ska-sdp-mock-dish-devices Documentation

Release 0.5.1

Julian Carrivick, Callan Gray and Matteo Di Carlo

Apr 12, 2024

TANGO DEVICES

This project defines a set of Docker images and Docker compose files that are useful for TANGO control system development.

MOCK DISH MASTER DEVICE

1.1 Introduction

Mock implementation of the SKA Dish Master Tango device based on the interface provided by https://gitlab.com/ska-telescope/ska-sim-dishmaster.

1.2 Usage

This mock implementation when in the *ON* state will simulate mocked attributes listed below of the real device using a mock dataset. Mocking must be started using the *Start* command and may be stopped (for restart) using the *Stop* command.

Additionally, the device will automatically transition to the *Off* state when at the end of the mock data stream.

For API usage see *ska_sdp_mock_dish_devices.MockDishMaster*.

1.3 States

State	Description
OFF	Device is initialized and is not simulating
ON	Attributes are being continuously simulated from mock data by the device

1.4 Properties

Property	Type	Values	Description
mock_achieved	Strin	‘’{“1”: “mnt/data/pointings.hdf”	Json mapping of scan id to a path of PointingTable collections in HDF5 format for *achievedPointing*
antenna_id	Int32	0	Antenna ID within the pointing data
time_scale	Float	1.0	Scale factor for the timing interval pointings are read

1.5 Commands

Command	Argument type	Return type	Action
Scan	int	None	Set device state to ON
EndScan	None	None	Set device state to OFF

1.6 Attributes

Attribute	Type	Format	Values	Description
achieved-Pointing	Float6	SPEC-TRUM[3]	[*Offset since Epoch*, az, el]	Marks the dish actual pointing direction in radians for a corresponding timestamp.

Note: For a full list of planned attributes see https://gitlab.com/ska-telescope/ska-sim-dishmaster/-/blob/master/src/ska_sim_dishmaster/dish_master.fgo

1.6.1 Offset since Epoch

These are specified as an offset in SI seconds (or multiples/sub-multiples of SI seconds) from the TAI epoch of midnight, 1 January 2000, which equates to 1999-12-31T23:59:28Z UTC. See https://confluence.skatelescope.org/display/SWSI/ADR-78+SKA+approach+to+timestamps+in+time+sensitive+data for more information.

Note: The UTC equivalent is different because at the start of Y2000 UTC was 32 seconds behind TAI (it is now further behind because of leap second adjustments).

CHAPTER

TWO

MOCK DISH LEAFNODE DEVICE

2.1 Introduction

Mock implementation of the SKA TMC Dish Leafnode Tango Device provided by https://gitlab.com/ska-telescope/ska-tmc/ska-tmc-dishleafnode.

2.2 Usage

This mock implementation when in the *ON* state will simulate mocked attributes listed below of the real device using a mock dataset. Mocking must be started using the *Start* command and may be stopped (for restart) using the *Stop* command.

Additionally, the device will automatically transition to the *Off* state when at the end of the mock data stream.

For API usage see `ska_sdp_mock_dish_devices.MockDishLeafnode`.

2.3 States

State	Description
OFF	Device is initialized and is not simulating
ON	Attributes are being continuously simulated from mock data by the device

2.4 Properties

Property	Type	Values	Description
mock_desired_	Strin	‘’{“1”: “mnt/data/pointings.hdf”	Json mapping of scan id to a path PointingTable collections in HDF5 format for *desiredPointing*
mock_offset_p	Strin	‘’{“1”: “mnt/data/pointings.hdf”	Json mapping of scan id to a path of PointingTable collections in HDF5 format for *sourceOffset*
antenna_id	Int32	0	Antenna ID within the pointing data
time_scale	Float	1.0	Scale factor for the timing interval pointings are read

2.5 Commands

Command	Argument type	Return type	Action
Scan	int	None	Set device state to ON
EndScan	None	None	Set device state to OFF

2.6 Attributes

Attribute	Type	Format	Values	Description
desired-Pointing	Float6	SPEC-TRUM[3]	[*Offset since Epoch*, az, el]	Marks the dish commanded pointing direction in radians for a corresponding timestamp.

Note: For a full list of planned attributes see https://gitlab.com/ska-telescope/ska-sim-dishmaster/-/blob/master/src/ska_sim_dishmaster/dish_master.fgo

CHAPTER THREE

API REFERENCE

3.1 ska_sdp_mock_dish_devices Package

3.1.1 Classes

MockDishLeafnode(cl, name)	A Mock implementation of ska-tmc-dishleafnode for interfaces used by SDP.
MockDishMaster(cl, name)	A Mock implementation of ska-sim-dishmaster for interfaces used by SDP.

MockDishLeafnode

class ska_sdp_mock_dish_devices.**MockDishLeafnode**(*cl*, *name*)

Bases: `Device`

A Mock implementation of ska-tmc-dishleafnode for interfaces used by SDP.

Attributes Summary

TangoClassName	
antenna_id	Index of the antenna to mock in the PointingTable data.
desired_pointing	TANGO attribute
mock_desired_paths	A json mapping of scan id to HDF5 filepaths adhering to the *ska-sdp-datamodels* PointingTable schema for the desiredPointing attribute.
mock_offset_paths	A json mapping of scan id to HDF5 filepaths adhering to the *ska-sdp-datamodels* PointingTable schema for the sourceOffset attribute.
source_offset	TANGO attribute
time_scale	Factor to adjust the cadence of pointing events by.

Methods Summary

EndScan()	Stops and resets the emulation of desired pointings.
Scan(scan_id)	Starts the emulation of desired pointings.
init_device()	Tango init_device method.
is_EndScan_allowed()	
is_Scan_allowed()	

Attributes Documentation

TangoClassName = 'MockDishLeafnode'

antenna_id: int

Index of the antenna to mock in the PointingTable data.

desired_pointing

TANGO attribute

mock_desired_paths: str

A json mapping of scan id to HDF5 filepaths adhering to the *ska-sdp-datamodels* PointingTable schema for the desiredPointing attribute.

mock_offset_paths: str

A json mapping of scan id to HDF5 filepaths adhering to the *ska-sdp-datamodels* PointingTable schema for the sourceOffset attribute.

source_offset

TANGO attribute

time_scale: float

Factor to adjust the cadence of pointing events by. By default, events will emit at the same cadence as the data, but this will allow events to be emitted at a faster or slower rate.

Methods Documentation

EndScan()

Stops and resets the emulation of desired pointings.

Scan(*scan_id: int*)

Starts the emulation of desired pointings. :param scan_id: scan id corresponding to :type scan_id: int :param a configured mock data path.:

Raises

`ValueError` – scan id out of range.

init_device()

Tango init_device method. Default implementation calls `get_device_properties()`

is_EndScan_allowed()

is_Scan_allowed()

MockDishMaster

class ska_sdp_mock_dish_devices.**MockDishMaster**(*cl*, *name*)

Bases: `Device`

A Mock implementation of ska-sim-dishmaster for interfaces used by SDP.

Attributes Summary

TangoClassName	
achieved_pointing	TANGO attribute
antenna_id	Index of the antenna to mock in the PointingTable data.
mock_achieved_paths	A json mapping of scan id to HDF5 filepaths adhering to the *ska-sdp-datamodels* PointingTable schema for the achievedPointing attribute.
time_scale	Factor to adjust the cadence of pointing events by.

Methods Summary

EndScan()	Ends the emulated scanning of achieved pointings.
Scan(scan_id)	Starts the emulated scanning of achieved pointings.
init_device()	Tango init_device method.
is_EndScan_allowed()	
is_Scan_allowed()	

Attributes Documentation

TangoClassName = 'MockDishMaster'

achieved_pointing

TANGO attribute

antenna_id: int

Index of the antenna to mock in the PointingTable data.

mock_achieved_paths: str

A json mapping of scan id to HDF5 filepaths adhering to the *ska-sdp-datamodels* PointingTable schema for the achievedPointing attribute.

time_scale: float

Factor to adjust the cadence of pointing events by. By default, events will emit at the same cadence as the data, but this will allow events to be emitted at a faster or slower rate.

Methods Documentation

`EndScan()`

Ends the emulated scanning of achieved pointings.

`Scan(`*scan_id: int*`)`

Starts the emulated scanning of achieved pointings.

Parameters

- **`scan_id`** (*`int`*) – scan id corresponding to
- **`path.`** (*`a configured mock data`*) –

Raises

`ValueError` – scan id out of range.

`init_device()`

Tango init_device method. Default implementation calls `get_device_properties()`

`is_EndScan_allowed()`

`is_Scan_allowed()`

CHAPTER

FOUR

DEVELOPER GUIDE

4.1 Get Started

4.1.1 Install dependencies

You will need:

- Python >=3.10
- Poetry >=1.8.2
- Docker

Before running `poetry install` to install the Python dependencies you will need a system tango library installed on your system (which is required by `pytango`).

For Debian/Ubuntu:

```
$ sudo apt update
$ sudo apt install -y curl git build-essential libboost-python-dev libtango-dev
```

Please note that:

- The `libtango-dev` will install an old version of the TANGO-controls framework (9.2.5);
- The best way to get the latest version of the framework is compiling it (instructions can be found here)
- MacOS is not supported
- Windows users will need to use WSL
- The above script has been tested with Ubuntu 22.04.

During this step, `libtango-dev` *installation might ask for the Tango Server IP:PORT. Just accept the default proposed value.*

Once you have that available you can install the python dependencies. Note that on some systems, you may need to explicitly provide the path to the tango C++ headers:

```
CPPFLAGS=-I/usr/include/tango poetry install
```

4.1.2 Run linting and testing

Since this project supports interfacing with Kafka, we need to spin up a instance for testing. For this we use Docker Compose so you will need to install docker engine, and docker compose.

When these are available you can run the tests using

```
$ poetry run make python-tests
```

Linting can be run in a similar way:

```
$ poetry run make python-lint
```

4.2 Other

4.2.1 Makefile targets

This project contains a Makefile which acts as a UI for building Docker images, testing images, and for launching interactive developer environments. For the documentation of the Makefile run `make help`.

4.2.2 TANGO References

- https://pytango.readthedocs.io/en/stable/contents.html
- https://pytango.readthedocs.io/en/stable/green_modes/green_modes_server.html
- https://pytango.readthedocs.io/en/stable/testing.html
- https://pytango.readthedocs.io/en/stable/client_api/index.html
- https://pytango.readthedocs.io/en/stable/server_api/server.html

4.3 ska-tango-images

Please note that this project make use of the charts and docker images for the TANGO-controls framework available at here.

4.4 Test execution

All tests created for the present project can run in simulated mode or in a real environment except for the ones marked as `post_deployment`.

`make test-deployment` runs all the application test procedures defined in the folder `tests` in a new pod in the k8s deployment. This target copies the tests folder into a new pod and execute the test with the option `--true-context` allowing the execution to happen against the real application. On success it copies the resulting output and test artefacts out of the container and into the folder `charts/build` directory, ready for inclusion in the CI server's downloadable artefacts.

`make python-test` runs the application test procedures (except the ones marked as `post_deployment`) defined in the folder `tests` without starting a new pod. The result will be found in the `build`.

CHAPTER

FIVE

CHANGE LOG

All notable changes to this project will be documented in this file. This project adheres to Semantic Versioning.

5.1 [Development]

5.2 [0.5.1]

5.2.1 Fixed

- Python logging setup now correctly follows the `-v` command line option.

5.3 [0.5.0]

5.3.1 Changed

- **BREAKING** Moved *sourceOffset* and *mock_offset_paths* to Mock Dish Leafnode

5.4 [0.4.0]

5.4.1 Added

- Added support for time axis in pointing table
- Added *sourceOffset* attribute to Mock Dish Master
- Added *mock_offset_paths* property for source offset data to Mock Dish Master

5.4.2 Changed

- **BREAKING** Renamed *mock_data_paths* attribute to *mock_achieved_paths* on Mock Dish Master
- **BREAKING** Renamed *mock_data_paths* attribute to *mock_desired_paths* on Mock Dish Leafnode

5.5 [0.3.0]

5.5.1 Added

- Added support for multiple scans of datasets.

5.5.2 Changed

- Changed *Start()* command to *Scan(scan_id)*
- Changed *Stop()* command to *EndScan()*
- Changed *mock_data_path* to *mock_data_paths*

5.6 [0.2.0]

5.6.1 Added

- Mock Dish Master and Mock Dish Leafnode doc pages

5.6.2 Changed

- Changed *desired_pointing* attribute name to *desiredPointing*
- Changed *achieved_pointing* attribute name to *achievedPointing*

5.7 [0.1.0]

5.7.1 Added

- Added Mock Dish Leafnode
- Added Mock Dish Master

PYTHON MODULE INDEX

INDEX