, 116
 - `verbose_print`, 115
 - `writeConfig`, 115
 - `writeConfigItem`, 115
- `skdaccess::geo::grace::data_fetcher::DataFetcher`
 - `__init__`, 275
 - `__str__`, 276
 - `ap_paramList`, 280
 - `downloadFullDataset`, 276
 - `end_date`, 280
 - `getConfig`, 276
 - `getConfigItem`, 277
 - `getDataLocation`, 277
 - `getMetadata`, 278
 - `multirun_enabled`, 278
 - `output`, 278
 - `perturb`, 278
 - `reset`, 278
 - `setDataLocation`, 279
 - `start_date`, 280
 - `verbose`, 281
 - `verbose_print`, 279
 - `writeConfig`, 279
 - `writeConfigItem`, 280
- `skdaccess::geo::grace::mascon::cache::data_fetcher::DataFetcher`
 - `__init__`, 219
 - `__str__`, 220
 - `ap_paramList`, 225
 - `cacheData`, 220
 - `checkIfDataExists`, 221
 - `end_date`, 226
 - `getConfig`, 221
 - `getConfigItem`, 221
 - `getDataLocation`, 222
 - `getHDFStorage`, 222

- getMasconPlacement, 222
- getMetadata, 223
- mascon_placement_url, 226
- mascon_url, 226
- multirun_enabled, 223
- output, 223
- perturb, 223
- reset, 224
- scale_factor_url, 226
- setDataLocation, 224
- start_date, 226
- verbose, 226
- verbose_print, 224
- writeConfig, 225
- writeConfigItem, 225
- skdaccess::geo::groundwater::data_fetcher::DataFetcher
 - __init__, 151
 - __str__, 152
 - ap_paramList, 157
 - cutoff, 157
 - downloadFullDataset, 152
 - end_date, 157
 - getConfig, 153
 - getConfigItem, 153
 - getDataLocation, 153
 - getMetadata, 154
 - getStationMetadata, 154
 - multirun_enabled, 154
 - output, 154
 - perturb, 155
 - reset, 155
 - setDataLocation, 155
 - start_date, 157
 - verbose, 157
 - verbose_print, 156
 - writeConfig, 156
 - writeConfigItem, 156
- skdaccess::geo::imsdnhs::data_fetcher::DataFetcher
 - __init__, 246
 - __str__, 247
 - ap_paramList, 251
 - coordinate_dict, 251
 - downloadFullDataset, 247
 - end_date, 251
 - getConfig, 247
 - getConfigItem, 248
 - getDataLocation, 248
 - getMetadata, 248
 - multirun_enabled, 249
 - output, 249
 - perturb, 249
 - reset, 249
 - setDataLocation, 249
 - start_date, 251
- verbose, 251
- verbose_print, 250
- writeConfig, 250
- writeConfigItem, 250
- skdaccess::geo::magnetometer::data_fetcher::DataFetcher
 - __init__, 127
 - __str__, 127
 - ap_paramList, 131
 - channels, 132
 - data_type, 132
 - end_time, 132
 - getConfig, 128
 - getConfigItem, 128
 - getDataMetadata, 128
 - getMetadata, 129
 - interval, 132
 - multirun_enabled, 129
 - output, 129
 - perturb, 129
 - reset, 130
 - retrieveOnlineData, 130
 - start_time, 132
 - verbose, 132
 - verbose_print, 130
 - writeConfig, 131
 - writeConfigItem, 131
- skdaccess::geo::mahali::rinex::data_fetcher::DataFetcher
 - __init__, 282
 - __str__, 283
 - ap_paramList, 289
 - cacheData, 283
 - checkIfDataExists, 284
 - date_range, 289
 - end_date, 289
 - generate_links, 289
 - getConfig, 285
 - getConfigItem, 285
 - getDataLocation, 285
 - getHDFStorage, 286
 - getMetadata, 286
 - multirun_enabled, 286
 - output, 287
 - perturb, 287
 - reset, 287
 - setDataLocation, 287
 - start_date, 289
 - verbose, 289
 - verbose_print, 288
 - writeConfig, 288
 - writeConfigItem, 288
- skdaccess::geo::mahali::rinex::data_wrapper::DataWrapper
 - __len__, 349

- addResult, [350](#)
- constants, [352](#)
- data, [352](#)
- get, [350](#)
- getIterator, [350](#)
- getResults, [351](#)
- getRunID, [351](#)
- info, [351](#)
- meta_data, [353](#)
- reset, [351](#)
- results, [353](#)
- run_id, [353](#)
- update, [352](#)
- updateMetadata, [352](#)
- skdaccess::geo::mahali::tec::data_fetcher::DataFetcher
 - __init__, [228](#)
 - __str__, [229](#)
 - ap_paramList, [234](#)
 - cacheData, [229](#)
 - checkIfDataExists, [229](#)
 - date_range, [234](#)
 - end_date, [234](#)
 - getConfig, [230](#)
 - getConfigItem, [230](#)
 - getDataLocation, [231](#)
 - getHDFStorage, [231](#)
 - getMetadata, [231](#)
 - multirun_enabled, [232](#)
 - output, [232](#)
 - perturb, [232](#)
 - reset, [232](#)
 - setDataLocation, [232](#)
 - start_date, [234](#)
 - verbose, [234](#)
 - verbose_print, [233](#)
 - writeConfig, [233](#)
 - writeConfigItem, [233](#)
- skdaccess::geo::mahali::temperature::data_fetcher::DataFetcher
 - __init__, [308](#)
 - __str__, [308](#)
 - ap_paramList, [312](#)
 - end_date, [312](#)
 - getConfig, [308](#)
 - getConfigItem, [309](#)
 - getMetadata, [309](#)
 - multirun_enabled, [309](#)
 - output, [310](#)
 - perturb, [310](#)
 - reset, [310](#)
 - retrieveOnlineData, [310](#)
 - start_date, [312](#)
 - verbose, [312](#)
 - verbose_print, [311](#)
 - writeConfig, [311](#)
 - writeConfigItem, [311](#)
- skdaccess::geo::modis::cache::cloud_mask::data_fetcher::DataFetcher
 - __init__, [185](#)
- skdaccess::geo::modis::cache::cloud_opacity::data_fetcher::DataFetcher
 - __init__, [176](#)
- skdaccess::geo::modis::cache::data_fetcher::DataFetcher
 - __init__, [188](#)
 - __str__, [189](#)
 - ap_paramList, [195](#)
 - cacheData, [189](#), [190](#)
 - checkIfDataExists, [190](#)
 - daynightboth, [196](#)
 - end_date, [196](#)
 - find_data, [191](#)
 - getConfig, [191](#)
 - getConfigItem, [191](#)
 - getDataLocation, [192](#)
 - getHDFStorage, [192](#)
 - getMetadata, [193](#)
 - grid, [196](#)
 - grid_fill, [196](#)
 - modis_id, [196](#)
 - modis_identifier, [196](#)
 - modis_platform, [196](#)
 - multirun_enabled, [193](#)
 - output, [193](#)
 - perturb, [193](#)
 - reset, [194](#)
 - setDataLocation, [194](#)
 - start_date, [197](#)
 - use_long_name, [197](#)
 - variable_list, [197](#)
 - verbose, [197](#)
 - verbose_print, [194](#)
 - writeConfig, [195](#)
 - writeConfigItem, [195](#)
- skdaccess::geo::modis::cache::reflectance::data_fetcher::DataFetcher
 - __init__, [186](#)
- skdaccess::geo::modis::stream::cloud_mask::data_fetcher::DataFetcher
 - __init__, [199](#)
- skdaccess::geo::modis::stream::cloud_opacity::data_fetcher::DataFetcher
 - __init__, [198](#)
- skdaccess::geo::modis::stream::data_fetcher::DataFetcher
 - __init__, [212](#)
 - __str__, [212](#)
 - ap_paramList, [216](#)
 - daynightboth, [216](#)

- end_date, 216
- getConfig, 212
- getConfigItem, 213
- getMetadata, 213
- grid, 216
- grid_fill, 216
- modis_id, 217
- modis_identifier, 217
- modis_platform, 217
- multirun_enabled, 213
- output, 214
- perturb, 214
- reset, 214
- retrieveOnlineData, 214
- start_date, 217
- use_long_name, 217
- variable_list, 217
- verbose, 217
- verbose_print, 215
- writeConfig, 215
- writeConfigItem, 215
- skdaccess::geo::modis::stream::reflectance::data_↵
 fetcher::DataFetcher
 __init__, 177
- skdaccess::geo::ngl_gps::data_fetcher::DataFetcher
 __init__, 261
 __str__, 262
 ap_paramList, 267
 data_type, 267
 downloadFullDataset, 262
 end_date, 267
 getAntennaLogs, 262
 getConfig, 263
 getConfigItem, 263
 getDataLocation, 263
 getMetadata, 264
 getStationMetadata, 264
 lat_range, 267
 lon_range, 267
 mdyratio, 267
 multirun_enabled, 264
 output, 265
 perturb, 265
 reset, 265
 setDataLocation, 265
 start_date, 268
 verbose, 268
 verbose_print, 266
 writeConfig, 266
 writeConfigItem, 266
- skdaccess::geo::pbo::data_fetcher::DataFetcher
 __init__, 236
 __str__, 238
 antenna_info, 243
 ap_paramList, 244
 default_columns, 244
 default_error_columns, 244
 downloadFullDataset, 238
 getAntennaLogs, 239
 getConfig, 239
 getConfigItem, 239
 getDataLocation, 240
 getInfo, 240
 getMetadata, 240
 getStationMetadata, 240
 index_date_only, 244
 meta_data, 244
 multirun_enabled, 241
 output, 241
 perturb, 241
 reset, 241
 setDataLocation, 242
 setStationList, 242
 station_list, 244
 use_progress_bar, 244
 verbose, 245
 verbose_print, 242
 writeConfig, 243
 writeConfigItem, 243
- skdaccess::geo::sentinel_1::cache::data_fetcher::Data_↵
 Fetcher
 __init__, 118
 __str__, 119
 ap_paramList, 124
 cacheData, 119
 checkIfDataExists, 120
 getConfig, 120
 getConfigItem, 120
 getDataLocation, 121
 getHDFStorage, 121
 getMetadata, 121
 local_paths, 124
 multirun_enabled, 122
 output, 122
 password, 124
 perturb, 122
 polarization, 125
 reset, 122
 satellite_url_list, 125
 setDataLocation, 123
 swath, 125
 url_list, 125
 username, 125
 verbose, 125
 verbose_print, 123
 writeConfig, 123
 writeConfigItem, 124
- skdaccess::geo::srtm::cache::data_fetcher::DataFetcher

- `__init__`, 159
- `__str__`, 160
- `ap_paramList`, 165
- `arcsecond_sampling`, 165
- `cacheData`, 160
- `checkIfDataExists`, 161
- `getConfig`, 161
- `getConfigItem`, 161
- `getDataLocation`, 162
- `getHDFStorage`, 162
- `getMetadata`, 162
- `lat_tile_end`, 166
- `lat_tile_start`, 166
- `lon_tile_end`, 166
- `lon_tile_start`, 166
- `mask_water`, 166
- `multirun_enabled`, 163
- `output`, 163
- `password`, 166
- `perturb`, 163
- `reset`, 163
- `setDataLocation`, 164
- `store_geolocation_grids`, 167
- `username`, 167
- `verbose`, 167
- `verbose_print`, 164
- `writeConfig`, 164
- `writeConfigItem`, 165
- `skdaccess::geo::uavsar::cache::data_fetcher::DataFetcher`
 - `__init__`, 169
 - `__str__`, 169
 - `ap_paramList`, 175
 - `cacheData`, 169
 - `checkIfDataExists`, 170
 - `getConfig`, 171
 - `getConfigItem`, 171
 - `getDataLocation`, 171
 - `getHDFStorage`, 172
 - `getMetadata`, 172
 - `llh_url`, 175
 - `memmap`, 175
 - `metadata_url_list`, 175
 - `multirun_enabled`, 172
 - `output`, 173
 - `perturb`, 173
 - `reset`, 173
 - `setDataLocation`, 173
 - `slc_url_list`, 175
 - `verbose`, 175
 - `verbose_print`, 174
 - `writeConfig`, 174
 - `writeConfigItem`, 174
- `skdaccess::geo::wyoming_sounding::cache::data_fetcher::DataFetcher`
 - `__init__`, 134
 - `__str__`, 135
 - `ap_paramList`, 140
 - `cacheData`, 135
 - `checkIfDataExists`, 136
 - `day_end`, 140
 - `day_start`, 140
 - `end_hour`, 141
 - `getConfig`, 136
 - `getConfigItem`, 136
 - `getDataLocation`, 137
 - `getHDFStorage`, 137
 - `getMetadata`, 137
 - `month_list`, 141
 - `multirun_enabled`, 138
 - `output`, 138
 - `perturb`, 138
 - `reset`, 138
 - `setDataLocation`, 139
 - `start_hour`, 141
 - `station_number`, 141
 - `verbose`, 141
 - `verbose_print`, 139
 - `writeConfig`, 139
 - `writeConfigItem`, 140
 - `year_list`, 141
- `skdaccess::geo::wyoming_sounding::stream::data_fetcher::DataFetcher`
 - `__init__`, 144
 - `__str__`, 145
 - `ap_paramList`, 148
 - `day_end`, 149
 - `day_start`, 149
 - `end_hour`, 149
 - `getConfig`, 145
 - `getConfigItem`, 145
 - `getMetadata`, 145
 - `month_list`, 149
 - `multirun_enabled`, 146
 - `output`, 146
 - `perturb`, 146
 - `reset`, 147
 - `retrieveOnlineData`, 147
 - `start_hour`, 149
 - `station_number`, 149
 - `verbose`, 149
 - `verbose_print`, 147
 - `writeConfig`, 148
 - `writeConfigItem`, 148
 - `year_list`, 150
- `skdaccess::planetary::ode::cache::data_fetcher::DataFetcher`

- __init__, 201
- __str__, 202
- ap_paramList, 207
- cacheData, 202
- checkIfDataExists, 203
- eastern_lon, 208
- file_name, 208
- getConfig, 203
- getConfigItem, 203
- getDataLocation, 204
- getHDFStorage, 204
- getMetadata, 205
- instrument, 208
- max_lat, 208
- max_ob_time, 208
- min_lat, 208
- min_ob_time, 208
- mission, 209
- multirun_enabled, 205
- number_product_limit, 209
- output, 205
- perturb, 205
- product_id, 209
- product_type, 209
- remove_ndv, 209
- reset, 206
- result_offset_number, 209
- setDataLocation, 206
- target, 209
- verbose, 210
- verbose_print, 206
- western_lon, 210
- writeConfig, 207
- writeConfigItem, 207
- skdaccess::solar::sdo::data_fetcher::DataFetcher
 - __init__, 314
 - __str__, 314
 - ap_paramList, 318
 - getConfig, 314
 - getConfigItem, 315
 - getMetadata, 315
 - multirun_enabled, 315
 - output, 316
 - perturb, 316
 - reset, 316
 - retrieveOnlineData, 316
 - verbose, 318
 - verbose_print, 317
 - writeConfig, 317
 - writeConfigItem, 317
- skdaccess::utilities::file_browser::FileBrowser
 - __init__, 359
 - dirs, 360
 - files, 360
 - path, 360
 - widget, 359
- skdaccess::utilities::file_util
 - openPandasHDFStoreLocking, 30
- skdaccess::utilities::grace_util
 - averageDates, 31
 - computeEWD, 31
 - dateMismatch, 31
 - getStartEndDate, 32
 - readTellusData, 32
- skdaccess::utilities::gw_util
 - combine_water_heights, 33
- skdaccess::utilities::image_util
 - convertBinCentersToEdges, 34
 - getExtentsFromCentersPlateCarree, 35
 - getGeoTransform, 35
 - lat_spline, 36
 - lon_spline, 36
 - SplineGeolocation, 36
 - x_offset, 36
 - x_spline, 36
 - y_offset, 36
 - y_spline, 36
- skdaccess::utilities::image_util::AffineGlobalCoords
 - __init__, 59
 - getPixelYX, 60
 - getProjectedYX, 60
- skdaccess::utilities::image_util::LinearGeolocation
 - __init__, 370
 - flip_y, 371
 - getExtents, 370
 - getLatLon, 370
 - getYX, 371
 - lat_extents, 371
 - lat_pixel_size, 371
 - len_x, 372
 - len_y, 372
 - lon_extents, 372
 - lon_pixel_size, 372
 - start_lat, 372
 - start_lon, 372
 - x_offset, 372
 - y_offset, 373
- skdaccess::utilities::image_util::SplineLatLon
 - __call__, 390
 - __init__, 389
 - lat_func, 391
 - lon_func, 391
 - x_offset, 391
 - y_offset, 391
- skdaccess::utilities::kepler_util
 - normalize, 37
- skdaccess::utilities::mahali_util
 - convert_date, 37

- parselonoFile, 38
- skdaccess::utilities::modis_util
 - calibrateModis, 39
 - checkBit, 39
 - createGrid, 39
 - getFileIDs, 40
 - getFileURLs, 41
 - getImageType, 41
 - getModisData, 42
 - readMODISData, 42
 - rescale, 42
- skdaccess::utilities::modis_util::LatLon
 - __call__, 367
 - __init__, 367
 - alat, 368
 - alon, 368
 - lat_data, 368
 - lon_data, 368
 - x_offset, 368
 - y_offset, 368
- skdaccess::utilities::ode_util
 - correct_CRISM_label, 43
 - correct_file_name_case_in_label, 44
 - correct_label_file, 44
 - get_files_urls, 44
 - get_query_url, 44
 - get_raster_array, 45
 - get_raster_extent, 45
 - query_files_urls, 46
 - query_yes_no, 47
- skdaccess::utilities::pbo_util
 - getLatLonRange, 47
 - getROIStations, 48
 - getStationCoords, 48
 - nostab_sys, 49
 - propagateErrors, 49
 - removeAntennaOffset, 50
 - stab_sys, 50
- skdaccess::utilities::sentinel_1_util
 - parseSatelliteData, 51
- skdaccess::utilities::sounding_util
 - generateQueries, 52
- skdaccess::utilities::sounding_util::SoundingParser
 - __init__, 386
 - data_dict, 387
 - handle_data, 386
 - handle_endtag, 387
 - handle_starttag, 387
 - in_header, 387
 - in_pre_tag, 388
 - label, 388
 - metadata_dict, 388
 - read_data, 388
 - tmp, 388
- skdaccess::utilities::srtm_util
 - getSRTMData, 53
 - getSRTMLatLon, 53
 - merge_srtm_tiles, 54
- skdaccess::utilities::support
 - convertToStr, 55
 - join_string, 55
 - progress_bar, 55
 - retrieveCommonDatesHDF, 56
- skdaccess::utilities::uavsar_util
 - readUAVSARMetadata, 56
- slc_url_list
 - skdaccess::geo::uavsar::cache::data_fetcher::Data←
Fetcher, 175
- solar/sdo/data_fetcher.py, 410
- spacecraft_list
 - skdaccess::astro::voyager::data_fetcher::Data←
Fetcher, 306
- SplineGeolocation
 - skdaccess::utilities::image_util, 36
- stab_sys
 - skdaccess::utilities::pbo_util, 50
- start_date
 - skdaccess::finance::timeseries::stream::Data←
Fetcher, 296
 - skdaccess::geo::gldas::data_fetcher::DataFetcher,
116
 - skdaccess::geo::grace::data_fetcher::DataFetcher,
280
 - skdaccess::geo::grace::mascon::cache::data_←
fetcher::DataFetcher, 226
 - skdaccess::geo::groundwater::data_fetcher::Data←
Fetcher, 157
 - skdaccess::geo::imsdnhs::data_fetcher::Data←
Fetcher, 251
 - skdaccess::geo::mahali::rinex::data_fetcher::Data←
Fetcher, 289
 - skdaccess::geo::mahali::tec::data_fetcher::Data←
Fetcher, 234
 - skdaccess::geo::mahali::temperature::data_fetcher←
::DataFetcher, 312
 - skdaccess::geo::modis::cache::data_fetcher::Data←
Fetcher, 197
 - skdaccess::geo::modis::stream::data_fetcher::←
DataFetcher, 217
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
268
- start_hour
 - skdaccess::geo::wyoming_sounding::cache::data_←
fetcher::DataFetcher, 141
 - skdaccess::geo::wyoming_sounding::stream::data←
_fetcher::DataFetcher, 149
- start_lat
 - skdaccess::utilities::image_util::LinearGeolocation,

- 372
- start_lon
 - skdaccess::utilities::image_util::LinearGeolocation, 372
- start_time
 - skdaccess::geo::magnetometer::data_fetcher::DataFetcher, 132
- station_list
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 244
- station_number
 - skdaccess::geo::wyoming_sounding::cache::data_fetcher::DataFetcher, 141
 - skdaccess::geo::wyoming_sounding::stream::data_fetcher::DataFetcher, 149
- store_geolocation_grids
 - skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, 167
- swath
 - skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, 125
- target
 - skdaccess::planetary::ode::cache::data_fetcher::DataFetcher, 209
- tmp
 - skdaccess::utilities::sounding_util::SoundingParser, 388
- update
 - skdaccess::framework::data_class::DataWrapperBase, 357
 - skdaccess::framework::data_class::ImageWrapper, 364
 - skdaccess::framework::data_class::SeriesDictionaryWrapper, 377
 - skdaccess::framework::data_class::SeriesWrapper, 383
 - skdaccess::framework::data_class::TableWrapper, 397
 - skdaccess::framework::data_class::XArrayWrapper, 403
 - skdaccess::geo::mahali::rinex::data_wrapper::DataWrapper, 352
- updateData
 - skdaccess::framework::data_class::ImageWrapper, 364
 - skdaccess::framework::data_class::TableWrapper, 398
- updateFrames
 - skdaccess::framework::data_class::TableWrapper, 398
- updateMetadata
 - skdaccess::framework::data_class::DataWrapperBase, 357
- skdaccess::framework::data_class::ImageWrapper, 365
- skdaccess::framework::data_class::SeriesDictionaryWrapper, 377
- skdaccess::framework::data_class::SeriesWrapper, 384
- skdaccess::framework::data_class::TableWrapper, 398
- skdaccess::framework::data_class::XArrayWrapper, 404
- skdaccess::geo::mahali::rinex::data_wrapper::DataWrapper, 352
- url_list
 - skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, 125
- use_long_name
 - skdaccess::geo::modis::cache::data_fetcher::DataFetcher, 197
 - skdaccess::geo::modis::stream::data_fetcher::DataFetcher, 217
- use_progress_bar
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 244
- username
 - skdaccess::geo::era_interim::cache::data_fetcher::DataFetcher, 259
 - skdaccess::geo::sentinel_1::cache::data_fetcher::DataFetcher, 125
 - skdaccess::geo::srtm::cache::data_fetcher::DataFetcher, 167
- utilities/file_browser.py, 418
- utilities/file_util.py, 418
- utilities/grace_util.py, 419
- utilities/gw_util.py, 419
- utilities/image_util.py, 419
- utilities/kepler_util.py, 420
- utilities/mahali_util.py, 420
- utilities/modis_util.py, 421
- utilities/ode_util.py, 421
- utilities/pbo_util.py, 422
- utilities/sentinel_1_util.py, 423
- utilities/sounding_util.py, 423
- utilities/srtm_util.py, 423
- utilities/support.py, 424
- utilities/uavsar_util.py, 424
- val
 - skdaccess::framework::param_class::AutoList, 64
 - skdaccess::framework::param_class::AutoListCycle, 69
 - skdaccess::framework::param_class::AutoListPermute, 73
 - skdaccess::framework::param_class::AutoListRemove, 77

- skdaccess::framework::param_class::AutoList↔
Subset, [81](#)
- skdaccess::framework::param_class::AutoParam, [85](#)
- skdaccess::framework::param_class::AutoParamList,
[87](#)
- skdaccess::framework::param_class::AutoParam↔
ListCycle, [90](#)
- skdaccess::framework::param_class::AutoParam↔
MinMax, [93](#)
- val_init
 - skdaccess::framework::param_class::AutoList, [65](#)
 - skdaccess::framework::param_class::AutoListCycle,
[69](#)
 - skdaccess::framework::param_class::AutoList↔
Permute, [73](#)
 - skdaccess::framework::param_class::AutoList↔
Remove, [78](#)
 - skdaccess::framework::param_class::AutoList↔
Subset, [82](#)
 - skdaccess::framework::param_class::AutoParam, [85](#)
 - skdaccess::framework::param_class::AutoParamList,
[87](#)
 - skdaccess::framework::param_class::AutoParam↔
ListCycle, [90](#)
 - skdaccess::framework::param_class::AutoParam↔
MinMax, [93](#)
- val_list
 - skdaccess::framework::param_class::AutoList, [65](#)
 - skdaccess::framework::param_class::AutoListCycle,
[70](#)
 - skdaccess::framework::param_class::AutoList↔
Permute, [74](#)
 - skdaccess::framework::param_class::AutoList↔
Remove, [78](#)
 - skdaccess::framework::param_class::AutoList↔
Subset, [82](#)
 - skdaccess::framework::param_class::AutoParamList,
[87](#)
 - skdaccess::framework::param_class::AutoParam↔
ListCycle, [90](#)
- val_max
 - skdaccess::framework::param_class::AutoParam↔
MinMax, [94](#)
- val_min
 - skdaccess::framework::param_class::AutoParam↔
MinMax, [94](#)
- variable_list
 - skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, [197](#)
 - skdaccess::geo::modis::stream::data_fetcher::↔
DataFetcher, [217](#)
- verbose
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
[102](#)
 - skdaccess::astro::spectra::stream::DataFetcher, [274](#)
 - skdaccess::astro::voyager::data_fetcher::Data↔
Fetcher, [306](#)
 - skdaccess::engineering::la::generic::stream::Data↔
Fetcher, [109](#)
 - skdaccess::engineering::webcam::mit_sailing↔
::stream::DataFetcher, [184](#)
 - skdaccess::finance::timeseries::stream::Data↔
Fetcher, [296](#)
 - skdaccess::framework::data_class::DataFetcher↔
Base, [323](#)
 - skdaccess::framework::data_class::DataFetcher↔
Cache, [331](#)
 - skdaccess::framework::data_class::DataFetcher↔
Local, [337](#)
 - skdaccess::framework::data_class::DataFetcher↔
Storage, [343](#)
 - skdaccess::framework::data_class::DataFetcher↔
Stream, [348](#)
 - skdaccess::geo::era_interim::cache::data_fetcher::↔
DataFetcher, [259](#)
 - skdaccess::geo::gldas::data_fetcher::DataFetcher,
[116](#)
 - skdaccess::geo::grace::data_fetcher::DataFetcher,
[281](#)
 - skdaccess::geo::grace::mascon::cache::data_↔
fetcher::DataFetcher, [226](#)
 - skdaccess::geo::groundwater::data_fetcher::Data↔
Fetcher, [157](#)
 - skdaccess::geo::imsdnhs::data_fetcher::Data↔
Fetcher, [251](#)
 - skdaccess::geo::magnetometer::data_fetcher::↔
DataFetcher, [132](#)
 - skdaccess::geo::mahali::rinex::data_fetcher::Data↔
Fetcher, [289](#)
 - skdaccess::geo::mahali::tec::data_fetcher::Data↔
Fetcher, [234](#)
 - skdaccess::geo::mahali::temperature::data_fetcher↔
::DataFetcher, [312](#)
 - skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, [197](#)
 - skdaccess::geo::modis::stream::data_fetcher::↔
DataFetcher, [217](#)
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
[268](#)
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, [245](#)
 - skdaccess::geo::sentinel_1::cache::data_fetcher::↔
DataFetcher, [125](#)
 - skdaccess::geo::srtm::cache::data_fetcher::Data↔
Fetcher, [167](#)
 - skdaccess::geo::uavsar::cache::data_fetcher::Data↔
Fetcher, [175](#)
 - skdaccess::geo::wyoming_sounding::cache::data_↔
fetcher::DataFetcher, [141](#)

- skdaccess::geo::wyoming_sounding::stream::data_↵
_fetcher::DataFetcher, 149
- skdaccess::planetary::ode::cache::data_fetcher::↵
DataFetcher, 210
- skdaccess::solar::sdo::data_fetcher::DataFetcher,
318
- verbose_print
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
101
 - skdaccess::astro::spectra::stream::DataFetcher, 272
 - skdaccess::astro::voyager::data_fetcher::Data_↵
Fetcher, 304
 - skdaccess::engineering::la::generic::stream::Data_↵
Fetcher, 107
 - skdaccess::engineering::webcam::mit_sailing_↵
::stream::DataFetcher, 182
 - skdaccess::finance::timeseries::stream::Data_↵
Fetcher, 294
 - skdaccess::framework::data_class::DataFetcher_↵
Base, 322
 - skdaccess::framework::data_class::DataFetcher_↵
Cache, 330
 - skdaccess::framework::data_class::DataFetcher_↵
Local, 335
 - skdaccess::framework::data_class::DataFetcher_↵
Storage, 342
 - skdaccess::framework::data_class::DataFetcher_↵
Stream, 347
 - skdaccess::geo::era_interim::cache::data_fetcher::↵
DataFetcher, 258
 - skdaccess::geo::gldas::data_fetcher::DataFetcher,
115
 - skdaccess::geo::grace::data_fetcher::DataFetcher,
279
 - skdaccess::geo::grace::mascon::cache::data_↵
fetcher::DataFetcher, 224
 - skdaccess::geo::groundwater::data_fetcher::Data_↵
Fetcher, 156
 - skdaccess::geo::imsdnhs::data_fetcher::Data_↵
Fetcher, 250
 - skdaccess::geo::magnetometer::data_fetcher::↵
DataFetcher, 130
 - skdaccess::geo::mahali::rinex::data_fetcher::Data_↵
Fetcher, 288
 - skdaccess::geo::mahali::tec::data_fetcher::Data_↵
Fetcher, 233
 - skdaccess::geo::mahali::temperature::data_fetcher_↵
::DataFetcher, 311
 - skdaccess::geo::modis::cache::data_fetcher::Data_↵
Fetcher, 194
 - skdaccess::geo::modis::stream::data_fetcher::↵
DataFetcher, 215
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
266
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, 242
 - skdaccess::geo::sentinel_1::cache::data_fetcher::↵
DataFetcher, 123
 - skdaccess::geo::srtm::cache::data_fetcher::Data_↵
Fetcher, 164
 - skdaccess::geo::uavsar::cache::data_fetcher::Data_↵
Fetcher, 174
 - skdaccess::geo::wyoming_sounding::cache::data_↵
fetcher::DataFetcher, 139
 - skdaccess::geo::wyoming_sounding::stream::data_↵
_fetcher::DataFetcher, 147
 - skdaccess::planetary::ode::cache::data_fetcher::↵
DataFetcher, 206
 - skdaccess::solar::sdo::data_fetcher::DataFetcher,
317
- western_lon
 - skdaccess::planetary::ode::cache::data_fetcher::↵
DataFetcher, 210
- widget
 - skdaccess::utilities::file_browser::FileBrowser, 359
- writeConfig
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
101
 - skdaccess::astro::spectra::stream::DataFetcher, 273
 - skdaccess::astro::voyager::data_fetcher::Data_↵
Fetcher, 305
 - skdaccess::engineering::la::generic::stream::Data_↵
Fetcher, 107
 - skdaccess::engineering::webcam::mit_sailing_↵
::stream::DataFetcher, 183
 - skdaccess::finance::timeseries::stream::Data_↵
Fetcher, 294
 - skdaccess::framework::data_class::DataFetcher_↵
Base, 322
 - skdaccess::framework::data_class::DataFetcher_↵
Cache, 330
 - skdaccess::framework::data_class::DataFetcher_↵
Local, 336
 - skdaccess::framework::data_class::DataFetcher_↵
Storage, 342
 - skdaccess::framework::data_class::DataFetcher_↵
Stream, 347
 - skdaccess::geo::era_interim::cache::data_fetcher::↵
DataFetcher, 258
 - skdaccess::geo::gldas::data_fetcher::DataFetcher,
115
 - skdaccess::geo::grace::data_fetcher::DataFetcher,
279
 - skdaccess::geo::grace::mascon::cache::data_↵
fetcher::DataFetcher, 225
 - skdaccess::geo::groundwater::data_fetcher::Data_↵
Fetcher, 156

- skdaccess::geo::imsdnh::data_fetcher::Data↔
Fetcher, [250](#)
- skdaccess::geo::magnetometer::data_fetcher::↔
DataFetcher, [131](#)
- skdaccess::geo::mahali::rinex::data_fetcher::Data↔
Fetcher, [288](#)
- skdaccess::geo::mahali::tec::data_fetcher::Data↔
Fetcher, [233](#)
- skdaccess::geo::mahali::temperature::data_fetcher↔
::DataFetcher, [311](#)
- skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, [195](#)
- skdaccess::geo::modis::stream::data_fetcher::↔
DataFetcher, [215](#)
- skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
[266](#)
- skdaccess::geo::pbo::data_fetcher::DataFetcher, [243](#)
- skdaccess::geo::sentinel_1::cache::data_fetcher::↔
DataFetcher, [123](#)
- skdaccess::geo::srtm::cache::data_fetcher::Data↔
Fetcher, [164](#)
- skdaccess::geo::uavsar::cache::data_fetcher::Data↔
Fetcher, [174](#)
- skdaccess::geo::wyoming_sounding::cache::data_↔
fetcher::DataFetcher, [139](#)
- skdaccess::geo::wyoming_sounding::stream::data↔
_fetcher::DataFetcher, [148](#)
- skdaccess::planetary::ode::cache::data_fetcher::↔
DataFetcher, [207](#)
- skdaccess::solar::sdo::data_fetcher::DataFetcher,
[317](#)
- writeConfigItem
 - skdaccess::astro::kepler::data_fetcher::DataFetcher,
[102](#)
 - skdaccess::astro::spectra::stream::DataFetcher, [273](#)
 - skdaccess::astro::voyager::data_fetcher::Data↔
Fetcher, [305](#)
 - skdaccess::engineering::la::generic::stream::Data↔
Fetcher, [108](#)
 - skdaccess::engineering::webcam::mit_sailing↔
::stream::DataFetcher, [183](#)
 - skdaccess::finance::timeseries::stream::Data↔
Fetcher, [294](#)
 - skdaccess::framework::data_class::DataFetcher↔
Base, [323](#)
 - skdaccess::framework::data_class::DataFetcher↔
Cache, [330](#)
 - skdaccess::framework::data_class::DataFetcher↔
Local, [336](#)
 - skdaccess::framework::data_class::DataFetcher↔
Storage, [342](#)
 - skdaccess::framework::data_class::DataFetcher↔
Stream, [347](#)
 - skdaccess::geo::era_interim::cache::data_fetcher::↔
DataFetcher, [258](#)
 - skdaccess::geo::gldas::data_fetcher::DataFetcher,
[115](#)
 - skdaccess::geo::grace::data_fetcher::DataFetcher,
[280](#)
 - skdaccess::geo::grace::mascon::cache::data_↔
fetcher::DataFetcher, [225](#)
 - skdaccess::geo::groundwater::data_fetcher::Data↔
Fetcher, [156](#)
 - skdaccess::geo::imsdnh::data_fetcher::Data↔
Fetcher, [250](#)
 - skdaccess::geo::magnetometer::data_fetcher::↔
DataFetcher, [131](#)
 - skdaccess::geo::mahali::rinex::data_fetcher::Data↔
Fetcher, [288](#)
 - skdaccess::geo::mahali::tec::data_fetcher::Data↔
Fetcher, [233](#)
 - skdaccess::geo::mahali::temperature::data_fetcher↔
::DataFetcher, [311](#)
 - skdaccess::geo::modis::cache::data_fetcher::Data↔
Fetcher, [195](#)
 - skdaccess::geo::modis::stream::data_fetcher::↔
DataFetcher, [215](#)
 - skdaccess::geo::ngl_gps::data_fetcher::DataFetcher,
[266](#)
 - skdaccess::geo::pbo::data_fetcher::DataFetcher, [243](#)
 - skdaccess::geo::sentinel_1::cache::data_fetcher::↔
DataFetcher, [124](#)
 - skdaccess::geo::srtm::cache::data_fetcher::Data↔
Fetcher, [165](#)
 - skdaccess::geo::uavsar::cache::data_fetcher::Data↔
Fetcher, [174](#)
 - skdaccess::geo::wyoming_sounding::cache::data_↔
fetcher::DataFetcher, [140](#)
 - skdaccess::geo::wyoming_sounding::stream::data↔
_fetcher::DataFetcher, [148](#)
 - skdaccess::planetary::ode::cache::data_fetcher::↔
DataFetcher, [207](#)
 - skdaccess::solar::sdo::data_fetcher::DataFetcher,
[317](#)
- x_offset
 - skdaccess::utilities::image_util, [36](#)
 - skdaccess::utilities::image_util::LinearGeolocation,
[372](#)
 - skdaccess::utilities::image_util::SplineLatLon, [391](#)
 - skdaccess::utilities::modis_util::LatLon, [368](#)
- x_spline
 - skdaccess::utilities::image_util, [36](#)
- y_offset
 - skdaccess::utilities::image_util, [36](#)
 - skdaccess::utilities::image_util::LinearGeolocation,
[373](#)
 - skdaccess::utilities::image_util::SplineLatLon, [391](#)

skdaccess::utilities::modis_util::LatLon, [368](#)
y_spline
 skdaccess::utilities::image_util, [36](#)
year_list
 skdaccess::astro::voyager::data_fetcher::Data↔
 Fetcher, [306](#)
 skdaccess::geo::wyoming_sounding::cache::data_↔
 fetcher::DataFetcher, [141](#)
 skdaccess::geo::wyoming_sounding::stream::data_↔
 _fetcher::DataFetcher, [150](#)