DARC

Generated by Doxygen 1.8.15

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

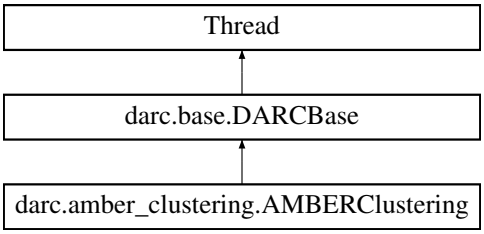
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 darc.amber_clustering.AMBERClustering Class Reference

Inheritance diagram for darc.amber_clustering.AMBERClustering:



Public Member Functions

- def __init__ (self)
- def process_command (self, command)
- def start_observation (self, obs_config)
- def stop_observation (self)
- def process_triggers (self)

Public Attributes

- **needs_source_queue**
- **needs_target_queue**
- **proc_thread**
- **hdr_mapping**
- **obs_config**
- **observation_running**
- **amber_triggers**

3.1.1 Detailed Description

```
Cluster AMBER clusters
```

3.1.2 Constructor & Destructor Documentation

3.1.2.1 __init__()

```
def darc.amber_clustering.AMBERClustering.__init__ (
              self )
```

Initialisation

Reimplemented from darc.base.DARCBase.

3.1.3 Member Function Documentation

3.1.3.1 process_command()

```
def darc.amber_clustering.AMBERClustering.process_command (
              self,
              command )
```

Process command received from queue
:return:

3.1.3.2 process_triggers()

```
def darc.amber_clustering.AMBERClustering.process_triggers (
              self )
```

Applies thresholding to clusters
Puts approved clusters on queue

3.1.3.3 start_observation()

```
def darc.amber_clustering.AMBERClustering.start_observation (
              self,
              obs_config )
```

Parse obs config and start listening for amber triggers on queue

3.1.3.4 stop_observation()

```
def darc.amber_clustering.AMBERClustering.stop_observation (
              self )
```

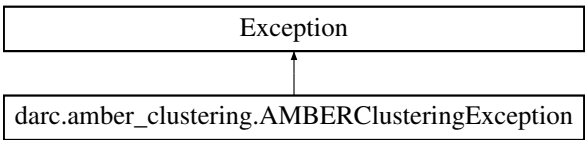
Stop observation

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

- darc/amber_clustering.py

3.2 darc.amber_clustering.AMBERClusteringException Class Reference

Inheritance diagram for darc.amber_clustering.AMBERClusteringException:

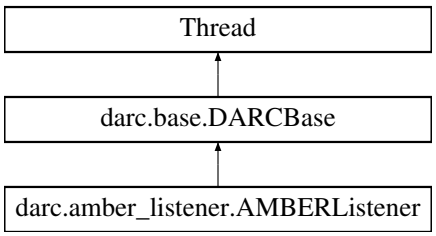


The documentation for this class was generated from the following file:

- darc/amber_clustering.py

3.3 darc.amber_listener.AMBERListener Class Reference

Inheritance diagram for darc.amber_listener.AMBERListener:



Public Member Functions

- def __init__ (self)
- def start_observation (self, obs_config)
- def stop_observation (self)
- def **process_command** (self)

Public Attributes

- **needs_target_queue**
- **observation_threads**
- **observation_events**

Private Member Functions

- def _follow_file (self, fname, event)

3.3.1 Detailed Description

```
Listens to AMBER triggers and puts them on a queue.
```

3.3.2 Constructor & Destructor Documentation

3.3.2.1 __init__()

```
def darc.amber_listener.AMBERListener.__init__ (
              self )
```

```
Initialisation
```

Reimplemented from darc.base.DARCBase.

3.3.3 Member Function Documentation

3.3.3.1 _follow_file()

```
def darc.amber_listener.AMBERListener._follow_file (
              self,
              fname,
              event )  [private]
```

```
Tail a file an put lines on queue
:param fname: file to follow
:param event: stop event
```

3.3.3.2 start_observation()

```
def darc.amber_listener.AMBERListener.start_observation (
              self,
              obs_config )
```

```
Start an observation
:param obs_config: observation config dict
:return:
```

3.3.3.3 stop_observation()

```
def darc.amber_listener.AMBERListener.stop_observation (
              self )
```

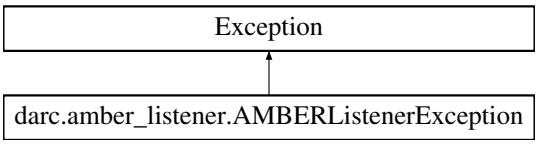
```
Stop any running observation
```

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

- darc/amber_listener.py

3.4 darc.amber_listener.AMBERListenerException Class Reference

Inheritance diagram for darc.amber_listener.AMBERListenerException:

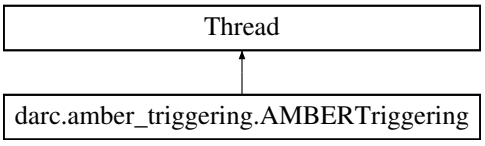


The documentation for this class was generated from the following file:

- darc/amber_listener.py

3.5 darc.amber_triggering.AMBERTriggering Class Reference

Inheritance diagram for darc.amber_triggering.AMBERTriggering:



Public Member Functions

- def **__init__** (self)
- def stop (self)
- def set_source_queue (self, queue)
- def **run** (self)
- def process_triggers (self, triggers)

Public Attributes

- **daemon**
- **stop_event**
- **amber_queue**
- **cluster_queue**
- **hdr_mapping**
- **start_time**
- **vo_config**
- **logger**
- **voevent_queue_server**

3.5.1 Detailed Description

```
Process AMBER triggers and turn into trigger message
```

3.5.2 Member Function Documentation

3.5.2.1 process_triggers()

```
def darc.amber_triggering.AMBERTriggering.process_triggers (
              self,
              triggers )
```

```
Applies thresholding to triggers
Put approved triggers on queue
:param triggers: list of triggers to process
```

3.5.2.2 set_source_queue()

```
def darc.amber_triggering.AMBERTriggering.set_source_queue (
              self,
              queue )
```

```
:param queue: Source of amber triggers
```

3.5.2.3 stop()

```
def darc.amber_triggering.AMBERTriggering.stop (
              self )
```

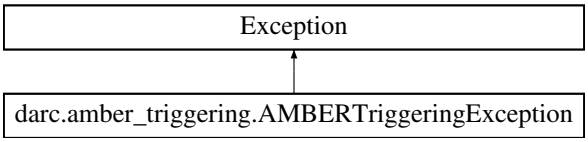
Stop the service

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

- darc/amber_triggering.py

3.6 darc.amber_triggering.AMBERTriggeringException Class Reference

Inheritance diagram for darc.amber_triggering.AMBERTriggeringException:

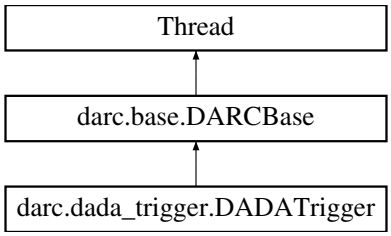


The documentation for this class was generated from the following file:

- darc/amber_triggering.py

3.7 darc.dada_trigger.DADATrigger Class Reference

Inheritance diagram for darc.dada_trigger.DADATrigger:



Public Member Functions

- def __init__ (self)
- def process_command (self, command)
- def cleanup (self)
- def send_event (self, triggers)

Public Attributes

- **thread**

3.7.1 Detailed Description

```
Generate and send dada_dbevent triggers
```

3.7.2 Constructor & Destructor Documentation

3.7.2.1 __init__()

```
def darc.dada_trigger.DADATrigger.__init__ (
              self )
```

```
Initialisation
```

Reimplemented from darc.base.DARCBase.

3.7.3 Member Function Documentation

3.7.3.1 cleanup()

```
def darc.dada_trigger.DADATrigger.cleanup (
              self )
```

```
Remove any remaining threads
```

Reimplemented from darc.base.DARCBase.

3.7.3.2 process_command()

```
def darc.dada_trigger.DADATrigger.process_command (
              self,
              command )
```

```
Process command received from queue
:param command: command dict
```

3.7.3.3 send_event()

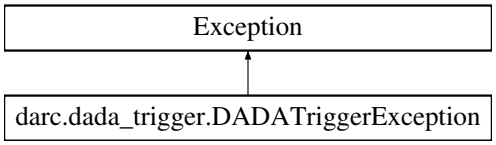
```
def darc.dada_trigger.DADATrigger.send_event (
              self,
              triggers )

Send trigger to dada_dbevent
:param triggers: list of trigger dictionaries
```

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

- darc/dada_trigger.py

3.8 darc.dada_trigger.DADATriggerException Class Reference

Inheritance diagram for darc.dada_trigger.DADATriggerException:

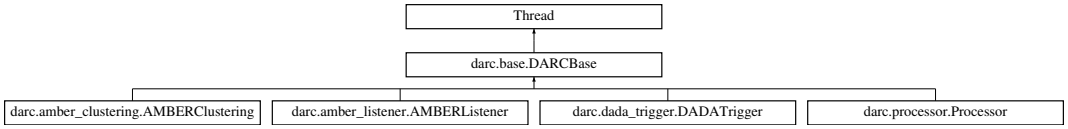


The documentation for this class was generated from the following file:

- darc/dada_trigger.py

3.9 darc.base.DARCBase Class Reference

Inheritance diagram for darc.base.DARCBase:



Public Member Functions

- def __init__ (self)
- def stop (self)
- def set_source_queue (self, queue)
- def set_target_queue (self, queue)
- def run (self)
- def **start_observation** (self, ∗args, ∗∗kwargs)
- def **stop_observation** (self, ∗args, ∗∗kwargs)
- def **cleanup** (self)
- def **process_command** (self, ∗args, ∗∗kwargs)

Public Attributes

- **daemon**
- **stop_event**
- **needs_source_queue**
- **needs_target_queue**
- **source_queue**
- **target_queue**
- **log_name**
- **logger**

3.9.1 Detailed Description

```
DARC Base class
Provides common methods
```

3.9.2 Constructor & Destructor Documentation

3.9.2.1 __init__()

```
def darc.base.DARCBase.__init__ (
              self )
```

```
Initialisation
```

Reimplemented in darc.amber_clustering.AMBERClustering, darc.processor.Processor, darc.amber_listener.AMBERListener, and darc.dada_trigger.DADATrigger.

3.9.3 Member Function Documentation

3.9.3.1 run()

```
def darc.base.DARCBase.run (
              self )
```

```
Main loop
```

3.9.3.2 set_source_queue()

```
def darc.base.DARCBase.set_source_queue (
              self,
              queue )
```

```
:param queue: Input queue
```

3.9.3.3 set_target_queue()

```
def darc.base.DARCBase.set_target_queue (
              self,
              queue )
```

```
:param queue: Output queue
```

3.9.3.4 stop()

```
def darc.base.DARCBase.stop (
              self )
```

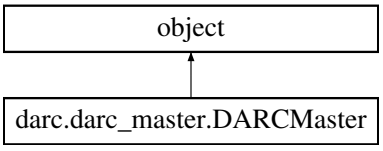
```
Set the stop event
```

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

- darc/base.py

3.10 darc.darc_master.DARCMaster Class Reference

Inheritance diagram for darc.darc_master.DARCMaster:



Public Member Functions

- def __init__ (self)
- def run (self)
- def parse_message (self, raw_message)
- def process_message (self, service, command, payload)
- def check_status (self, service)
- def start_service (self, service)
- def stop_service (self, service)
- def restart_service (self, service)
- def create_thread (self, service)
- def stop (self)
- def start_observation (self, config_file)
- def stop_observation (self)

Public Attributes

- **stop_event**
- **hostname**
- **amber_listener_queue**
- **amber_trigger_queue**
- **dadatrigger_queue**
- **processor_queue**
- **all_queues**
- **services**
- **service_mapping**
- **logger**
- **threads**
- **command_socket**

Private Member Functions

- def _load_config (self)
- def _load_yaml (self, config_file)
- def _load_parset (self, config_file)
- def _reload (self, service)

3.10.1 Constructor & Destructor Documentation

3.10.1.1 __init__()

```
def darc.darc_master.DARCMaster.__init__ (
            self )
```

Setup queues, config, logging

3.10.2 Member Function Documentation

3.10.2.1 _load_config()

```
def darc.darc_master.DARCMaster._load_config (
              self )  [private]
```

Load configuration file

3.10.2.2 _load_parset()

```
def darc.darc_master.DARCMaster._load_parset (
              self,
              config_file )  [private]
```

Load parset file and convert to observation config
:param config_file: Path to parset file
:return: observation config dict

3.10.2.3 _load_yaml()

```
def darc.darc_master.DARCMaster._load_yaml (
              self,
              config_file )  [private]
```

Load yaml file and convert to observation config
:param config_file: Path to yaml file
:return: observation config dict

3.10.2.4 _reload()

```
def darc.darc_master.DARCMaster._reload (
              self,
              service )  [private]
```

Reimport service from .py file
:param service: which service to reload
:return:

3.10.2.5 check_status()

```
def darc.darc_master.DARCMaster.check_status (
              self,
              service )
```

```
:param service: Service to check status of
:return: status, reply
```

3.10.2.6 create_thread()

```
def darc.darc_master.DARCMaster.create_thread (
              self,
              service )
```

```
:param service: service to create a new thread for
```

3.10.2.7 parse_message()

```
def darc.darc_master.DARCMaster.parse_message (
              self,
              raw_message )
```

```
Parse raw received message
:param raw_message: message as single string
:return: status (str), reply (dict)
```

3.10.2.8 process_message()

```
def darc.darc_master.DARCMaster.process_message (
              self,
              service,
              command,
              payload )
```

```
:param service: service to interact with
:param command: command to run
:param payload: payload for command
:return: status (str), reply (dict)
```

3.10.2.9 restart_service()

```
def darc.darc_master.DARCMaster.restart_service (
              self,
              service )
```

```
:param service: service to restart
```

3.10.2.10 run()

```
def darc.darc_master.DARCMaster.run (
              self )
```

```
Listen for message on the command socket
```

3.10.2.11 start_observation()

```
def darc.darc_master.DARCMaster.start_observation (
              self,
              config_file )
```

```
Start an observation
:param config_file: Path to observation config file
```

3.10.2.12 start_service()

```
def darc.darc_master.DARCMaster.start_service (
              self,
              service )
```

```
:param service: service to start
:return: status
```

3.10.2.13 stop()

```
def darc.darc_master.DARCMaster.stop (
              self )
```

```
Stop all services and exit
```

3.10.2.14 stop_observation()

```
def darc.darc_master.DARCMaster.stop_observation (
              self )
```

```
Stop an observation
:return: status, reply message
```

3.10.2.15 stop_service()

```
def darc.darc_master.DARCMaster.stop_service (
              self,
              service )
```

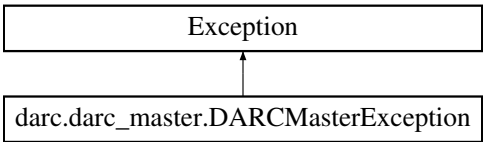
```
:param service: service to stop
:return: status, reply
```

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

- darc/darc_master.py

3.11 darc.darc_master.DARCMasterException Class Reference

Inheritance diagram for darc.darc_master.DARCMasterException:

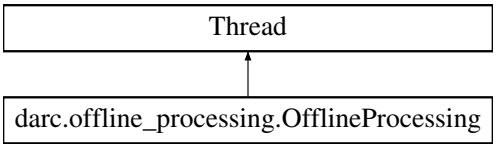


The documentation for this class was generated from the following file:

- darc/darc_master.py

3.12 darc.offline_processing.OfflineProcessing Class Reference

Inheritance diagram for darc.offline_processing.OfflineProcessing:



Public Member Functions

- def **__init__** (self)
- def set_source_queue (self, queue)
- def stop (self)
- def run (self)

Public Attributes

- **daemon**
- **stop_event**
- **observation_queue**
- **threads**
- **config**
- **logger**
- **host_type**

Private Member Functions

- def _start_observation_master (self, obs_config)
- def _start_observation_worker (self, obs_config)
- def _merge_triggers (self, obs_config)
- def _cluster (self, obs_config, filterbank_name, tab=None, ind=None, sbmin=None, sbmax=None, out=None)
- def _merge_hdf5 (self, obs_config, output_file)
- def _classify (self, obs_config, input_file)
- def _merge_plots (self, obs_config)
- def _gather_results (self, obs_config, ∗∗kwargs)
- def _get_coordinates (self, obs_config)
- def _get_overview (self, obs_config)
- def _get_ymw16 (self, obs_config)
- def _fold_pulsar (self, source, obs_config)
- def _load_parset (self, obs_config)

3.12.1 Member Function Documentation

3.12.1.1 _classify()

```
def darc.offline_processing.OfflineProcessing._classify (
            self,
            obs_config,
            input_file )  [private]
```

```
Run the ML classifier in python 3 virtualenv
:param obs_config: Observation config
:param input_file: HDF5 file to process
```

3.12.1.2 _cluster()

```
def darc.offline_processing.OfflineProcessing._cluster (
              self,
              obs_config,
              filterbank_name,
              tab = None,
              ind = None,
              sbmin = None,
              sbmax = None,
              out = None )  [private]
```

```
Run triggers.py
:param obs_config: Observation config
:param filterbank_name: Full path to filterbank file (TAB/IAB) or prefix (SB) to use
:param tab: TAB number to process (0 for IAB, absent/None for SB)
:param ind: Index of out array where
:param sbmin: First SB to process (SB mode only)
:param sbmax: Last SB to process (SB mode only)
:param out: array where return value is put at index <ind> (optional)
```

3.12.1.3 _fold_pulsar()

```
def darc.offline_processing.OfflineProcessing._fold_pulsar (
              self,
              source,
              obs_config )  [private]
```

```
Fold pulsar with PRESTO
:param source: pulsar name including B or J
:param obs_config: Observation config
```

3.12.1.4 _gather_results()

```
def darc.offline_processing.OfflineProcessing._gather_results (
              self,
              obs_config,
             ** kwargs )  [private]
```

```
Gather output results and put in central location
:param obs_config: Observation config
:param kwargs: number of candidates, output prefixes
```

3.12.1.5 _get_coordinates()

```
def darc.offline_processing.OfflineProcessing._get_coordinates (
              self,
              obs_config )  [private]
```

Generate the coordinates file from the pointing directions
:param obs_config: Observation config
#:return: dict with RA, Dec, gl, gb for each CB

3.12.1.6 _get_overview()

```
def darc.offline_processing.OfflineProcessing._get_overview (
              self,
              obs_config )  [private]
```

Generate observation overview file
:param obs_config: Observation config

3.12.1.7 _get_ymw16()

```
def darc.offline_processing.OfflineProcessing._get_ymw16 (
              self,
              obs_config )  [private]
```

Get YMW16 DM
:param obs_config: Observation config
:return: YMW16 DM

3.12.1.8 _load_parset()

```
def darc.offline_processing.OfflineProcessing._load_parset (
              self,
              obs_config )  [private]
```

Load the observation parset
:param obs_config: Observation config
:return: parset as dict

3.12.1.9 _merge_hdf5()

```
def darc.offline_processing.OfflineProcessing._merge_hdf5 (
              self,
              obs_config,
              output_file )  [private]
```

Merge HDF5 files generated by clustering
:param obs_config: Observation config
:param output_file: Filename of merged HDF5 data

3.12.1.10 _merge_plots()

```
def darc.offline_processing.OfflineProcessing._merge_plots (
              self,
              obs_config )  [private]
```

Merge classifier output plots
:param obs_config: Observation config

3.12.1.11 _merge_triggers()

```
def darc.offline_processing.OfflineProcessing._merge_triggers (
              self,
              obs_config )  [private]
```

Merge AMBER triggers into one file
:param obs_config: Observation config

3.12.1.12 _start_observation_master()

```
def darc.offline_processing.OfflineProcessing._start_observation_master (
              self,
              obs_config )  [private]
```

Start observation on master node
:param obs_config: Observation config

3.12.1.13 _start_observation_worker()

```
def darc.offline_processing.OfflineProcessing._start_observation_worker (
              self,
              obs_config )  [private]
```

Start observation on worker node
:param obs_config: Observation config

3.12.1.14 run()

```
def darc.offline_processing.OfflineProcessing.run (
              self )
```

Wait for start_observation command

3.12.1.15 set_source_queue()

```
def darc.offline_processing.OfflineProcessing.set_source_queue (
              self,
              queue )
```

:param queue: Source of start_observation commands

3.12.1.16 stop()

```
def darc.offline_processing.OfflineProcessing.stop (
              self )
```

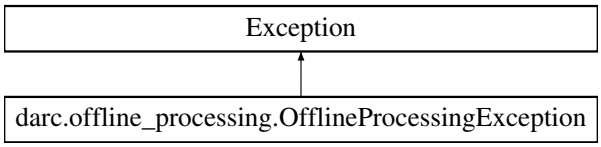
Stop the service

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

- darc/offline_processing.py

3.13 darc.offline_processing.OfflineProcessingException Class Reference

Inheritance diagram for darc.offline_processing.OfflineProcessingException:

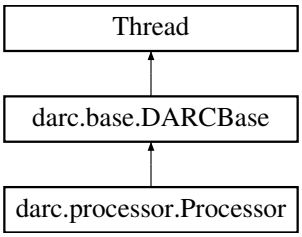


The documentation for this class was generated from the following file:

- darc/offline_processing.py

3.14 darc.processor.Processor Class Reference

Inheritance diagram for darc.processor.Processor:



Public Member Functions

- def __init__ (self)
- def **process_command** (self, command)
- def start_observation (self, obs_config)
- def stop_observation (self)
- def read_and_process_data (self, obs_config)
- def lofar_trigger (self, trigger)
- def iquv_trigger (self, trigger)

Public Attributes

- **needs_target_queue**
- **thread**
- **observation_running**
- **rtproc**
- **model_freqtime**
- **model_dmtime**

3.14.1 Detailed Description

```
Process dada events
```

3.14.2 Constructor & Destructor Documentation

3.14.2.1 __init__()

```
def darc.processor.Processor.__init__ (
              self )
```

```
Initialisation
```

Reimplemented from darc.base.DARCBase.

3.14.3 Member Function Documentation

3.14.3.1 iquv_trigger()

```
def darc.processor.Processor.iquv_trigger (
              self,
              trigger )
```

```
Trigger IQUV data dump
```

3.14.3.2 lofar_trigger()

```
def darc.processor.Processor.lofar_trigger (
              self,
              trigger )
```

```
Contact VOEvent server to send LOFAR trigger
```

3.14.3.3 read_and_process_data()

```
def darc.processor.Processor.read_and_process_data (
              self,
              obs_config )
```

```
Read ringbuffer pages and process them
:param obs_config: observation config dict
```

3.14.3.4 start_observation()

```
def darc.processor.Processor.start_observation (
              self,
              obs_config )
```

```
Start observation
```

3.14.3.5 stop_observation()

```
def darc.processor.Processor.stop_observation (
              self )
```

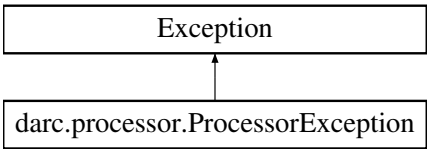
```
Stop observation
```

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

- darc/processor.py

3.15 darc.processor.ProcessorException Class Reference

Inheritance diagram for darc.processor.ProcessorException:

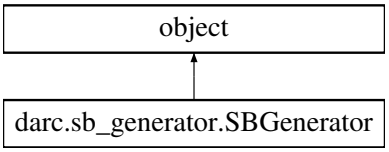


The documentation for this class was generated from the following file:

- darc/processor.py

3.16 darc.sb_generator.SBGenerator Class Reference

Inheritance diagram for darc.sb_generator.SBGenerator:



Public Member Functions

- def **__init__** (self, fname=None, science_case=None)
- def reversed (self)
- def reversed (self, state)
- def from_table (cls, fname)
- def from_science_case (cls, science_case)
- def get_map (self, sb)
- def synthesize_beam (self, data, sb)

Public Attributes

- **sb_table**
- **nsub**
- **numtab**
- **numsb**
- **science_case**
- **fname**
- **sb_mapping**

Private Member Functions

- def _load_table (self)

Private Attributes

- **__reversed**

3.16.1 Member Function Documentation

3.16.1.1 _load_table()

```
def darc.sb_generator.SBGenerator._load_table (
              self )  [private]
```

Load the SB table

3.16.1.2 from_science_case()

```
def darc.sb_generator.SBGenerator.from_science_case (
              cls,
              science_case )
```

```
Initalize default table for given science cases
:param science_case: science case (3 or 4)
:return: SBGenerator object
```

3.16.1.3 from_table()

```
def darc.sb_generator.SBGenerator.from_table (
              cls,
              fname )
```

```
Initalize with provided SB table
:param fname: Path to SB table
:return: SBGenerator object
```

3.16.1.4 get_map()

```
def darc.sb_generator.SBGenerator.get_map (
              self,
              sb )
```

```
Return mapping of requested SB
:param sb: beam to return mapping for
:return: SB mapping for requested beam
```

3.16.1.5 reversed() [1/2]

```
def darc.sb_generator.SBGenerator.reversed (
              self )
```

```
Whether or not the SB table is reversed for use on filterbank data
:return: reversed (bool)
```

3.16.1.6 reversed() [2/2]

```
def darc.sb_generator.SBGenerator.reversed (
              self,
              state )

Reverse the SB table for use on filterbank data
:param state: bool, whether or not to reverse the table
```

3.16.1.7 synthesize_beam()

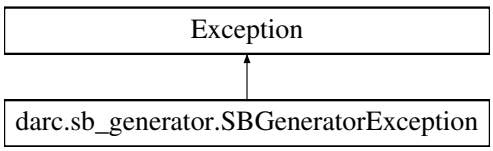
```
def darc.sb_generator.SBGenerator.synthesize_beam (
              self,
              data,
              sb )

Synthesize beam
:param data: TAB data with shape [TAB, freq, time]
:param sb: SB index
:return: SB data with shape [freq, time]
```

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

- darc/sb_generator.py

3.17 darc.sb_generator.SBGeneratorException Class Reference

Inheritance diagram for darc.sb_generator.SBGeneratorException:

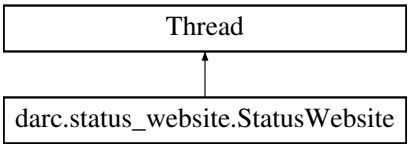


The documentation for this class was generated from the following file:

- darc/sb_generator.py

3.18 darc.status_website.StatusWebsite Class Reference

Inheritance diagram for darc.status_website.StatusWebsite:



Public Member Functions

- def **__init__** (self)
- def run (self)
- def stop (self)
- def publish_status (self, statuses)
- def get_template (self)
- def make_offline_page (self)

Public Attributes

- **stop_event**
- **daemon**
- **services_master**
- **services_worker**
- **all_nodes**
- **logger**

3.18.1 Member Function Documentation

3.18.1.1 get_template()

```
def darc.status_website.StatusWebsite.get_template (
              self )
```

Return the HTML template

3.18.1.2 make_offline_page()

```
def darc.status_website.StatusWebsite.make_offline_page (
              self )
```

Create page for when status website is offline

3.18.1.3 publish_status()

```
def darc.status_website.StatusWebsite.publish_status (
              self,
              statuses )
```

Publish status as simple html webpage

3.18.1.4 run()

```
def darc.status_website.StatusWebsite.run (
              self )
```

3.18.1.5 stop()

```
def darc.status_website.StatusWebsite.stop (
              self )
```

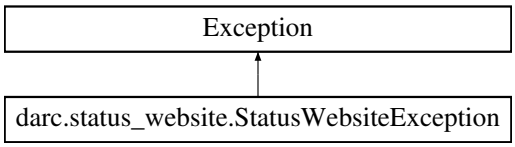
```
Stop the service
```

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

- darc/status_website.py

3.19 darc.status_website.StatusWebsiteException Class Reference

Inheritance diagram for darc.status_website.StatusWebsiteException:

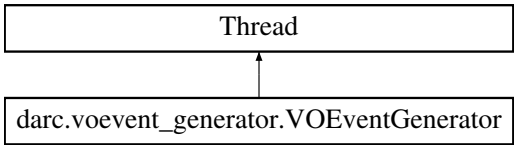


The documentation for this class was generated from the following file:

- darc/status_website.py

3.20 darc.voevent_generator.VOEventGenerator Class Reference

Inheritance diagram for darc.voevent_generator.VOEventGenerator:



Public Member Functions

- def **__init__** (self)
- def stop (self)
- def run (self)
- def create_and_send (self, trigger)
- def **NewVOEvent** (self, dm, dm_err, width, snr, flux, ra, dec, semiMaj, semiMin, ymw16, name, importance, utc, gl, gb)

Public Attributes

- **stop_event**
- **daemon**
- **voevent_server**
- **logger**
- **voevent_queue**

3.20.1 Detailed Description

```
Generate VOEvent from incoming trigger
```

3.20.2 Member Function Documentation

3.20.2.1 create_and_send()

```
def darc.voevent_generator.VOEventGenerator.create_and_send (
              self,
              trigger )
```

```
Creates VOEvent
Sends if enabled in config
:param trigger: Trigger event
```

3.20.2.2 run()

```
def darc.voevent_generator.VOEventGenerator.run (
              self )
```

```
Read triggers from queue and call processing for each trigger
```

3.20.2.3 stop()

```
def darc.voevent_generator.VOEventGenerator.stop (
              self )
```

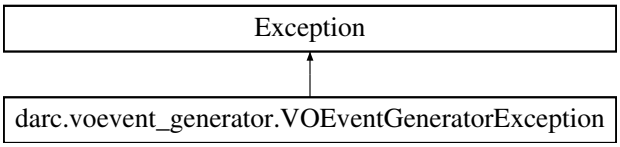
Stop the service

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

- darc/voevent_generator.py

3.21 darc.voevent_generator.VOEventGeneratorException Class Reference

Inheritance diagram for darc.voevent_generator.VOEventGeneratorException:

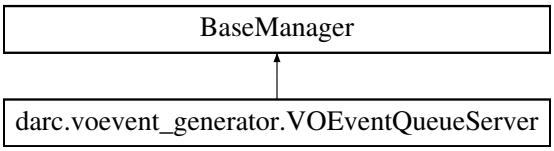


The documentation for this class was generated from the following file:

- darc/voevent_generator.py

3.22 darc.voevent_generator.VOEventQueueServer Class Reference

Inheritance diagram for darc.voevent_generator.VOEventQueueServer:



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

- darc/voevent_generator.py

Index