# developer.skatelescope.org Documentation

***Release 1.3.1***

**Marco Bartolini**

**Nov 23, 2021**

# Home

Documentation Status pipeline status coverage report

Repository for handling SKA Telescope Model information. This is a key component of the SKA architecture, centralising knowledge required to reason about the telescope as a whole.

# CHAPTER 1

# Scope

Requirement SKA1-SYS_REQ-2645 defines the telescope model as follows: “A dynamic computational model of the Telescope shall be used to answer all queries about the state of the Telescope. The telescope model shall consist of configuration information, numerical models, empirical parameters, and conventions”.

The particular “queries” we are concerned with here are where SKA sub-systems need to agree with one another about information - such as SDP and dishes agreeing about the beam shape, or CSP and TMC agreeing about the make-up of the correlator and its connections to links. In these cases this information evolves very slowly, and therefore it is not appropriate for the two sides to agree dynamically - instead the telescope model is used as a shared source of truth.

CHAPTER 2

# Development

The Telescope Model is developed jointly by all teams working on the SKA telescope. To make this work, all changes will have to be tested thoroughly and pass a code review via merge request.

Testing should ensure that all code paths are checked, i.e. we want to reach 100% coverage. We also aim to minimise regressions of any kind. This means that most code and data should be versioned *within* the Telescope Model, with old behaviour staying supported until a sufficient depreciation period has passed.

# CHAPTER 3

## Adding a new schema (version)

To add a new interface, you will have to adjust a number of places in the library. For a new SKA interface `<interface>` with `<elem>` as the leading sub-system, do the following steps:

1. Add

   ```
   <ELEM>_<INTERFACE>_PREFIX = "https://schema.skao.int/ska-<elem>-<interface>/"`
   ```

   to `src/ska_telmodel/<elem>/version.py`. This is the interface namespace URI.

2. Add a `get_<elem>_<interface>_schema(version :str, strict :bool)` function to `src/ska_telmodel/<elem>/schema.py`, returning an apppropriate `Schema` object. This is the main schema definition. Please add documentation as far as possible, this will be put both into the JSON schema as well as the documentation.

3. Adjust `schema_by_uri` in `src/ska_telmodel/schema.py` to call `get_<elem>_<interface>_schema` for schemas starting with `<ELEM>_<INTERFACE>_PREFIX` so that your schema can be found.

4. Add a documentation file `docs/src/ska_<elem>_<interface>.rst` with a line along the lines of

   ```
   .. ska-schema:: https://schema.skao.int/ska-<elem>-<interface>/<ver>
   ```

   to ensure documentation is generated

5. Add a test in `test_<elem>_schemas.py` to ensure test coverage. This is especially easy if you add an example to the schema (see next sub-section).

If you just want to add new schema version, skip steps (1) and (3) and extend existing definitions in the remaining steps.

CHAPTER 4

# Adding a new example

It is a good idea to always provide an up-to-date example for every schema version. Assuming the schema is defined, the steps are fairly similar:

1. Add a `get_<elem>_<interface>_example(version :str)` function to `src/ska_telmodel/<elem>/schema.py`, returning an `dict`. If you have multiple examples, you can add a `str` parameter to select the appropriate one.
2. Adjust `example_by_uri` in `src/ska_telmodel/schema.py` to call `get_<elem>_<interface>_example` for schemas starting with `<ELEM>_<INTERFACE>_PREFIX` so that your example can be found.
3. Add your example to `docs/src/ska_<elem>_<interface>.rst` by adding a line like

   ```
   .. ska-schema-example:: https://schema.skao.int/ska-<elem>-<interface>/<ver>
   ```

   inside the `.. ska-schema` block of the appropriate version

CHAPTER 5

# Code Style

This project uses the Black Code Formatter. To ensure that all code is formatted as required, run the following before you commit:

```
$ sh script/blackify.sh
```

CHAPTER 6

# Change Log

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

## 6.1 1.3.1

- Update values in example file for CSP Configure schema
- Enhance CSP Schema version check logic

## 6.2 1.3.0

- Add version 2.0 of CSP Configure schema to support standardised keys (ADR-35)
- Add version 2.0 of TMC schemas for SKA-Low to support standardised keys (ADR-35)

## 6.3 1.2.0

- Add version 0.3 of SDP schemas to support standardised keys (ADR-35)

## 6.4 1.1.0

- Introduce TMC configuration to the TMC SubArrayNode.Configure schema

## 6.5 1.0.0

- Introduced schema for TMC CentralNode and TMC SubArrayNode, currently just for SKA LOW.
- Introduced schema for MCCSController and MCCSSubarray

## 6.6 0.3.0

- Generate schema description into Sphinx documentation instead of using bootprint
- Replaces specialised validation routines by a general one that selects the schema by the URI.

## 6.7 0.2.0

- Implementation of changes in CSP configuration string according ADR-18
- Especially add stubs for PSS and PST configuration
- Rework version handling to use URIs as suggested by ADR-22

## 6.8 0.1.4

- Accept raw dictionaries instead of strings

## 6.9 0.1.3

- Added SDP schema verifications

## 6.10 0.1.2

- Added CSP schema verification

## 6.11 0.1.1

- Renamed *outputChannelOffset* to *fspChannelOffset*

## 6.12 0.1.0

- Initial release
- Added CSP interface generation

# CHAPTER 7

# Telescope Model Public API

## 7.1 ska_telmodel

Interface module for generating CSP configuration.

Handles parsing and validation of inputs and passes them on to the internal configuration functions in *config*.

ska_telmodel.csp.interface.**make_csp_config**(*csp_interface_version: Union[int, str], sdp_interface_version: Union[int, str], scan_type: str, csp_config_str: Union[str, dict], sdp_receive_addrs_map_str: Union[str, dict]*) → str

Generate a CSP scan configuration for a scan using SDP receive addresses.

This should be used right before CSP is configured so that data streams are sent to the right ingest nodes.

**Parameters**

- **csp_interface_version** – Version of CSP interface (URI)
- **sdp_interface_version** – Version of SDP interface (URI)
- **scan_type** – Type of scan to configure
- **csp_config_in** – General CSP configuration
- **sdp_receive_addrs** – Receive addresses map for scan types, generated by SDP

**Returns** A validated JSON string with CSP configuration.

**Raise** *ValueError* when the input JSON configuration fails validation.

Used for checking CSP configuration strings for conformance

ska_telmodel.csp.schema.**get_cbf_config_schema**(*version: str, strict: bool*) → schema.Schema

Central Beam Former configuration schema

**Parameters**

- **version** – Interface Version URI
- **strict** – Schema strictness

**Returns** the JSON Schema for the MID.CBF configuration.

ska_telmodel.csp.schema.**get_common_config_schema**(*version: str*, *strict: bool*) → schema.Schema

CSP Subarray common configuration schema.

**Parameters**

- **version** – Interface Version URI
- **strict** – Schema strictness

**Returns** the JSON Schema for the CSP subarray common configuration (ADR-18).

ska_telmodel.csp.schema.**get_csp_config_schema**(*version: str*, *strict: bool*) → schema.Schema

Returns a schema to verify a CSP configuration

**Parameters**

- **version** – Interface version
- **strict** – Strict mode - refuse even harmless schema violations (like extra keys). DO NOT USE FOR INPUT VALIDATION!

**Returns** The JSON Schema for the CSP configuration.

**Raise** *ValueError* exception on mismatch major version or invalid JSON Schema URI

ska_telmodel.csp.schema.**get_fsp_config_schema**(*version: str*, *strict: bool*)

Frequency slice processor configuration schema

**Parameters**

- **version** – Interface Version URI
- **strict** – Schema strictness

**Returns** the JSON schema for the MID.CBF FSP configuration.

ska_telmodel.csp.schema.**get_pss_beam_config_schema**(*version: str*, *strict: bool*) → schema.Schema

Pulsar Search Beam specific items

**Parameters**

- **version** – Interface Version URI
- **strict** – Schema strictness

**Returns** the JSON Schema for the PSS beam configuration.

ska_telmodel.csp.schema.**get_pss_config_schema**(*version: str*, *strict: bool*) → schema.Schema

Pulsar Search specific items

**Parameters**

- **version** – Interface Version URI
- **strict** – Schema strictness

**Returns** the JSON Schema for the PSS configuration.

ska_telmodel.csp.schema.**get_pst_config_schema**(*version: str*, *strict: bool*) → schema.Schema

Pulsar Timing specific items

**Parameters**

- **version** – Interface Version URI
- **strict** – Schema strictness

**Returns** the JSON Schema for the PST configuration.

ska_telmodel.csp.schema.**get_search_window_config_schema**(*version: str*, *strict: bool*) → schema.Schema

SearchWindow configuration schema

**Parameters**

- **version** – Interface Version URI
- **strict** – Schema strictness

**Returns** the JSON Schema for the MID.CBF SearchWindow configuration.

ska_telmodel.csp.schema.**get_subarray_config_schema**(*version: str*, *strict: bool*) → schema.Schema

CSP Subarray configuration schema

**Parameters**

- **version** – Interface Version URI
- **strict** – Schema strictness

**Returns** the JSON Schema for the CSP subarray specific configuraiton.

ska_telmodel.csp.schema.**get_vlbi_config_schema**(*version: str*, *strict: bool*)

VLBI specific items

**Parameters**

- **version** – Interface Version URI
- **strict** – Schema strictness

**Returns** the JSON schema for the MID.CBF VLBI configuration.

ska_telmodel.csp.schema.**use_camel_case**(*version: str*) → bool

Checks whether the given CSP schema version uses camel-case attribute names.

**param version** Interface Version URI

**returns** True or False according to schema version number

CHAPTER 8

# Telescope Model Internals

## 8.1 ska_telmodel.schema

ska_telmodel.schema.**example_by_uri**(*version: str*, **args*) → dict
Generates an example for a particular schema

**Parameters**

- **version** – Interface URI
- **args** – Extra parameters depending on interface (strings)

**Returns** Dictionary

ska_telmodel.schema.**schema_by_uri**(*version: str*, *strict: int = 1*, ***kwargs*) → schema.Schema
Looks up interface schema based on interface identifier

**Parameters**

- **version** – Interface URI
- **strict** – Strictness level

**Returns** Interface schema

ska_telmodel.schema.**validate**(*version: str*, *config: dict*, *strictness: int = 1*)
Validate a dictionary against schema

Will automatically determine the schema to check against

**Parameters**

- **version** – Interface with version
- **config** – Dictionary to validate
- **strictness** – Strictness level (0: permissive warnings, 1: permissive errors + strict warnings, 2: strict errors). DO NOT USE STRICTNESS 2 IN PRODUCTION!

## 8.2 ska_telmodel.csp.config

ska_telmodel.csp.config.**add_receive_addresses**(*scan_type: str*, *csp_config: dict*, *scan_receive_addrs: dict*, *csp_interface_version: str*) → dict

Add SDP receive addresses into CSP configuration

**Parameters**

- **scan_trype** – Scan type executed
- **csp_config** – CSP input configuration
- **sdp_receive_addrs** – SDP receive addresses for scan

**Returns** New CSP configuration

ska_telmodel.csp.config.**get_fsp_channel_offset**(*csp_config_in: dict*) → int

Determines first channel ID within an FSP

ska_telmodel.csp.config.**get_fsp_output_channel_offset**(*fsp_config: dict*, *fsp_id: str*, *fsp_ch_offset: str*) → int

Determines the FSP channel offset. Either read from the dictionary or reconstructed.

**Parameters** **fsp_config** – FSP configuration structure

## 8.3 ska_telmodel.channel_map

Tools for working with JSON compressed channel maps.

The SKA is meant to have a large number of channels, which means that any type of per-channel configuration might become very cumbersome to transfer and reason about. To prevent such issues we are using a simple run-length encoding to "compress" the representation. The idea is that if we write:

```
[ [0,0], [200,1], [400, 3] ]
```

We essentially mean the dictionary:

```
{ 0: 0, 1: 0, ..., 199:0, 200:1, ..., 399:1, 400: 3, ...}
```

Furthermore runs of numbers are supported, by adding an increment:

```
[ [0,0,1], [200,1] ]
```

Means:

```
{ 0: 0, 1: 1, 2:2, ..., 199:100, 200:1, ...}
```

ska_telmodel.channel_map.**channel_map_at**(*channel_map: List[list]*, *channel: int*, *make_entry: bool = False*) → Any

Query a value from a channel map

**Parameters**

- **channel_map** – Queried map
- **channel** – Channel ID to query
- **make_entry** – Return an channel map entry (including increment) instead of just the value

**Returns** Value from map

ska_telmodel.channel_map.**shift_channel_map**(*channel_map: List[list], channel_shift: int*) → List[list]

Shift a channel map by some channel distance

**Parameters**

- **channel_map** – Channel map to use
- **channel_shift** – Shift to apply

ska_telmodel.channel_map.**split_channel_map**(*channel_map: List[list], first_channel: int, channel_group_steps: int, rebase_groups: int = None, minimum_groups: int = 0*) → List[List[list]]

Split a channel map using a constant channel step length

**Parameters**

- **channel_map** – Channel map to split. Each entry is expected to have the start channel in the first field, and mapped data in the remaining entries
- **first_channel** – First channel to appear in the map
- **channel_group_steps** – Chunks to split the channel map into
- **rebase_groups** – Start every group at given channel index (None: left as-is)
- **minimum_groups** – Minimum number of groups to return

**Returns** List of channel maps

ska_telmodel.channel_map.**split_channel_map_at**(*channel_map: List[list], channel_groups: List[int], rebase_groups: int = None*) → List[List[list]]

Split a channel map at certain points

**Parameters**

- **channel_map** – Channel map to split. Each entry is expected to have the start channel in the first field, and mapped data in the remaining entries
- **channel_groups** – Boundaries between channel groups. The *n*-th returned channel map will cover channels *channel_groups[n]..channel_groups[n+1]-1*. Needs to have at least two entries.
- **rebase_groups** – Start every group at given channel index (None: left as-is)

**Returns** List of channel maps

CHAPTER 9

# ska-csp-configure

## 9.1 CSP config 2.1

Example (CSP PSS configuration for science_a scan)

```
{
    "interface": "https://schema.skao.int/ska-csp-configure/2.0",
    "subarray": {
        "subarray_name": "science period 23"
    },
    "common": {
        "config_id": "sbi-mvp01-20200325-00001-science_A",
        "frequency_band": "1",
        "subarray_id": 1
    },
    "cbf": {
        "fsp": [{
            "fsp_id": 1,
            "function_mode": "CORR",
            "frequency_slice_id": 1,
            "integration_factor": 1,
            "zoom_factor": 0,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 0,
            "output_link_map": [
                [0, 0],
                [200, 1]
            ],
            "output_host": [
                [0, "192.168.0.1"],
                [400, "192.168.0.2"]
```

(continues on next page)

(continued from previous page)

```
            ],
            "output_mac": [
                [0, "06-00-00-00-00-00"]
            ],
            "output_port": [
                [0, 9000, 1],
                [400, 9000, 1]
            ]
        }, {
            "fsp_id": 2,
            "function_mode": "CORR",
            "frequency_slice_id": 2,
            "integration_factor": 1,
            "zoom_factor": 1,
            "zoom_window_tuning": 650000,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 744,
            "output_link_map": [
                [0, 4],
                [200, 5]
            ],
            "output_host": [
                [0, "192.168.0.3"],
                [400, "192.168.0.4"]
            ],
            "output_mac": [
                [0, "06-00-00-00-00-01"]
            ],
            "output_port": [
                [0, 9000, 1],
                [400, 9000, 1]
            ]
        }],
        "vlbi": {}
    },
    "pst": {}
}
```

Example (TMC input)

```
{
    "interface": "https://schema.skao.int/ska-csp-configure/2.0",
    "subarray": {
        "subarray_name": "science period 23"
    },
    "common": {
        "config_id": "sbi-mvp01-20200325-00001-science_A",
        "frequency_band": "1",
        "subarray_id": 1
    },
    "cbf": {
        "fsp": [{
            "fsp_id": 1,
            "function_mode": "CORR",
```

(continues on next page)

(continued from previous page)


```
            "frequency_slice_id": 1,
            "integration_factor": 1,
            "zoom_factor": 0,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 0,
            "output_link_map": [
                [0, 0],
                [200, 1]
            ]
        }, {
            "fsp_id": 2,
            "function_mode": "CORR",
            "frequency_slice_id": 2,
            "integration_factor": 1,
            "zoom_factor": 1,
            "zoom_window_tuning": 650000,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 744,
            "output_link_map": [
                [0, 4],
                [200, 5]
            ]
        }],
        "vlbi": {}
    },
    "pst": {}
}
```


Example (CSP PSS configuration for science_a scan)


```
{
    "interface": "https://schema.skao.int/ska-csp-configure/2.0",
    "subarray": {
        "subarray_name": "science period 23"
    },
    "common": {
        "config_id": "sbi-mvp01-20200325-00001-science_A",
        "frequency_band": "1",
        "subarray_id": 1
    },
    "cbf": {
        "fsp": [{
            "fsp_id": 1,
            "function_mode": "CORR",
            "frequency_slice_id": 1,
            "integration_factor": 1,
            "zoom_factor": 0,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
```


(continues on next page)

(continued from previous page)

```
            "channel_offset": 0,
            "output_link_map": [
                [0, 0],
                [200, 1]
            ],
            "output_host": [
                [0, "192.168.0.1"],
                [400, "192.168.0.2"]
            ],
            "output_mac": [
                [0, "06-00-00-00-00-00"]
            ],
            "output_port": [
                [0, 9000, 1],
                [400, 9000, 1]
            ]
        }, {
            "fsp_id": 2,
            "function_mode": "CORR",
            "frequency_slice_id": 2,
            "integration_factor": 1,
            "zoom_factor": 1,
            "zoom_window_tuning": 650000,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 744,
            "output_link_map": [
                [0, 4],
                [200, 5]
            ],
            "output_host": [
                [0, "192.168.0.3"],
                [400, "192.168.0.4"]
            ],
            "output_mac": [
                [0, "06-00-00-00-00-01"]
            ],
            "output_port": [
                [0, 9000, 1],
                [400, 9000, 1]
            ]
        }],
        "vlbi": {}
    },
    "pst": {}
}
```

Example (CSP PSS configuration for cal_a scan)

```
{
    "interface": "https://schema.skao.int/ska-csp-configure/2.0",
    "subarray": {
        "subarray_name": "science period 23"
    },
    "common": {
```

(continues on next page)

(continued from previous page)

```
        "config_id": "sbi-mvp01-20200325-00001-science_A",
        "frequency_band": "1",
        "subarray_id": 1
    },
    "cbf": {
        "fsp": [{
            "fsp_id": 1,
            "function_mode": "CORR",
            "frequency_slice_id": 1,
            "integration_factor": 1,
            "zoom_factor": 0,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 0,
            "output_link_map": [
                [0, 0],
                [200, 1]
            ],
            "output_host": [
                [0, "192.168.1.1"]
            ],
            "output_port": [
                [0, 9000, 1]
            ]
        }, {
            "fsp_id": 2,
            "function_mode": "CORR",
            "frequency_slice_id": 2,
            "integration_factor": 1,
            "zoom_factor": 1,
            "zoom_window_tuning": 650000,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 744,
            "output_link_map": [
                [0, 4],
                [200, 5]
            ],
            "output_host": [
                [0, "192.168.1.1"]
            ],
            "output_port": [
                [0, 9744, 1]
            ]
        }],
        "vlbi": {}
    },
    "pst": {}
}
```

| | | |
|---|---|---|
| https://schema.skao.int/ska-csp-configure/2.1 | | |
| type | *object* | |
| properties | | |
| • **interface** | type | *string* |
| • **subarray** | subarray section, containing the parameters relevant only for the current sub-array device. This section is not forwarded to any subelement. | |
| | type | *object* |
| | properties | |
| | • **subarray_name** | Name and scope of current subarray the sub-array. |
| | | type *string* |
| | additionalProperties | False |
| • **common** | Common section, containing the parameters and the sections belonging to all CSP sub elements. This sections is forwarded to all sub-elements. | |
| | *Common CSP config 2.1* | |
| • **cbf** | Central Beam Former specific parameters. This section contains the parameters relevant only for CBF sub-element. This section is forwarded only to CBF subelement. Most of it to be borrowed from IICD | |
| | *CBF config 2.1* | |
| • pss | *PSS configuration 2.1* | |
| • pst | Pulsar Timing specific parameters. To be borrowed from IICD This section contains the parameters relevant only for PST. This section is forwarded only to PST subelement. | |
| | *PST configuration 2.1* | |
| additionalProperties | False | |

### 9.1.1 Common CSP config 2.1

Common section, containing the parameters and the sections belonging to all CSP sub elements. This sections is forwarded to all sub-elements.

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • config_id | type | *string* |
| • **frequency_band** | Frequency band applies for all the receptors (VCCs) that belong to the sub-array. | |
| | type | *string* |
| | pattern | ^(1\|2\|3\|4\|5(a\|b))$ |
| • band_5_tuning | Center frequency for the Band-of-Interest. Required if Band is 5a or 5b; not specified for other Bands (not configurable for Band 1, 2, 3 and 4).<br>Input for Band 5a and 5b consists of two 2.5 GHz streams; the center frequency can be independently tuned for each stream.<br>The following nomenclature is used to refer to Band 5a and 5b streams: 5a1, 5a2, 5b1, 5b2. | |
| | type | *array* |
| | items | type *number* |
| • **subarray_id** | Subarray number | |
| | type | *integer* |
| additionalProperties | False | |

### 9.1.2 CBF config 2.1

Central Beam Former specific parameters. This section contains the parameters relevant only for CBF sub-element. This section is forwarded only to CBF subelement. Most of it to be borrowed from IICD

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • frequency_band_offset_stream1 | Optionally, an offset can be specified so that the entire observed band is shifted (to accommodate a Zoom Window that crosses a 'natural' Frequency Slice boundary). If specified, applies for all the receptors in the sub-array. Bands 1, 2, 3 and 4: input from the receptor consists of a single data stream; the Frequency Band Offset (FBO) should be specified for Stream 1 only. Bands 5a and 5b: input from the receptor consists of two data streams; the FBO can be specified for each stream independently. Note: For Band 5a and 5b the frequency shift is performed by the receptor (DISH). Note: This is optional and does not need to be implemented in PI3, but would be great for demo; if Team Buttons is looking for opportunities to showcase interesting GUIs, Zoom Windows are perfect opportunity (would require TMC and CSP to support these two parameters, corrBandwidth values > 0 and zoom window tuning.) | |
| | type | *integer* |
| • frequency_band_offset_stream2 | See *frequencyBandOffsetStream1* | |
| | type | *integer* |
| • delay_model_subscription_point | FQDN of TMC.DelayModel TANGO attribute which exposes delay values for all the dishes assigned to a Subarray in JSON format. Delay values are updated every 10 seconds. | |
| | type | *string* |
| • doppler_phase_corr_subscription_point | The same model applies for all receptors that belong to the subarray. Delivered by TMC using publish-subscribe mechanism (see ICD Section 3.8.8.5.3). The Doppler phase correction, by default, applies only to the CSP_Mid Processing Mode Correlation; optionally may apply to other Processing Modes as well. | |
| | type | *string* |
| • rfi_flagging_mask | Specified as needed in advance of the scan start and/or during the scan. Delivered using publish-subscribe mechanism (see ICD Section 3.8.8.5.7). | |
| | type | *object* |
| | properties | |
| | additionalProperties | False |
| • **fsp** | type | *array* |
| | items | *FSP config 2.1* |
| • vlbi | Very Long Baseline Interferometry specific parameters. To be borrowed from IICD This section contains the parameters relevant only for VLBI. This section is forwarded only to CSP subelement. | |
| | *VLBI config 2.1* | |
| • search_window | type | *array* |
| | items | Up to two 300 MHz Search Windows can be optionally configured and used as input for Transient Data Capture and/or Pulsar Search beamforming. |
| | | *Search window config 2.1* |
| additionalProperties | False | |

### 9.1.3 FSP config 2.1

| | | | | |
|---|---|---|---|---|
| type | *object* | | | |
| properties | | | | |
| • **fsp_id** | type | *integer* | | |
| • **function_mode** | allOf | type | *string* | |
| | | enum | CORR, PSS-BF, PST-BF, VLBI | |
| • receptors | Optionally a subset of receptors to be correlated can be specified. If not specified, all receptors that belong to the subarray are cross-correlated (i.e. visibilities for all the baselines in the subarray are generated and transmitted to SDP). | | | |
| | type | *array* | | |
| | items | anyOf | type | *integer* |
| | | | type | *string* |
| • **frequency_slice_id** | Frequency Slice to be processed on this FSP (valid range depends on the Frequency Band). | | | |
| | type | *integer* | | |
| • **zoom_factor** | Bandwidth to be correlated calculated as FSBW/2n, where n is in range [0..6]. When n=0 the full Frequency Slice bandwidth is correlated. BW > 0 implies 'Zoom Window' configuration; the spectral Zoom Window tuning must be specified. | | | |
| | type | *integer* | | |
| • zoom_window_tuning | The Zoom Window tuning provided in absolute terms as RF center frequency. Based on that, CSP_Mid calculates tuning within the data stream received from the receptor. Must be selected so that the entire Zoom Window is within the Frequency Slice. If partially out of the FS a warning is generated. If completely outside of the FS an exception is generated. Step size <= 0.01MHz. The Frequency Band Offset can be used to shift the entire observed band in order to accommodate a Zoom Window that spans across a Frequency Slice boundary. | | | |
| | type | *integer* | | |
| • **integration_factor** | Integration time for the correlation products, defines multiple of 140 milliseconds. | | | |
| | type | *integer* | | |
| • channel_averaging_map | Table of up to 20 x 2 integers. Each of entries contains: * Start channel ID, and * averaging factor. Explanation: Each FSP produces 14880 (TBC) fine channels across the correlated bandwidth (Frequency Slice or Zoom Window). Channels are evenly spaced in frequency. TM shall provide the table that for each FSP and each group of 744 channels (there are 20 groups per FSP) indicates the channel averaging factor. More precisely, for each group the TMC provided table specifies: * the channel ID (integer) of the first channel, and * the averaging factor, as follows: 0 means do not send channels to SDP, 1 means no averaging, 2 means average two adjacent channels, 3 means average three adjacent channels, and so on. If no entry is present for an FSP, the averaging settings of the previous FSP are still applicable. | | | |
| | type | *array* | | |
| | items | type | *array* | |
| | | items | type | *integer* |

Continued on next page

Table 1 – continued from previous page

| | | | | |
|---|---|---|---|---|
| • channel_offset | Channel ID to use for visibilities of the first channel produced by this FSP. For example, if the channel offset is 5000 the first channel group would span IDs 5000-5743.<br>Note that this offset does not apply to channel maps in this structure (such as *channelAveragingMap* or *outputHost*). | | | |
| | type | *integer* | | |
| • output_link_map | Output links to emit visibilities on for every channel, given as a list of start channel ID to link ID. Where no value is given for concrete channel, the previous value should be used. | | | |
| | type | *array* | | |
| | items | type | *array* | |
| | | items | anyOf | type *integer*<br>type *string* |
| • output_host | Output host to send visibilities to for every channel, given as a list of start channel ID to host IP addresses in dot-decimal notation. Where no value is given for a concrete channel, the previous value should be used. | | | |
| | type | *array* | | |
| | items | type | *array* | |
| | | items | anyOf | type *integer*<br>type *string* |
| • output_port | Output port to send visibilities to for every channel, given as a list of start channel ID to port number. Where no value is given for a concrete channel, the previous value should be used. | | | |
| | type | *array* | | |
| | items | type | *array* | |
| | | items | type | *integer* |
| • output_mac | Output MAC address to send visibilities to for every channel, given as a list of start channel ID to IEEE 802 MAC addresses. Where no value is given for a concrete channel, the previous value should be used. | | | |
| | type | *array* | | |
| | items | type | *array* | |
| | | items | anyOf | type *integer*<br>type *string* |
| additionalProperties | False | | | |

### 9.1.4 VLBI config 2.1

Very Long Baseline Interferometry specific parameters. To be borrowed from IICD This section contains the parameters relevant only for VLBI. This section is forwarded only to CSP subelement.

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • dummy_param | type | *string* |
| additionalProperties | False | |

### 9.1.5 Search window config 2.1

Up to two 300 MHz Search Windows can be optionally configured and used as input for Transient Data Capture and/or Pulsar Search beam-forming.

| | | | | |
|---|---|---|---|---|
| type | *object* | | | |
| properties | | | | |
| • **search_window_id** | Identifier of the 300MHz Search Window. Unique within a sub-array. | | | |
| | type | *integer* | | |
| • **search_window_tuning** | The Search Window tuning is provided in absolute terms as RF center frequency. The Search Window must be placed within the observed band. If partially out of the observed Band a warning is generated. If completely outside of the observed Band an exception is generated. | | | |
| | type | *integer* | | |
| • **tdc_enable** | Enable / disable Transient Data Capturefor the Search Window. | | | |
| | type | *boolean* | | |
| • tdc_num_bits | Number of bits per sample (for the Transient Data Capture). Required if TDC is enabled, otherwise not specified. | | | |
| | type | *integer* | | |
| • tdc_period_before_epoch | Users can trade the period of time for which data are saved and transmitted for the sample bit-width and/or the number of Search Windows. The exact information regarding the memory capacity per receptor and supported range will be provided in construction.<br>The epoch is specified in the command that triggers TDC off-loading (transmission of data). | | | |
| | type | *integer* | | |
| • tdc_period_after_epoch | see *tdcPeriodBeforeEpoch* | | | |
| | type | *integer* | | |
| • tdc_destination_address | Destination addresses (MAC, IP, port) for off-loading of the content of the Transient Data Capture Buffer, specified per receptor. The destination addresses for the content of the Transient Data Capture can be provided either as a part of the scan configuration or by the command that triggers transmission of the captured data. The latter, if provided, overrides previously set addresses.<br>Required if TDC is enabled, otherwise not specified. | | | |
| | type | *array* | | |
| | items | anyOf | type | *integer* |
| | | | type | *string* |
| additionalProperties | False | | | |

### 9.1.6 PSS configuration 2.1

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **beam_bandwidth** | Beam bandwidth (MHz) | |
| | type | *integer* |
| • **channels_per_beam** | Number of channels per beam | |
| | type | *integer* |
| • **acceleration_search** | Processing Mode: Acceleration Search (a.k.a. Pulsar Search) and Single Pulse Search (a.k.a. Transient Search) can be performed concurrently. | |
| | type | *boolean* |
| • **single_pulse_search** | Processing Mode: Acceleration Search (a.k.a. Pulsar Search) and Single Pulse Search (a.k.a. Transient Search) can be performed concurrently. | |
| | type | *boolean* |
| • **integration_time** | Scan duration. | |
| | type | *integer* |
| • acc_range | Range in source acceleration to be searched. | |
| | type | *integer* |

Continued on next page

Table 2 – continued from previous page

| | | | |
|---|---|---|---|
| • **number_of_trials** | Number of trials to be performed. | | |
| | type | *integer* | |
| • **time_resolution** | Time resolution of input data. | | |
| | type | *integer* | |
| • **ps_dm** | Dispersion corretion for acceleration search. | | |
| | type | *number* | |
| • **sps_dm** | Dispersion corretion for transient search. | | |
| | type | *number* | |
| • **timesample_per_block** | Number of time samples in each block of data. | | |
| | type | *integer* | |
| • **sub_bands** | Number of frequency band groups summed up during folding. | | |
| | type | *integer* | |
| • **buffer_size** | Size of the buffer receiving raw data. (2**buffer_size) | | |
| | type | *integer* | |
| • **hsum_control** | Number of the “harmonic folds” on the initial Fourier power-spectrum summed up. | | |
| | type | *integer* | |
| • **cxft_control** | CXFT control parameters. | | |
| | type | *object* | |
| • **cand_sift** | Constraints on matches between candidates. | | |
| | type | *object* | |
| • **cand_output** | Define data sinks and subscriber to be notified. | | |
| | type | *object* | |
| • **sp_threshold** | Threshold for a single pulse trigger. (Tuned to system noise and RFI env.) | | |
| | type | *number* | |
| • **sp_opt_pars** | Single pulse optimization parameters. | | |
| | type | *object* | |
| • **dred_beam_stats** | DRED: statistics of spectra to derive the normalization factors. | | |
| | type | *object* | |
| • **cdos_control** | CDOS: control parameters and related statistical data. | | |
| | type | *object* | |
| • **rfim_control** | RFIM control parameters. | | |
| | type | *object* | |
| • **fldo_control** | FLDO control parameters. | | |
| | type | *object* | |
| | properties | | |
| | • **phase_split** | type | *boolean* |
| | • **channel_scale** | type | *boolean* |
| | • **max_phases** | type | *integer* |
| | additionalProperties | False | |
| • **beam** | type | *array* | |
| | items | *PSS beam config 2.1* | |
| additionalProperties | False | | |

### 9.1.7 PSS beam config 2.1

| | | | |
|---|---|---|---|
| type | *object* | | |
| properties | | | |
| • **beam_id** | Search Beam ID. | | |
| | type | *integer* | |
| • ra | Right Ascension of sub-array beam target, in degrees. | | |
| | type | *number* | |
| • dec | Declination of sub-array beam target, in degrees. | | |
| | type | *number* | |
| • reference_frame | reference frame for pointing coordinates | | |
| | allOf | type | *string* |
| | | enum | ICRS, HORIZON |
| • **centre_frequency** | Centre frequency of the search beam. | | |
| | type | *number* | |
| • **beam_delay_centre** | Beam delay center, relative to the array delay center. | | |
| | anyOf | type | *number* |
| | | type | *string* |
| • dest_host | Per beam destination host address for PSS output. | | |
| | type | *string* | |
| • dest_port | Per beam destination port for PSS output. | | |
| | type | *integer* | |
| additionalProperties | False | | |

### 9.1.8 PST configuration 2.1

Pulsar Timing specific parameters. To be borrowed from IICD This section contains the parameters relevant only for PST. This section is forwarded only to PST subelement.

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • dummy_param | type | *string* |
| additionalProperties | False | |

## 9.2 CSP config 2.0

Example (TMC input)

```
{
    "interface": "https://schema.skao.int/ska-csp-configure/2.0",
    "subarray": {
        "subarray_name": "science period 23"
    },
    "common": {
        "config_id": "sbi-mvp01-20200325-00001-science_A",
        "frequency_band": "1",
        "subarray_id": 1
    },
```

(continues on next page)

(continued from previous page)

```
    "cbf": {
        "fsp": [{
            "fsp_id": 1,
            "function_mode": "CORR",
            "frequency_slice_id": 1,
            "integration_factor": 1,
            "zoom_factor": 0,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 0,
            "output_link_map": [
                [0, 0],
                [200, 1]
            ]
        }, {
            "fsp_id": 2,
            "function_mode": "CORR",
            "frequency_slice_id": 2,
            "integration_factor": 1,
            "zoom_factor": 1,
            "zoom_window_tuning": 650000,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 744,
            "output_link_map": [
                [0, 4],
                [200, 5]
            ]
        }],
        "vlbi": {}
    },
    "pst": {}
}
```

Example (CSP configuration for science_a scan)

```
{
    "interface": "https://schema.skao.int/ska-csp-configure/2.0",
    "subarray": {
        "subarray_name": "science period 23"
    },
    "common": {
        "config_id": "sbi-mvp01-20200325-00001-science_A",
        "frequency_band": "1",
        "subarray_id": 1
    },
    "cbf": {
        "fsp": [{
            "fsp_id": 1,
            "function_mode": "CORR",
            "frequency_slice_id": 1,
            "integration_factor": 1,
            "zoom_factor": 0,
```

(continues on next page)

(continued from previous page)

```
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 0,
            "output_link_map": [
                [0, 0],
                [200, 1]
            ],
            "output_host": [
                [0, "192.168.0.1"],
                [400, "192.168.0.2"]
            ],
            "output_mac": [
                [0, "06-00-00-00-00-00"]
            ],
            "output_port": [
                [0, 9000, 1],
                [400, 9000, 1]
            ]
        }, {
            "fsp_id": 2,
            "function_mode": "CORR",
            "frequency_slice_id": 2,
            "integration_factor": 1,
            "zoom_factor": 1,
            "zoom_window_tuning": 650000,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 744,
            "output_link_map": [
                [0, 4],
                [200, 5]
            ],
            "output_host": [
                [0, "192.168.0.3"],
                [400, "192.168.0.4"]
            ],
            "output_mac": [
                [0, "06-00-00-00-00-01"]
            ],
            "output_port": [
                [0, 9000, 1],
                [400, 9000, 1]
            ]
        }],
        "vlbi": {}
    },
    "pst": {}
}
```

Example (CSP configuration for cal_a scan)

```
{
    "interface": "https://schema.skao.int/ska-csp-configure/2.0",
```

(continues on next page)

(continued from previous page)


```
    "subarray": {
        "subarray_name": "science period 23"
    },
    "common": {
        "config_id": "sbi-mvp01-20200325-00001-science_A",
        "frequency_band": "1",
        "subarray_id": 1
    },
    "cbf": {
        "fsp": [{
            "fsp_id": 1,
            "function_mode": "CORR",
            "frequency_slice_id": 1,
            "integration_factor": 1,
            "zoom_factor": 0,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 0,
            "output_link_map": [
                [0, 0],
                [200, 1]
            ],
            "output_host": [
                [0, "192.168.1.1"]
            ],
            "output_port": [
                [0, 9000, 1]
            ]
        }, {
            "fsp_id": 2,
            "function_mode": "CORR",
            "frequency_slice_id": 2,
            "integration_factor": 1,
            "zoom_factor": 1,
            "zoom_window_tuning": 650000,
            "channel_averaging_map": [
                [0, 2],
                [744, 0]
            ],
            "channel_offset": 744,
            "output_link_map": [
                [0, 4],
                [200, 5]
            ],
            "output_host": [
                [0, "192.168.1.1"]
            ],
            "output_port": [
                [0, 9744, 1]
            ]
        }],
        "vlbi": {}
    },
    "pst": {}
}
```

| | | |
|---|---|---|
| https://schema.skao.int/ska-csp-configure/2.0 | | |
| type | *object* | |
| properties | | |
| • **interface** | type | *string* |
| • **subarray** | subarray section, containing the parameters relevant only for the current sub-array device. This section is not forwarded to any subelement. | |
| | type | *object* |
| | properties | |
| | • **subarray_name** | Name and scope of current subarray the sub-array. |
| | | type *string* |
| | additionalProperties | False |
| • **common** | Common section, containing the parameters and the sections belonging to all CSP sub elements. This sections is forwarded to all sub-elements. | |
| | *Common CSP config 2.0* | |
| • **cbf** | Central Beam Former specific parameters. This section contains the parameters relevant only for CBF sub-element. This section is forwarded only to CBF subelement. Most of it to be borrowed from IICD | |
| | *CBF config 2.0* | |
| • pss | *PSS configuration 2.0* | |
| • pst | Pulsar Timing specific parameters. To be borrowed from IICD This section contains the parameters relevant only for PST. This section is forwarded only to PST subelement. | |
| | *PST configuration 2.0* | |
| additionalProperties | False | |

### 9.2.1 Common CSP config 2.0

Common section, containing the parameters and the sections belonging to all CSP sub elements. This sections is forwarded to all sub-elements.

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • config_id | type | *string* |
| • **frequency_band** | Frequency band applies for all the receptors (VCCs) that belong to the sub-array. | |
| | type | *string* |
| | pattern | ^(1\|2\|3\|4\|5(a\|b))$ |
| • band_5_tuning | Center frequency for the Band-of-Interest. Required if Band is 5a or 5b; not specified for other Bands (not configurable for Band 1, 2, 3 and 4).<br>Input for Band 5a and 5b consists of two 2.5 GHz streams; the center frequency can be independently tuned for each stream.<br>The following nomenclature is used to refer to Band 5a and 5b streams: 5a1, 5a2, 5b1, 5b2. | |
| | type | *array* |
| | items | type *number* |
| • **subarray_id** | Subarray number | |
| | type | *integer* |
| additionalProperties | False | |

### 9.2.2 CBF config 2.0

Central Beam Former specific parameters. This section contains the parameters relevant only for CBF sub-element. This section is forwarded only to CBF subelement. Most of it to be borrowed from IICD

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • frequency_band_offset_stream1 | Optionally, an offset can be specified so that the entire observed band is shifted (to accommodate a Zoom Window that crosses a 'natural' Frequency Slice boundary). If specified, applies for all the receptors in the sub-array. Bands 1, 2, 3 and 4: input from the receptor consists of a single data stream; the Frequency Band Offset (FBO) should be specified for Stream 1 only. Bands 5a and 5b: input from the receptor consists of two data streams; the FBO can be specified for each stream independently. Note: For Band 5a and 5b the frequency shift is performed by the receptor (DISH). Note: This is optional and does not need to be implemented in PI3, but would be great for demo; if Team Buttons is looking for opportunities to showcase interesting GUIs, Zoom Windows are perfect opportunity (would require TMC and CSP to support these two parameters, corrBandwidth values > 0 and zoom window tuning.) | |
| | type | *integer* |
| • frequency_band_offset_stream2 | See *frequencyBandOffsetStream1* | |
| | type | *integer* |
| • delay_model_subscription_point | FQDN of TMC.DelayModel TANGO attribute which exposes delay values for all the dishes assigned to a Subarray in JSON format. Delay values are updated every 10 seconds. | |
| | type | *string* |
| • doppler_phase_corr_subscription_point | The same model applies for all receptors that belong to the subarray. Delivered by TMC using publish-subscribe mechanism (see ICD Section 3.8.8.5.3). The Doppler phase correction, by default, applies only to the CSP_Mid Processing Mode Correlation; optionally may apply to other Processing Modes as well. | |
| | type | *string* |
| • rfi_flagging_mask | Specified as needed in advance of the scan start and/or during the scan. Delivered using publish-subscribe mechanism (see ICD Section 3.8.8.5.7). | |
| | type | *object* |
| | properties | |
| | additionalProperties | False |
| • **fsp** | type | *array* |
| | items | *FSP config 2.0* |
| • vlbi | Very Long Baseline Interferometry specific parameters. To be borrowed from IICD This section contains the parameters relevant only for VLBI. This section is forwarded only to CSP subelement. | |
| | *VLBI config 2.0* | |
| • search_window | type | *array* |
| | items | Up to two 300 MHz Search Windows can be optionally configured and used as input for Transient Data Capture and/or Pulsar Search beamforming. |
| | | *Search window config 2.0* |
| additionalProperties | False | |

### 9.2.3 FSP config 2.0

| | | | | |
|---|---|---|---|---|
| type | *object* | | | |
| properties | | | | |
| • **fsp_id** | type | *integer* | | |
| • **function_mode** | allOf | type | *string* | |
| | | enum | CORR, PSS-BF, PST-BF, VLBI | |
| • receptors | Optionally a subset of receptors to be correlated can be specified. If not specified, all receptors that belong to the subarray are cross-correlated (i.e. visibilities for all the baselines in the subarray are generated and transmitted to SDP). | | | |
| | type | *array* | | |
| | items | anyOf | type | *integer* |
| | | | type | *string* |
| • **frequency_slice_id** | Frequency Slice to be processed on this FSP (valid range depends on the Frequency Band). | | | |
| | type | *integer* | | |
| • **zoom_factor** | Bandwidth to be correlated calculated as FSBW/2n, where n is in range [0..6].<br>When n=0 the full Frequency Slice bandwidth is correlated.<br>BW > 0 implies 'Zoom Window' configuration; the spectral Zoom Window tuning must be specified. | | | |
| | type | *integer* | | |
| • zoom_window_tuning | The Zoom Window tuning provided in absolute terms as RF center frequency. Based on that, CSP_Mid calculates tuning within the data stream received from the receptor. Must be selected so that the entire Zoom Window is within the Frequency Slice. If partially out of the FS a warning is generated. If completely outside of the FS an exception is generated.<br>Step size <= 0.01MHz.<br>The Frequency Band Offset can be used to shift the entire observed band in order to accommodate a Zoom Window that spans across a Frequency Slice boundary. | | | |
| | type | *integer* | | |
| • **integration_factor** | Integration time for the correlation products, defines multiple of 140 milliseconds. | | | |
| | type | *integer* | | |
| • channel_averaging_map | Table of up to 20 x 2 integers. Each of entries contains: * Start channel ID, and * averaging factor.<br>Explanation: Each FSP produces 14880 (TBC) fine channels across the correlated bandwidth (Frequency Slice or Zoom Window). Channels are evenly spaced in frequency.<br>TM shall provide the table that for each FSP and each group of 744 channels (there are 20 groups per FSP) indicates the channel averaging factor. More precisely, for each group the TMC provided table specifies: * the channel ID (integer) of the first channel, and * the averaging factor, as follows:<br>0 means do not send channels to SDP,<br>1 means no averaging,<br>2 means average two adjacent channels,<br>3 means average three adjacent channels,<br>and so on.<br>If no entry is present for an FSP, the averaging settings of the previous FSP are still applicable. | | | |
| | type | *array* | | |
| | items | type | *array* | |
| | | items | type | *integer* |

Continued on next page

Table 3 – continued from previous page

| | | | | | |
|---|---|---|---|---|---|
| • channel_offset | Channel ID to use for visibilities of the first channel produced by this FSP. For example, if the channel offset is 5000 the first channel group would span IDs 5000-5743.<br>Note that this offset does not apply to channel maps in this structure (such as *channelAveragingMap* or *outputHost*). | | | | |
| | type | *integer* | | | |
| • output_link_map | Output links to emit visibilities on for every channel, given as a list of start channel ID to link ID. Where no value is given for concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | anyOf | type | *integer* |
| | | | | type | *string* |
| • output_host | Output host to send visibilities to for every channel, given as a list of start channel ID to host IP addresses in dot-decimal notation. Where no value is given for a concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | anyOf | type | *integer* |
| | | | | type | *string* |
| • output_port | Output port to send visibilities to for every channel, given as a list of start channel ID to port number. Where no value is given for a concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | type | *integer* | |
| • output_mac | Output MAC address to send visibilities to for every channel, given as a list of start channel ID to IEEE 802 MAC addresses. Where no value is given for a concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | anyOf | type | *integer* |
| | | | | type | *string* |
| additionalProperties | False | | | | |

### 9.2.4 VLBI config 2.0

Very Long Baseline Interferometry specific parameters. To be borrowed from IICD This section contains the parameters relevant only for VLBI. This section is forwarded only to CSP subelement.

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • dummy_param | type | *string* |
| additionalProperties | False | |

### 9.2.5 Search window config 2.0

Up to two 300 MHz Search Windows can be optionally configured and used as input for Transient Data Capture and/or Pulsar Search beam-forming.

| | | | | |
|---|---|---|---|---|
| type | *object* | | | |
| properties | | | | |
| • **search_window_id** | Identifier of the 300MHz Search Window. Unique within a sub-array. | | | |
| | type | *integer* | | |
| • **search_window_tuning** | The Search Window tuning is provided in absolute terms as RF center frequency. The Search Window must be placed within the observed band. If partially out of the observed Band a warning is generated. If completely outside of the observed Band an exception is generated. | | | |
| | type | *integer* | | |
| • **tdc_enable** | Enable / disable Transient Data Capturefor the Search Window. | | | |
| | type | *boolean* | | |
| • tdc_num_bits | Number of bits per sample (for the Transient Data Capture). Required if TDC is enabled, otherwise not specified. | | | |
| | type | *integer* | | |
| • tdc_period_before_epoch | Users can trade the period of time for which data are saved and transmitted for the sample bit-width and/or the number of Search Windows. The exact information regarding the memory capacity per receptor and supported range will be provided in construction.<br>The epoch is specified in the command that triggers TDC off-loading (transmission of data). | | | |
| | type | *integer* | | |
| • tdc_period_after_epoch | see *tdcPeriodBeforeEpoch* | | | |
| | type | *integer* | | |
| • tdc_destination_address | Destination addresses (MAC, IP, port) for off-loading of the content of the Transient Data Capture Buffer, specified per receptor. The destination addresses for the content of the Transient Data Capture can be provided either as a part of the scan configuration or by the command that triggers transmission of the captured data. The latter, if provided, overrides previously set addresses.<br>Required if TDC is enabled, otherwise not specified. | | | |
| | type | *array* | | |
| | items | anyOf | type | *integer* |
| | | | type | *string* |
| additionalProperties | False | | | |

### 9.2.6 PSS configuration 2.0

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • dummy_param | type | *string* |
| additionalProperties | False | |

### 9.2.7 PST configuration 2.0

Pulsar Timing specific parameters. To be borrowed from IICD This section contains the parameters relevant only for PST. This section is forwarded only to PST subelement.

| type | *object* | |
|---|---|---|
| properties | | |
| • dummy_param | type | *string* |
| additionalProperties | False | |

## 9.3 CSP config 1.0

Example (TMC input)

```
{
    "interface": "https://schema.skatelescope.org/ska-csp-configure/1.0",
    "subarray": {
        "subarrayName": "science period 23"
    },
    "common": {
        "id": "sbi-mvp01-20200325-00001-science_A",
        "frequencyBand": "1",
        "subarrayID": 1
    },
    "cbf": {
        "fsp": [{
            "fspID": 1,
            "functionMode": "CORR",
            "frequencySliceID": 1,
            "integrationTime": 1400,
            "corrBandwidth": 0,
            "channelAveragingMap": [
                [0, 2],
                [744, 0]
            ],
            "fspChannelOffset": 0,
            "outputLinkMap": [
                [0, 0],
                [200, 1]
            ]
        }, {
            "fspID": 2,
            "functionMode": "CORR",
            "frequencySliceID": 2,
            "integrationTime": 1400,
            "corrBandwidth": 0,
            "channelAveragingMap": [
                [0, 2],
                [744, 0]
            ],
            "fspChannelOffset": 744,
            "outputLinkMap": [
                [0, 4],
                [200, 5]
            ]
        }],
        "vlbi": {}
```

(continues on next page)

(continued from previous page)

```
    }
}
```

Example (CSP configuration for science_a scan)

```
{
    "interface": "https://schema.skatelescope.org/ska-csp-configure/1.0",
    "subarray": {
        "subarrayName": "science period 23"
    },
    "common": {
        "id": "sbi-mvp01-20200325-00001-science_A",
        "frequencyBand": "1",
        "subarrayID": 1
    },
    "cbf": {
        "fsp": [{
            "fspID": 1,
            "functionMode": "CORR",
            "frequencySliceID": 1,
            "integrationTime": 1400,
            "corrBandwidth": 0,
            "channelAveragingMap": [
                [0, 2],
                [744, 0]
            ],
            "fspChannelOffset": 0,
            "outputLinkMap": [
                [0, 0],
                [200, 1]
            ],
            "outputHost": [
                [0, "192.168.0.1"],
                [400, "192.168.0.2"]
            ],
            "outputMac": [
                [0, "06-00-00-00-00-00"]
            ],
            "outputPort": [
                [0, 9000, 1],
                [400, 9000, 1]
            ]
        }, {
            "fspID": 2,
            "functionMode": "CORR",
            "frequencySliceID": 2,
            "integrationTime": 1400,
            "corrBandwidth": 0,
            "channelAveragingMap": [
                [0, 2],
                [744, 0]
            ],
            "fspChannelOffset": 744,
            "outputLinkMap": [
                [0, 4],
                [200, 5]
            ],
```

(continues on next page)

(continued from previous page)

```
            "outputHost": [
                [0, "192.168.0.3"],
                [400, "192.168.0.4"]
            ],
            "outputMac": [
                [0, "06-00-00-00-00-01"]
            ],
            "outputPort": [
                [0, 9000, 1],
                [400, 9000, 1]
            ]
        }],
        "vlbi": {}
    }
}
```

Example (CSP configuration for cal_a scan)

```
{
    "interface": "https://schema.skatelescope.org/ska-csp-configure/1.0",
    "subarray": {
        "subarrayName": "science period 23"
    },
    "common": {
        "id": "sbi-mvp01-20200325-00001-science_A",
        "frequencyBand": "1",
        "subarrayID": 1
    },
    "cbf": {
        "fsp": [{
            "fspID": 1,
            "functionMode": "CORR",
            "frequencySliceID": 1,
            "integrationTime": 1400,
            "corrBandwidth": 0,
            "channelAveragingMap": [
                [0, 2],
                [744, 0]
            ],
            "fspChannelOffset": 0,
            "outputLinkMap": [
                [0, 0],
                [200, 1]
            ],
            "outputHost": [
                [0, "192.168.1.1"]
            ],
            "outputPort": [
                [0, 9000, 1]
            ]
        }, {
            "fspID": 2,
            "functionMode": "CORR",
            "frequencySliceID": 2,
            "integrationTime": 1400,
            "corrBandwidth": 0,
            "channelAveragingMap": [
```

(continues on next page)

(continued from previous page)

```
                [0, 2],
                [744, 0]
            ],
            "fspChannelOffset": 744,
            "outputLinkMap": [
                [0, 4],
                [200, 5]
            ],
            "outputHost": [
                [0, "192.168.1.1"]
            ],
            "outputPort": [
                [0, 9744, 1]
            ]
        }],
        "vlbi": {}
    }
}
```

| https://schema.skatelescope.org/ska-csp-configure/1.0 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **subarray** | subarray section, containing the parameters relevant only for the current sub-array device. This section is not forwarded to any subelement. | |
| | type | *object* |
| | properties | |
| | • **subarrayName** | Name and scope of current subarray the sub-array. |
| | | type: *string* |
| | additionalProperties | False |
| • **common** | Common section, containing the parameters and the sections belonging to all CSP sub elements. This sections is forwarded to all sub-elements. | |
| | *Common CSP config 1.0* | |
| • **cbf** | Central Beam Former specific parameters. This section contains the parameters relevant only for CBF sub-element. This section is forwarded only to CBF subelement. Most of it to be borrowed from IICD | |
| | *CBF config 1.0* | |
| • pss | *PSS configuration 1.0* | |
| • pst | Pulsar Timing specific parameters. To be borrowed from IICD This section contains the parameters relevant only for PST. This section is forwarded only to PST subelement. | |
| | *PST configuration 1.0* | |
| additionalProperties | False | |

### 9.3.1 Common CSP config 1.0

Common section, containing the parameters and the sections belonging to all CSP sub elements. This sections is forwarded to all sub-elements.

| | | | |
|---|---|---|---|
| type | *object* | | |
| properties | | | |
| • id | type | *string* | |
| • **frequencyBand** | Frequency band applies for all the receptors (VCCs) that belong to the sub-array. | | |
| | type | *string* | |
| | pattern | ^(1\|2\|3\|4\|5(a\|b))$ | |
| • band5Tuning | Center frequency for the Band-of-Interest. Required if Band is 5a or 5b; not specified for other Bands (not configurable for Band 1, 2, 3 and 4).<br>Input for Band 5a and 5b consists of two 2.5 GHz streams; the center frequency can be independently tuned for each stream.<br>The following nomenclature is used to refer to Band 5a and 5b streams: 5a1, 5a2, 5b1, 5b2. | | |
| | type | *array* | |
| | items | type | *number* |
| • **subarrayID** | Subarray number | | |
| | type | *integer* | |
| additionalProperties | False | | |

### 9.3.2 CBF config 1.0

Central Beam Former specific parameters. This section contains the parameters relevant only for CBF sub-element. This section is forwarded only to CBF subelement. Most of it to be borrowed from IICD

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • frequencyBandOffsetStream1 | Optionally, an offset can be specified so that the entire observed band is shifted (to accommodate a Zoom Window that crosses a 'natural' Frequency Slice boundary). If specified, applies for all the receptors in the sub-array. Bands 1, 2, 3 and 4: input from the receptor consists of a single data stream; the Frequency Band Offset (FBO) should be specified for Stream 1 only. Bands 5a and 5b: input from the receptor consists of two data streams; the FBO can be specified for each stream independently. Note: For Band 5a and 5b the frequency shift is performed by the receptor (DISH). Note: This is optional and does not need to be implemented in PI3, but would be great for demo; if Team Buttons is looking for opportunities to showcase interesting GUIs, Zoom Windows are perfect opportunity (would require TMC and CSP to support these two parameters, corrBandwidth values > 0 and zoom window tuning.) | |
| | type | *integer* |
| • frequencyBandOffsetStream2 | See *frequencyBandOffsetStream1* | |
| | type | *integer* |
| • delayModelSubscriptionPoint | FQDN of TMC.DelayModel TANGO attribute which exposes delay values for all the dishes assigned to a Subarray in JSON format. Delay values are updated every 10 seconds. | |
| | type | *string* |
| • dopplerPhaseCorrSubscriptionPoint | The same model applies for all receptors that belong to the subarray. Delivered by TMC using publish-subscribe mechanism (see ICD Section 3.8.8.5.3). The Doppler phase correction, by default, applies only to the CSP_Mid Processing Mode Correlation; optionally may apply to other Processing Modes as well. | |
| | type | *string* |
| • rfiFlaggingMask | Specified as needed in advance of the scan start and/or during the scan. Delivered using publish-subscribe mechanism (see ICD Section 3.8.8.5.7). | |
| | type | *object* |
| | properties | |
| | additionalProperties | False |
| • **fsp** | type | *array* |
| | items | *FSP config 1.0* |
| • vlbi | Very Long Baseline Interferometry specific parameters. To be borrowed from IICD This section contains the parameters relevant only for VLBI. This section is forwarded only to CSP subelement. | |
| | *VLBI config 1.0* | |
| • search_window | type | *array* |
| | items | Up to two 300 MHz Search Windows can be optionally configured and used as input for Transient Data Capture and/or Pulsar Search beamforming. |
| | | *Search window config 1.0* |
| additionalProperties | False | |

### 9.3.3 FSP config 1.0

| | | | | |
|---|---|---|---|---|
| type | *object* | | | |
| properties | | | | |
| • **fspID** | type | *integer* | | |
| • **functionMode** | allOf | type | *string* | |
| | | enum | CORR, PSS-BF, PST-BF, VLBI | |
| • receptors | Optionally a subset of receptors to be correlated can be specified. If not specified, all receptors that belong to the subarray are cross-correlated (i.e. visibilities for all the baselines in the subarray are generated and transmitted to SDP). | | | |
| | type | *array* | | |
| | items | anyOf | type | *integer* |
| | | | type | *string* |
| • **frequencySliceID** | Frequency Slice to be processed on this FSP (valid range depends on the Frequency Band). | | | |
| | type | *integer* | | |
| • **corrBandwidth** | Bandwidth to be correlated calculated as FSBW/2n, where n is in range [0..6].<br>When n=0 the full Frequency Slice bandwidth is correlated.<br>BW > 0 implies 'Zoom Window' configuration; the spectral Zoom Window tuning must be specified. | | | |
| | type | *integer* | | |
| • zoomWindowTuning | The Zoom Window tuning provided in absolute terms as RF center frequency. Based on that, CSP_Mid calculates tuning within the data stream received from the receptor. Must be selected so that the entire Zoom Window is within the Frequency Slice. If partially out of the FS a warning is generated. If completely outside of the FS an exception is generated.<br>Step size <= 0.01MHz.<br>The Frequency Band Offset can be used to shift the entire observed band in order to accommodate a Zoom Window that spans across a Frequency Slice boundary. | | | |
| | type | *integer* | | |
| • **integrationTime** | Integration time for the correlation products, defines multiple of 140 milliseconds. | | | |
| | const | 1400 | | |
| • channelAveragingMap | Table of up to 20 x 2 integers. Each of entries contains: * Start channel ID, and * Averaging factor.<br>Explanation: Each FSP produces 14880 (TBC) fine channels across the correlated bandwidth (Frequency Slice or Zoom Window). Channels are evenly spaced in frequency.<br>TM shall provide the table that for each FSP and each group of 744 channels (there are 20 groups per FSP) indicates the channel averaging factor. More precisely, for each group the TMC provided table specifies: * the channel ID (integer) of the first channel, and * the averaging factor, as follows:<br>0 means do not send channels to SDP,<br>1 means no averaging,<br>2 means average two adjacent channels,<br>3 means average three adjacent channels,<br>and so on.<br>If no entry is present for an FSP, the averaging settings of the previous FSP are still applicable. | | | |
| | type | *array* | | |
| | items | type | *array* | |
| | | items | type | *integer* |
| • fspChannelOffset | Channel ID to use for visibilities of the first channel produced by this FSP. For example, if the fsp channel offset is 5000 the first channel group would span IDs 5000-5743.<br>Note that this offset does not apply to channel maps in this structure (such as *channelAveragingMap* or *outputHost*). | | | |

Continued on next page

Table 4 – continued from previous page

| | | | | | |
|---|---|---|---|---|---|
| | type | *integer* | | | |
| • outputLinkMap | Output links to emit visibilities on for every channel, given as a list of start channel ID to link ID. Where no value is given for concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | anyOf | type | *integer* |
| | | | | type | *string* |
| • outputHost | Output host to send visibilities to for every channel, given as a list of start channel ID to host IP addresses in dot-decimal notation. Where no value is given for a concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | anyOf | type | *integer* |
| | | | | type | *string* |
| • outputPort | Output port to send visibilities to for every channel, given as a list of start channel ID to port number. Where no value is given for a concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | type | *integer* | |
| • outputMac | Output MAC address to send visibilities to for every channel, given as a list of start channel ID to IEEE 802 MAC addresses. Where no value is given for a concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | anyOf | type | *integer* |
| | | | | type | *string* |
| additionalProperties | False | | | | |

### 9.3.4 VLBI config 1.0

Very Long Baseline Interferometry specific parameters. To be borrowed from IICD This section contains the parameters relevant only for VLBI. This section is forwarded only to CSP subelement.

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • dummy_param | type | *string* |
| additionalProperties | False | |

### 9.3.5 Search window config 1.0

Up to two 300 MHz Search Windows can be optionally configured and used as input for Transient Data Capture and/or Pulsar Search beam-forming.

| | | | | |
|---|---|---|---|---|
| type | *object* | | | |
| properties | | | | |
| • **searchWindowID** | Identifier of the 300MHz Search Window. Unique within a sub-array. | | | |
| | type | *integer* | | |
| • **searchWindowTuning** | The Search Window tuning is provided in absolute terms as RF center frequency. The Search Window must be placed within the observed band. If partially out of the observed Band a warning is generated. If completely outside of the observed Band an exception is generated. | | | |
| | type | *integer* | | |
| • **tdcEnable** | Enable / disable Transient Data Capturefor the Search Window. | | | |
| | type | *boolean* | | |
| • tdcNumBits | Number of bits per sample (for the Transient Data Capture). Required if TDC is enabled, otherwise not specified. | | | |
| | type | *integer* | | |
| • tdcPeriodBeforeEpoch | Users can trade the period of time for which data are saved and transmitted for the sample bit-width and/or the number of Search Windows. The exact information regarding the memory capacity per receptor and supported range will be provided in construction.<br>The epoch is specified in the command that triggers TDC off-loading (transmission of data). | | | |
| | type | *integer* | | |
| • tdcPeriodAfterEpoch | see *tdcPeriodBeforeEpoch* | | | |
| | type | *integer* | | |
| • tdcDestinationAddresses | Destination addresses (MAC, IP, port) for off-loading of the content of the Transient Data Capture Buffer, specified per receptor. The destination addresses for the content of the Transient Data Capture can be provided either as a part of the scan configuration or by the command that triggers transmission of the captured data. The latter, if provided, overrides previously set addresses.<br>Required if TDC is enabled, otherwise not specified. | | | |
| | type | *array* | | |
| | items | anyOf | type | *integer* |
| | | | type | *string* |
| additionalProperties | False | | | |

### 9.3.6 PSS configuration 1.0

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • dummy_param | type | *string* |
| additionalProperties | False | |

### 9.3.7 PST configuration 1.0

Pulsar Timing specific parameters. To be borrowed from IICD This section contains the parameters relevant only for PST. This section is forwarded only to PST subelement.

| type | *object* | |
|---|---|---|
| properties | | |
| • dummy_param | type | *string* |
| additionalProperties | False | |

## 9.4 CSP config 0.1

Example (TMC input)

```
{
    "id": "sbi-mvp01-20200325-00001-science_A",
    "frequencyBand": "1",
    "fsp": [{
        "fspID": 1,
        "functionMode": "CORR",
        "frequencySliceID": 1,
        "integrationTime": 1400,
        "corrBandwidth": 0,
        "channelAveragingMap": [
            [0, 2],
            [744, 0]
        ],
        "fspChannelOffset": 0,
        "outputLinkMap": [
            [0, 0],
            [200, 1]
        ]
    }, {
        "fspID": 2,
        "functionMode": "CORR",
        "frequencySliceID": 2,
        "integrationTime": 1400,
        "corrBandwidth": 0,
        "channelAveragingMap": [
            [0, 2],
            [744, 0]
        ],
        "fspChannelOffset": 744,
        "outputLinkMap": [
            [0, 4],
            [200, 5]
        ]
    }]
}
```

Example (CSP configuration for science_a scan)

```
{
    "id": "sbi-mvp01-20200325-00001-science_A",
    "frequencyBand": "1",
    "fsp": [{
        "fspID": 1,
        "functionMode": "CORR",
```

(continues on next page)

(continued from previous page)

```
        "frequencySliceID": 1,
        "integrationTime": 1400,
        "corrBandwidth": 0,
        "channelAveragingMap": [
            [0, 2],
            [744, 0]
        ],
        "fspChannelOffset": 0,
        "outputLinkMap": [
            [0, 0],
            [200, 1]
        ],
        "outputHost": [
            [0, "192.168.0.1"],
            [400, "192.168.0.2"]
        ],
        "outputMac": [
            [0, "06-00-00-00-00-00"]
        ],
        "outputPort": [
            [0, 9000, 1],
            [400, 9000, 1]
        ]
    }, {
        "fspID": 2,
        "functionMode": "CORR",
        "frequencySliceID": 2,
        "integrationTime": 1400,
        "corrBandwidth": 0,
        "channelAveragingMap": [
            [0, 2],
            [744, 0]
        ],
        "fspChannelOffset": 744,
        "outputLinkMap": [
            [0, 4],
            [200, 5]
        ],
        "outputHost": [
            [0, "192.168.0.3"],
            [400, "192.168.0.4"]
        ],
        "outputMac": [
            [0, "06-00-00-00-00-01"]
        ],
        "outputPort": [
            [0, 9000, 1],
            [400, 9000, 1]
        ]
    }]
}
```

Example (CSP configuration for cal_a scan)

```
{
    "id": "sbi-mvp01-20200325-00001-science_A",
    "frequencyBand": "1",
```

(continues on next page)

(continued from previous page)

```
    "fsp": [{
        "fspID": 1,
        "functionMode": "CORR",
        "frequencySliceID": 1,
        "integrationTime": 1400,
        "corrBandwidth": 0,
        "channelAveragingMap": [
            [0, 2],
            [744, 0]
        ],
        "fspChannelOffset": 0,
        "outputLinkMap": [
            [0, 0],
            [200, 1]
        ],
        "outputHost": [
            [0, "192.168.1.1"]
        ],
        "outputPort": [
            [0, 9000, 1]
        ]
    }, {
        "fspID": 2,
        "functionMode": "CORR",
        "frequencySliceID": 2,
        "integrationTime": 1400,
        "corrBandwidth": 0,
        "channelAveragingMap": [
            [0, 2],
            [744, 0]
        ],
        "fspChannelOffset": 744,
        "outputLinkMap": [
            [0, 4],
            [200, 5]
        ],
        "outputHost": [
            [0, "192.168.1.1"]
        ],
        "outputPort": [
            [0, 9744, 1]
        ]
    }]
}
```

| | | | |
|---|---|---|---|
| https://schema.skatelescope.org/ska-csp-configure/0.1 | | | |
| type | *object* | | |
| properties | | | |
| • id | type | *string* | |
| • **frequencyBand** | Frequency band applies for all the receptors (VCCs) that belong to the sub-array. | | |
| | type | *string* | |
| | pattern | ^(1\|2\|3\|4\|5(a\|b))$ | |
| • band5Tuning | Center frequency for the Band-of-Interest. Required if Band is 5a or 5b; not specified for other Bands (not configurable for Band 1, 2, 3 and 4).<br>Input for Band 5a and 5b consists of two 2.5 GHz streams; the center frequency can be independently tuned for each stream.<br>The following nomenclature is used to refer to Band 5a and 5b streams: 5a1, 5a2, 5b1, 5b2. | | |
| | type | *array* | |
| | items | type | *number* |
| • **fsp** | type | *array* | |
| | items | *FSP config 0.1* | |
| additionalProperties | False | | |

### 9.4.1 FSP config 0.1

| | | | | |
|---|---|---|---|---|
| type | *object* | | | |
| properties | | | | |
| • **fspID** | type | *integer* | | |
| • **functionMode** | allOf | type | *string* | |
| | | enum | CORR, PSS-BF, PST-BF, VLBI | |
| • receptors | Optionally a subset of receptors to be correlated can be specified. If not specified, all receptors that belong to the subarray are cross-correlated (i.e. visibilities for all the baselines in the subarray are generated and transmitted to SDP). | | | |
| | type | *array* | | |
| | items | anyOf | type | *integer* |
| | | | type | *string* |
| • **frequencySliceID** | Frequency Slice to be processed on this FSP (valid range depends on the Frequency Band). | | | |
| | type | *integer* | | |
| • **corrBandwidth** | Bandwidth to be correlated calculated as FSBW/2n, where n is in range [0..6].<br>When n=0 the full Frequency Slice bandwidth is correlated.<br>BW > 0 implies 'Zoom Window' configuration; the spectral Zoom Window tuning must be specified. | | | |
| | type | *integer* | | |
| • zoomWindowTuning | The Zoom Window tuning provided in absolute terms as RF center frequency. Based on that, CSP_Mid calculates tuning within the data stream received from the receptor. Must be selected so that the entire Zoom Window is within the Frequency Slice. If partially out of the FS a warning is generated. If completely outside of the FS an exception is generated.<br>Step size <= 0.01MHz.<br>The Frequency Band Offset can be used to shift the entire observed band in order to accommodate a Zoom Window that spans across a Frequency Slice boundary. | | | |

Continued on next page

Table 5 – continued from previous page

| | | | | | |
|---|---|---|---|---|---|
| | type | *integer* | | | |
| • **integrationTime** | Integration time for the correlation products, defines multiple of 140 milliseconds. | | | | |
| | const | 1400 | | | |
| • channelAveragingMap | Table of up to 20 x 2 integers. Each of entries contains: * Start channel ID, and * Averaging factor.<br>Explanation: Each FSP produces 14880 (TBC) fine channels across the correlated bandwidth (Frequency Slice or Zoom Window). Channels are evenly spaced in frequency.<br>TM shall provide the table that for each FSP and each group of 744 channels (there are 20 groups per FSP) indicates the channel averaging factor. More precisely, for each group the TMC provided table specifies: * the channel ID (integer) of the first channel, and * the averaging factor, as follows:<br>0 means do not send channels to SDP,<br>1 means no averaging,<br>2 means average two adjacent channels,<br>3 means average three adjacent channels,<br>and so on.<br>If no entry is present for an FSP, the averaging settings of the previous FSP are still applicable. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | type | *integer* | |
| • fspChannelOffset | Channel ID to use for visibilities of the first channel produced by this FSP. For example, if the channel offset is 5000 the first channel group would span IDs 5000-5743.<br>Note that this offset does not apply to channel maps in this structure (such as *channelAveragingMap* or *outputHost*). | | | | |
| | type | *integer* | | | |
| • outputLinkMap | Output links to emit visibilities on for every channel, given as a list of start channel ID to link ID. Where no value is given for concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | anyOf | type | *integer* |
| | | | | type | *string* |
| • outputHost | Output host to send visibilities to for every channel, given as a list of start channel ID to host IP addresses in dot-decimal notation. Where no value is given for a concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | anyOf | type | *integer* |
| | | | | type | *string* |
| • outputPort | Output port to send visibilities to for every channel, given as a list of start channel ID to port number. Where no value is given for a concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | type | *integer* | |
| • outputMac | Output MAC address to send visibilities to for every channel, given as a list of start channel ID to IEEE 802 MAC addresses. Where no value is given for a concrete channel, the previous value should be used. | | | | |
| | type | *array* | | | |
| | items | type | *array* | | |
| | | items | anyOf | type | *integer* |
| | | | | type | *string* |
| additionalProperties | False | | | | |

# CHAPTER 10

# ska-sdp-assignres

## 10.1 SDP assign resources 0.3

Example

```
{
    "eb_id": "eb-mvp01-20210623-00000",
    "max_length": 100.0,
    "scan_types": [{
        "scan_type_id": "science",
        "reference_frame": "ICRS",
        "ra": "02:42:40.771",
        "dec": "-00:00:47.84",
        "channels": [{
            "count": 744,
            "start": 0,
            "stride": 2,
            "freq_min": 350000000.0,
            "freq_max": 368000000.0,
            "link_map": [
                [0, 0],
                [200, 1],
                [744, 2],
                [944, 3]
            ]
        }, {
            "count": 744,
            "start": 2000,
            "stride": 1,
            "freq_min": 360000000.0,
            "freq_max": 368000000.0,
            "link_map": [
                [2000, 4],
                [2200, 5]
```

(continues on next page)

(continued from previous page)

```
            ]
        }]
    }, {
        "scan_type_id": "calibration",
        "reference_frame": "ICRS",
        "ra": "12:29:06.699",
        "dec": "02:03:08.598",
        "channels": [{
            "count": 744,
            "start": 0,
            "stride": 2,
            "freq_min": 350000000.0,
            "freq_max": 368000000.0,
            "link_map": [
                [0, 0],
                [200, 1],
                [744, 2],
                [944, 3]
            ]
        }, {
            "count": 744,
            "start": 2000,
            "stride": 1,
            "freq_min": 360000000.0,
            "freq_max": 368000000.0,
            "link_map": [
                [2000, 4],
                [2200, 5]
            ]
        }]
    }],
    "processing_blocks": [{
        "pb_id": "pb-mvp01-20210623-00000",
        "workflow": {
            "kind": "realtime",
            "name": "vis_receive",
            "version": "0.1.0"
        },
        "parameters": {}
    }, {
        "pb_id": "pb-mvp01-20210623-00001",
        "workflow": {
            "kind": "realtime",
            "name": "test_realtime",
            "version": "0.1.0"
        },
        "parameters": {}
    }, {
        "pb_id": "pb-mvp01-20210623-00002",
        "workflow": {
            "kind": "batch",
            "name": "ical",
            "version": "0.1.0"
        },
        "parameters": {},
        "dependencies": [{
            "pb_id": "pb-mvp01-20210623-00000",
```

(continues on next page)

(continued from previous page)


```
            "kind": ["visibilities"]
        }]
    }, {
        "pb_id": "pb-mvp01-20210623-00003",
        "workflow": {
            "kind": "batch",
            "name": "dpreb",
            "version": "0.1.0"
        },
        "parameters": {},
        "dependencies": [{
            "pb_id": "pb-mvp01-20210623-00002",
            "kind": ["calibration"]
        }]
    }]
}
```


| https://schema.skao.int/ska-sdp-assignres/0.3 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • transaction_id | type | *string* |
| | pattern | ^txn-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ |
| • **eb_id** | type | *string* |
| | pattern | ^eb-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ |
| • max_length | type | *number* |
| • **scan_types** | Scan types to be supported on subarray | |
| | type | *array* |
| | items | *scan type 0.3* |
| • **processing_blocks** | type | *array* |
| | items | *processing block 0.3* |
| additionalProperties | False | |

### 10.1.1 scan type 0.3

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **scan_type_id** | const | (any scan type) |
| • reference_frame | const | ICRS |
| • ra | type | *string* |
| • dec | type | *string* |
| • channels | type | *array* |
| | items | Channel map |
| | | *scan channels 0.3* |
| additionalProperties | False | |

### 10.1.2 scan channels 0.3

Channel map

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **count** | Number of channels | |
| | type | *integer* |
| • **start** | type | *integer* |
| • **stride** | type | *integer* |
| • **freq_min** | type | *number* |
| • **freq_max** | type | *number* |
| • link_map | type | *array* |
| | items | |
| additionalProperties | False | |

### 10.1.3 processing block 0.3

| | | | | | |
|---|---|---|---|---|---|
| type | *object* | | | | |
| properties | | | | | |
| • **pb_id** | type | *string* | | | |
| | pattern | ^pb-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ | | | |
| • **workflow** | type | *object* | | | |
| | properties | | | | |
| | • **kind** | type | *string* | | |
| | • **name** | type | *string* | | |
| | • **version** | type | *string* | | |
| | additionalProperties | False | | | |
| • parameters | | | | | |
| • dependencies | type | *array* | | | |
| | items | type | *object* | | |
| | | properties | | | |
| | | • **pb_id** | type | *string* | |
| | | | pattern | ^pb-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ | |
| | | • **kind** | type | *array* | |
| | | | items | type | *string* |
| | | additionalProperties | False | | |
| additionalProperties | False | | | | |

## 10.2 SDP assign resources 0.2

Example

```
{
    "id": "sbi-mvp01-20200325-00001",
    "max_length": 100.0,
    "scan_types": [{
        "id": "science",
        "coordinate_system": "ICRS",
        "ra": "02:42:40.771",
        "dec": "-00:00:47.84",
        "channels": [{
            "count": 744,
            "start": 0,
            "stride": 2,
            "freq_min": 350000000.0,
            "freq_max": 368000000.0,
            "link_map": [
                [0, 0],
                [200, 1],
                [744, 2],
                [944, 3]
            ]
        }, {
```

(continues on next page)

(continued from previous page)

```
            "count": 744,
            "start": 2000,
            "stride": 1,
            "freq_min": 360000000.0,
            "freq_max": 368000000.0,
            "link_map": [
                [2000, 4],
                [2200, 5]
            ]
        }]
    }, {
        "id": "calibration",
        "coordinate_system": "ICRS",
        "ra": "12:29:06.699",
        "dec": "02:03:08.598",
        "channels": [{
            "count": 744,
            "start": 0,
            "stride": 2,
            "freq_min": 350000000.0,
            "freq_max": 368000000.0,
            "link_map": [
                [0, 0],
                [200, 1],
                [744, 2],
                [944, 3]
            ]
        }, {
            "count": 744,
            "start": 2000,
            "stride": 1,
            "freq_min": 360000000.0,
            "freq_max": 368000000.0,
            "link_map": [
                [2000, 4],
                [2200, 5]
            ]
        }]
    }],
    "processing_blocks": [{
        "id": "pb-mvp01-20200325-00001",
        "workflow": {
            "type": "realtime",
            "id": "vis_receive",
            "version": "0.1.0"
        },
        "parameters": {}
    }, {
        "id": "pb-mvp01-20200325-00002",
        "workflow": {
            "type": "realtime",
            "id": "test_realtime",
            "version": "0.1.0"
        },
        "parameters": {}
    }, {
        "id": "pb-mvp01-20200325-00003",
```

(continues on next page)

(continued from previous page)

```
        "workflow": {
            "type": "batch",
            "id": "ical",
            "version": "0.1.0"
        },
        "parameters": {},
        "dependencies": [{
            "pb_id": "pb-mvp01-20200325-00001",
            "type": ["visibilities"]
        }]
    }, {
        "id": "pb-mvp01-20200325-00004",
        "workflow": {
            "type": "batch",
            "id": "dpreb",
            "version": "0.1.0"
        },
        "parameters": {},
        "dependencies": [{
            "pb_id": "pb-mvp01-20200325-00003",
            "type": ["calibration"]
        }]
    }]
}
```

| https://schema.skao.int/ska-sdp-assignres/0.2 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **id** | type | *string* |
| | pattern | ^sbi-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ |
| • max_length | type | *number* |
| • **scan_types** | Scan types to be supported on subarray | |
| | type | *array* |
| | items | *scan type 0.2* |
| • **processing_blocks** | type | *array* |
| | items | *processing block 0.2* |
| additionalProperties | False | |

### 10.2.1 scan type 0.2

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **id** | const | (any scan type) |
| • coordinate_system | const | ICRS |
| • ra | type | *string* |
| • dec | type | *string* |
| • channels | type | *array* |
| | items | Channel map |
| | | *scan channels 0.2* |
| additionalProperties | False | |

### 10.2.2 scan channels 0.2

Channel map

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **count** | Number of channels | |
| | type | *integer* |
| • **start** | type | *integer* |
| • **stride** | type | *integer* |
| • **freq_min** | type | *number* |
| • **freq_max** | type | *number* |
| • link_map | type | *array* |
| | items | |
| additionalProperties | False | |

### 10.2.3 processing block 0.2

| | | | | |
|---|---|---|---|---|
| type | *object* | | | |
| properties | | | | |
| • **id** | type | *string* | | |
| | pattern | ^pb-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ | | |
| • **workflow** | type | *object* | | |
| | properties | | | |
| | • **type** | type | *string* | |
| | • **id** | type | *string* | |
| | • **version** | type | *string* | |
| | additionalProperties | False | | |
| • parameters | | | | |
| • dependencies | type | *array* | | |
| | items | type | *object* | |
| | | properties | | |
| | | • **pb_id** | type | *string* |
| | | | pattern | ^pb-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ |
| | | • **type** | type | *array* |
| | | | items | type *string* |
| | | additionalProperties | False | |
| additionalProperties | False | | | |

## 10.3 SDP assign resources 0.1

| | | |
|---|---|---|
| https://schema.skao.int/ska-sdp-assignres/0.1 | | |
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **id** | type | *string* |
| | pattern | ^sbi-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ |
| • max_length | type | *number* |
| • **scan_types** | Scan types to be supported on subarray | |
| | type | *array* |
| | items | *scan type 0.1* |
| • **processing_blocks** | type | *array* |
| | items | *processing block 0.1* |
| additionalProperties | False | |

### 10.3.1 scan type 0.1

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **id** | const | (any scan type) |
| • coordinate_system | const | ICRS |
| • ra | type | *string* |
| • dec | type | *string* |
| • channels | type | *array* |
| | items | Channel map |
| | | *scan channels 0.1* |
| additionalProperties | False | |

### 10.3.2 scan channels 0.1

Channel map

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **count** | Number of channels | |
| | type | *integer* |
| • **start** | type | *integer* |
| • **stride** | type | *integer* |
| • **freq_min** | type | *number* |
| • **freq_max** | type | *number* |
| • link_map | type | *array* |
| | items | |
| additionalProperties | False | |

### 10.3.3 processing block 0.1

| | | | | |
|---|---|---|---|---|
| type | *object* | | | |
| properties | | | | |
| • **id** | type | *string* | | |
| | pattern | ^pb-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ | | |
| • **workflow** | type | *object* | | |
| | properties | | | |
| | • **type** | type | *string* | |
| | • **id** | type | *string* | |
| | • **version** | type | *string* | |
| | additionalProperties | False | | |
| • parameters | | | | |
| • dependencies | type | *array* | | |
| | items | type | *object* | |
| | | properties | | |
| | | • **pb_id** | type | *string* |
| | | | pattern | ^pb-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ |
| | | • **type** | type | *array* |
| | | | items | type: *string* |
| | | additionalProperties | False | |
| additionalProperties | False | | | |

## 10.4 SDP assign resources 0.0

| | | |
|---|---|---|
| https://schema.skao.int/ska-sdp-assignres/0.0 | | |
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **id** | type | *string* |
| | pattern | ^sbi-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ |
| • max_length | type | *number* |
| • **scan_types** | Scan types to be supported on subarray | |
| | type | *array* |
| | items | *scan type 0.0* |
| • **processing_blocks** | type | *array* |
| | items | *processing block 0.0* |
| additionalProperties | False | |

### 10.4.1 scan type 0.0

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **id** | const | (any scan type) |
| • coordinate_system | const | ICRS |
| • ra | type | *string* |
| • dec | type | *string* |
| • channels | type | *array* |
| | items | Channel map |
| | | *scan channels 0.0* |
| additionalProperties | False | |

### 10.4.2 scan channels 0.0

Channel map

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **count** | Number of channels | |
| | type | *integer* |
| • **start** | type | *integer* |
| • **stride** | type | *integer* |
| • **freq_min** | type | *number* |
| • **freq_max** | type | *number* |
| • link_map | type | *array* |
| | items | |
| additionalProperties | False | |

### 10.4.3 processing block 0.0

| | | | | | |
|---|---|---|---|---|---|
| type | *object* | | | | |
| properties | | | | | |
| • **id** | type | *string* | | | |
| | pattern | ^pb-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ | | | |
| • **workflow** | type | *object* | | | |
| | properties | | | | |
| | • **type** | type | *string* | | |
| | • **id** | type | *string* | | |
| | • **version** | type | *string* | | |
| | additionalProperties | False | | | |
| • parameters | | | | | |
| • dependencies | type | *array* | | | |
| | items | type | *object* | | |
| | | properties | | | |
| | | • **pb_id** | type | *string* | |
| | | | pattern | ^pb-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ | |
| | | • **type** | type | *array* | |
| | | | items | type | *string* |
| | | additionalProperties | False | | |
| additionalProperties | False | | | | |

# CHAPTER 11

# ska-sdp-configure

## 11.1 SDP configure 0.3

Example

```
{
    "scan_type": "science"
}
```

Example with new scan types

```
{
    "new_scan_types": [{
        "scan_type_id": "new_calibration",
        "channels": [{
            "count": 372,
            "start": 0,
            "stride": 2,
            "freq_min": 350000000.0,
            "freq_max": 358000000.0,
            "link_map": [
                [0, 0],
                [200, 1]
            ]
        }]
    }],
    "scan_type": "new_calibration"
}
```

| https://schema.skao.int/ska-sdp-configure/0.3 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • transaction_id | type | *string* |
| | pattern | ^txn-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ |
| • **scan_type** | type | *string* |
| • new_scan_types | type | *array* |
| | items | *scan type 0.3* |
| additionalProperties | False | |

### 11.1.1 scan type 0.3

| type | *object* | |
|---|---|---|
| properties | | |
| • **scan_type_id** | const | (any scan type) |
| • reference_frame | const | ICRS |
| • ra | type | *string* |
| • dec | type | *string* |
| • channels | type | *array* |
| | items | Channel map |
| | | *scan channels 0.3* |
| additionalProperties | False | |

### 11.1.2 scan channels 0.3

Channel map

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **count** | Number of channels | |
| | type | *integer* |
| • **start** | type | *integer* |
| • **stride** | type | *integer* |
| • **freq_min** | type | *number* |
| • **freq_max** | type | *number* |
| • link_map | type | *array* |
| | items | |
| additionalProperties | False | |

## 11.2 SDP configure 0.2

Example

```
{
    "scan_type": "science"
}
```

Example with new scan types

```
{
    "new_scan_types": [{
        "id": "new_calibration",
        "channels": [{
            "count": 372,
            "start": 0,
            "stride": 2,
            "freq_min": 350000000.0,
            "freq_max": 358000000.0,
            "link_map": [
                [0, 0],
                [200, 1]
            ]
        }]
    }],
    "scan_type": "new_calibration"
}
```

| https://schema.skao.int/ska-sdp-configure/0.2 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **scan_type** | type | *string* |
| • new_scan_types | type | *array* |
| | items | *scan type 0.2* |
| additionalProperties | False | |

### 11.2.1 scan type 0.2

| type | *object* | |
|---|---|---|
| properties | | |
| • **id** | const | (any scan type) |
| • coordinate_system | const | ICRS |
| • ra | type | *string* |
| • dec | type | *string* |
| • channels | type | *array* |
| | items | Channel map |
| | | *scan channels 0.2* |
| additionalProperties | False | |

### 11.2.2 scan channels 0.2

Channel map

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **count** | Number of channels | |
| | type | *integer* |
| • **start** | type | *integer* |
| • **stride** | type | *integer* |
| • **freq_min** | type | *number* |
| • **freq_max** | type | *number* |
| • link_map | type | *array* |
| | items | |
| additionalProperties | False | |

## 11.3 SDP configure 0.1

| | | |
|---|---|---|
| https://schema.skao.int/ska-sdp-configure/0.1 | | |
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **scan_type** | type | *string* |
| • new_scan_types | type | *array* |
| | items | *scan type 0.1* |
| additionalProperties | False | |

### 11.3.1 scan type 0.1

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **id** | const | (any scan type) |
| • coordinate_system | const | ICRS |
| • ra | type | *string* |
| • dec | type | *string* |
| • channels | type | *array* |
| | items | Channel map |
| | | *scan channels 0.1* |
| additionalProperties | False | |

### 11.3.2 scan channels 0.1

Channel map

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **count** | Number of channels | |
| | type | *integer* |
| • **start** | type | *integer* |
| • **stride** | type | *integer* |
| • **freq_min** | type | *number* |
| • **freq_max** | type | *number* |
| • link_map | type | *array* |
| | items | |
| additionalProperties | False | |

## 11.4 SDP configure 0.0

| | | |
|---|---|---|
| https://schema.skao.int/ska-sdp-configure/0.0 | | |
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **scan_type** | type | *string* |
| • new_scan_types | type | *array* |
| | items | *scan type 0.0* |
| additionalProperties | False | |

### 11.4.1 scan type 0.0

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **id** | const | (any scan type) |
| • coordinate_system | const | ICRS |
| • ra | type | *string* |
| • dec | type | *string* |
| • channels | type | *array* |
| | items | Channel map |
| | | *scan channels 0.0* |
| additionalProperties | False | |

### 11.4.2 scan channels 0.0

Channel map

| | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **count** | Number of channels | |
| | type | *integer* |
| • **start** | type | *integer* |
| • **stride** | type | *integer* |
| • **freq_min** | type | *number* |
| • **freq_max** | type | *number* |
| • link_map | type | *array* |
| | items | |
| additionalProperties | False | |

# CHAPTER 12

# ska-sdp-scan

## 12.1 SDP scan 0.3

Example

```
{
    "scan_id": 1
}
```

| https://schema.skao.int/ska-sdp-scan/0.3 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • transaction_id | type | *string* |
| | pattern | ^txn-[a-z0-9]+-[0-9]{8}-[a-z0-9]+$ |
| • **scan_id** | type | *integer* |
| additionalProperties | False | |

## 12.2 SDP scan 0.2

Example

```
{
    "id": 1
}
```

| https://schema.skao.int/ska-sdp-scan/0.2 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **id** | type | *integer* |
| additionalProperties | False | |

## 12.3 SDP scan 0.1

| https://schema.skao.int/ska-sdp-scan/0.1 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **id** | type | *integer* |
| additionalProperties | False | |

## 12.4 SDP scan 0.0

| https://schema.skao.int/ska-sdp-scan/0.0 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • interface | type | *string* |
| • **id** | type | *integer* |
| additionalProperties | False | |

# CHAPTER 13

# ska-sdp-recvaddrs

## 13.1 SDP receive addresses map 0.3

| | | | |
|---|---|---|---|
| https://schema.skao.int/ska-sdp-recvaddrs/0.3 | | | |
| type | *object* | | |
| properties | | | |
| • interface | type | *string* | |
| • **(any scan type)** | type | *object* | |
| | properties | | |
| | • **host** | type | *array* |
| | | items | |
| | • mac | type | *array* |
| | | items | |
| | • port | type | *array* |
| | | items | |
| | additionalProperties | False | |
| additionalProperties | False | | |

## 13.2 SDP receive addresses map 0.2

| | | | |
|---|---|---|---|
| https://schema.skao.int/ska-sdp-recvaddrs/0.2 | | | |
| type | *object* | | |
| properties | | | |
| • interface | type | *string* | |
| • **(any scan type)** | type | *object* | |
| | properties | | |
| | • **host** | type | *array* |
| | | items | |
| | • mac | type | *array* |
| | | items | |
| | • port | type | *array* |
| | | items | |
| | additionalProperties | False | |
| additionalProperties | False | | |

## 13.3 SDP receive addresses 0.1

| | | | | | | |
|---|---|---|---|---|---|---|
| https://schema.skao.int/ska-sdp-recvaddrs/0.1 | | | | | | |
| type | *object* | | | | | |
| properties | | | | | | |
| • interface | type | *string* | | | | |
| • **scanId** | type | *integer* | | | | |
| • **totalChannels** | type | *integer* | | | | |
| • **receiveAddresses** | type | *array* | | | | |
| | items | type | *object* | | | |
| | | properties | | | | |
| | | • **phaseBinId** | type | *integer* | | |
| | | • **fspId** | type | *integer* | | |
| | | • **hosts** | type | *array* | | |
| | | | items | type | *object* | |
| | | | | properties | | |
| | | | | • **host** | type | *string* |
| | | | | • **channels** | type | *array* |
| | | | | | items | type |
| | | | | | | properties |
| | | | | | | • **portOffset** type *integer* |
| | | | | | | • **startChannel** type *integer* |
| | | | | | | • **numChannels** type *integer* |
| | | | | | | additionalProperties false |
| | | | | additionalProperties false | | |
| | | additionalProperties false | | | | |
| additionalProperties false | | | | | | |

## 13.4 SDP receive addresses 0.0

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| https://schema.skao.int/ska-sdp-recvaddrs/0.0 | | | | | | | |
| type | *object* | | | | | | |
| properties | | | | | | | |
| • **interface** | type | *string* | | | | | |
| • **scanId** | type | *integer* | | | | | |
| • **totalChannels** | type | *integer* | | | | | |
| • **receiveAddresses** | type | *array* | | | | | |
| | items | type | *object* | | | | |
| | | properties | | | | | |
| | | • **phaseBinId** | type | *integer* | | | |
| | | • **fspId** | type | *integer* | | | |
| | | • **hosts** | type | *array* | | | |
| | | | items | type | *object* | | |
| | | | | properties | | | |
| | | | | • **host** | type | *string* | |
| | | | | • **channels** | type | *array* | |
| | | | | | items | type | *object* |
| | | | | | | properties | |
| | | | | | | • **portOffset** | type | *integer* |
| | | | | | | • **startChannel** | type | *integer* |
| | | | | | | • **numChannels** | type | *integer* |
| | | | | | | additionalProperties | false |
| | | | | additionalProperties | false | | |
| | | additionalProperties | false | | | | |
| additionalProperties | false | | | | | | |

# CHAPTER 14

# ska-low-mccs-assignedresources

## 14.1 Low MCCS assigned resources 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-mccs-assignedresources/1.0",
    "subarray_beam_ids": [1],
    "station_ids": [
        [1, 2]
    ],
    "channel_blocks": [3]
}
```

| | | | | |
|---|---|---|---|---|
| https://schema.skatelescope.org/ska-low-mccs-assignedresources/1.0 | | | | |
| type | *object* | | | |
| properties | | | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | | | |
| | type | *string* | | |
| • **subarray_beam_ids** | IDs of the MCCS sub-array beams allocated to this MCCS subarray.<br>Each ID must be between 1 and 48, the maximum number of MCCS sub-array beams.<br>As of PI10, only one MCCS sub-array beam can be configured per allocation request. Multiple beams must be allocated via multiple allocation requests. | | | |
| | type | *array* | | |
| | items | type | *integer* | |
| • **station_ids** | IDs of MCCS stations allocated to each sub-array beam.<br>Each ID must be between 1 and 512, the maximum number of stations. | | | |
| | type | *array* | | |
| | items | type | *array* | |
| | | items | type | *integer* |
| • **channel_blocks** | Number of channel blocks allocated to each sub-array beam.<br>Maximum number of channel blocks = 48. | | | |
| | type | *array* | | |
| | items | type | *integer* | |
| additionalProperties | False | | | |

# CHAPTER 15

# ska-low-mccs-assignresources

## 15.1 Low MCCS assign resources 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-mccs-assignresources/1.0",
    "subarray_id": 1,
    "subarray_beam_ids": [1],
    "station_ids": [
        [1, 2]
    ],
    "channel_blocks": [3]
}
```

| | | | | |
|---|---|---|---|---|
| https://schema.skatelescope.org/ska-low-mccs-assignresources/1.0 | | | | |
| type | *object* | | | |
| properties | | | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | | | |
| | type | *string* | | |
| • **subarray_id** | ID of sub-array targeted by this resource allocation request | | | |
| | type | *integer* | | |
| • **subarray_beam_ids** | IDs of the MCCS sub-array beams to allocate to this MCCS subarray.<br>Each ID must be between 1 and 48, the maximum number of sub-array beams.<br>As of PI10, only one MCCS sub-array beam can be configured per allocation request. Multiple beams must be allocated via multiple allocation requests. | | | |
| | type | *array* | | |
| | items | type | *integer* | |
| • **station_ids** | IDs of MCCS stations to allocate to this sub-array beam.<br>Each ID must be between 1 and 512, the maximum number of stations. | | | |
| | type | *array* | | |
| | items | type | *array* | |
| | | items | type | *integer* |
| • **channel_blocks** | Number of channel blocks to allocate to this sub-array beam.<br>Maximum number of channel blocks = 48. | | | |
| | type | *array* | | |
| | items | type | *integer* | |
| additionalProperties | False | | | |

# CHAPTER 16

# ska-low-mccs-releaseresources

## 16.1 Low MCCS resource release 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-mccs-releaseresources/1.0",
    "subarray_id": 1,
    "release_all": true
}
```

| https://schema.skatelescope.org/ska-low-mccs-releaseresources/1.0 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | |
| | type | *string* |
| • **subarray_id** | ID of the MCCS sub-array which should release resources. | |
| | type | *integer* |
| • **release_all** | true to release all resources, false to release only the resources defined in this payload.<br>Note: partial resource release for MCCS is not implemented and the identification of the resources to release is not yet part of the schema. | |
| | type | *boolean* |
| additionalProperties | False | |

# CHAPTER 17

# ska-low-mccs-configure

## 17.1 Low MCCS configure 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-mccs-configure/1.0",
    "stations": [{
        "station_id": 1
    }, {
        "station_id": 2
    }],
    "subarray_beams": [{
        "subarray_beam_id": 1,
        "station_ids": [1, 2],
        "update_rate": 0.0,
        "channels": [
            [0, 8, 1, 1],
            [8, 8, 2, 1],
            [24, 16, 2, 1]
        ],
        "sky_coordinates": [0.0, 180.0, 0.0, 45.0, 0.0],
        "antenna_weights": [1.0, 1.0, 1.0],
        "phase_centre": [0.0, 0.0]
    }]
}
```

| https://schema.skatelescope.org/ska-low-mccs-configure/1.0 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | |
| | type | *string* |

Continued on next page

Table 1 – continued from previous page

| | | | | | |
|---|---|---|---|---|---|
| • **stations** | IDs of the MCCS stations to configure.<br>Maximum array size = 512, the maximum number of MCCS stations. | | | | |
| | type | *array* | | | |
| | items | type | *object* | | |
| | | properties | | | |
| | | • **station_id** | type | *integer* | |
| | | | additionalProperties | False | |
| • **subarray_beams** | MCCS sub-array beam configuration. | | | | |
| | type | *array* | | | |
| | items | type | *object* | | |
| | | properties | | | |
| | | • **subarray_beam_id** | ID of MCCS sub-array beam to configure.<br>Must be an integer between 1 and 48. | | |
| | | | type | *integer* | |
| | | • **station_ids** | IDs of MCCS stations within this sub-array beamto configure.<br>Array size must be less than 512, the maximum number of MCCS stations.<br>Each item in the list must be an integer between 1 and 512. | | |
| | | | type | *array* | |
| | | | items | type | *integer* |
| | | • **update_rate** | Update rate for pointing information.<br>Value must be 0.0 or greater.<br>TODO: clarify whether this is specified as a frequency or as a cadence, plus units. | | |
| | | | type | *number* | |
| | | • **channels** | Channel block configurations.<br>Each item in the list is a channel block configuration, each specified as a list of 4 numbers as follows:<br>[start channel, number of channels, beam index, sub-station index]<br>Constraints are:<br>0 < start channel < 376<br>start channel must be a multiple of 8<br>8 < number of channels < 48<br>1 < beam index < 48<br>1 < sub-station index < 8 | | |
| | | | type | *array* | |
| | | | items | type | *array* |
| | | | | items | type | *integer* |
| | | • **antenna_weights** | Antenna weights.<br>Maximum array size = 512 (=256 antennas x2 pols per sub-array beam).<br>Antennas signals can be weighted to modify the station beam, varying from 0.0 for full exclusion to potentially 256.0 for an antenna contribution compensated for the number of antennas in the beam. This value is an amplitude multiplier added to that antenna signal before adding into the sum.<br>Weights apply to all channels assigned to a beam. | | |
| | | | type | *array* | |

Continued on next page

Table 1 – continued from previous page

| | | | | | |
|---|---|---|---|---|---|
| | | | items | type | *number* |
| | | • **phase_centre** | Phase centre offset for the station beam, in metres. The reference position for station phase must be modified to reflect antenna weighting and their contribution to the station beam. This offset can be can considered the desired centre of mass for the station. Constraints: array size = 2 -20 < phase centre value < 20 | | |
| | | | type | *array* | |
| | | | items | type | *number* |
| | | • **sky_coordinates** | Azimuth/elevation of sub-array beam target, in degrees. | | |
| | | | type | *array* | |
| | | | items | type | *number* |
| | | additionalProperties | False | | |
| additionalProperties | False | | | | |

# CHAPTER 18

# ska-low-mccs-scan

## 18.1 Low MCCS scan 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-mccs-scan/1.0",
    "scan_id": 1,
    "start_time": 0.0
}
```

| https://schema.skatelescope.org/ska-low-mccs-scan/1.0 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | |
| | type | *string* |
| • **scan_id** | Scan ID to associate with the data.<br>The scan ID and SBI ID are used together to uniquely associate the data taken with the telescope configuration in effect at the moment of observation. | |
| | type | *integer* |
| • **start_time** | Start time for the scan.<br>Currently unused and can be set to 0.0. | |
| | type | *number* |
| additionalProperties | False | |

# CHAPTER 19

# ska-low-tmc-assignresources

## 19.1 Low TMC assign resources 2.0

Example JSON.

```
{
    "interface": "https://schema.skao.in/ska-low-tmc-assignresources/2.0",
    "transaction_id": "txn-....-00001",
    "subarray_id": 1,
    "mccs": {
        "subarray_beam_ids": [1],
        "station_ids": [
            [1, 2]
        ],
        "channel_blocks": [3]
    }
}
```

| | | | | |
|---|---|---|---|---|
| https://schema.skao.int/ska-low-tmc-assignresources/2.0 | | | | |
| type | *object* | | | |
| properties | | | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | | | |
| | type | *string* | | |
| • transaction_id | A transaction id specific to the command | | | |
| | type | *string* | | |
| • **subarray_id** | ID of sub-array targeted by this resource allocation request | | | |
| | type | *integer* | | |
| • **mccs** | MCCS specification for resource allocation. | | | |
| | type | *object* | | |
| | properties | | | |
| | • **subarray_beam_ids** | IDs of the MCCS sub-array beams to allocate to this subarray.<br>Each ID must be between 1 and 48, the maximum number of sub-array beams.<br>As of PI10, only one MCCS sub-array beam can be configured per allocation request. Multiple beams must be allocated via multiple allocation requests. | | |
| | | type | *array* | |
| | | items | type | *integer* |
| | • **station_ids** | IDs of MCCS stations to allocate to this sub-array beam.<br>Each ID must be between 1 and 512, the maximum number of stations. | | |
| | | type | *array* | |
| | | items | type | *array* |
| | | | items | type | *integer* |
| | • **channel_blocks** | Number of channel blocks to allocate to this sub-array beam.<br>Maximum number of channel blocks = 48. | | |
| | | type | *array* | |
| | | items | type | *integer* |
| | additionalProperties | False | | |
| additionalProperties | False | | | |

## 19.2 Low TMC assign resources 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-tmc-assignresources/1.0",
    "subarray_id": 1,
    "mccs": {
        "subarray_beam_ids": [1],
        "station_ids": [
            [1, 2]
        ],
        "channel_blocks": [3]
    }
}
```

| | | | | |
|---|---|---|---|---|
| https://schema.skatelescope.org/ska-low-tmc-assignresources/1.0 | | | | |
| type | *object* | | | |
| properties | | | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | | | |
| | type | *string* | | |
| • transaction_id | A transaction id specific to the command | | | |
| | type | *string* | | |
| • **subarray_id** | ID of sub-array targeted by this resource allocation request | | | |
| | type | *integer* | | |
| • **mccs** | MCCS specification for resource allocation. | | | |
| | type | *object* | | |
| | properties | | | |
| | • **subarray_beam_ids** | IDs of the MCCS sub-array beams to allocate to this subarray.<br>Each ID must be between 1 and 48, the maximum number of sub-array beams.<br>As of PI10, only one MCCS sub-array beam can be configured per allocation request. Multiple beams must be allocated via multiple allocation requests. | | |
| | | type | *array* | |
| | | items | type | *integer* |
| | • **station_ids** | IDs of MCCS stations to allocate to this sub-array beam.<br>Each ID must be between 1 and 512, the maximum number of stations. | | |
| | | type | *array* | |
| | | items | type | *array* |
| | | | items | type | *integer* |
| | • **channel_blocks** | Number of channel blocks to allocate to this sub-array beam.<br>Maximum number of channel blocks = 48. | | |
| | | type | *array* | |
| | | items | type | *integer* |
| | additionalProperties | False | | |
| additionalProperties | False | | | |

# CHAPTER 20

# ska-low-tmc-configure

## 20.1 Low TMC configure 2.0

Example JSON.

```
{
    "interface": "https://schema.skao.in/ska-low-tmc-configure/2.0",
    "transaction_id": "txn-....-00001",
    "mccs": {
        "stations": [{
            "station_id": 1
        }, {
            "station_id": 2
        }],
        "subarray_beams": [{
            "subarray_beam_id": 1,
            "station_ids": [1, 2],
            "update_rate": 0.0,
            "channels": [
                [0, 8, 1, 1],
                [8, 8, 2, 1],
                [24, 16, 2, 1]
            ],
            "antenna_weights": [1.0, 1.0, 1.0],
            "phase_centre": [0.0, 0.0],
            "target": {
                "reference_frame": "HORIZON",
                "target_name": "DriftScan",
                "az": 180.0,
                "el": 45.0
            }
        }]
    },
    "tmc": {
```

(continues on next page)

(continued from previous page)

```
        "scan_duration": 10.0
    }
}
```

| | | | | | |
|---|---|---|---|---|---|
| https://schema.skao.int/ska-low-tmc-configure/2.0 | | | | | |
| type | *object* | | | | |
| properties | | | | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | | | | |
| | type | *string* | | | |
| • transaction_id | A transaction id specific to the command | | | | |
| | type | *string* | | | |
| • **mccs** | MCCS configuration specification. | | | | |
| | type | *object* | | | |
| | properties | | | | |
| | • **stations** | IDs of the MCCS stations to configure.<br>Maximum array size = 512, the maximum number of MCCS stations. | | | |
| | | type | *array* | | |
| | | items | type | *object* | |
| | | | properties | | |
| | | | • **station_id** | MCCS Station ID.<br>Each ID must be between 1 and 512. | |
| | | | | type | *integer* |
| | | | additionalProperties | False | |
| | • **subarray_beams** | MCCS sub-array beam configuration. | | | |
| | | type | *array* | | |
| | | items | type | *object* | |
| | | | properties | | |
| | | | • **subarray_beam_id** | ID of MCCS sub-array beam to configure.<br>ID must be an integer between 1 and 48. | |
| | | | | type | *integer* |
| | | | • **station_ids** | IDs of MCCS stations within this sub-array beamto configure.<br>Array size must be less than 512, the maximum number of MCCS stations.<br>Each item in the list must be an integer between 1 and 512. | |
| | | | | type | *array* |
| | | | | items | type | *integer* |
| | | | • **update_rate** | Update rate for pointing information.<br>Value must be 0.0 or greater.<br>TODO: clarify whether this is specified as a frequency or as a cadence, plus units. | |
| | | | | type | *number* |

Continued on next page

Table 1 – continued from previous page

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | • **channels** | Channel block configurations.<br>Each item in the list is a channel block configuration, each specified as a list of 4 numbers as follows:<br>[start channel, number of channels, beam index, sub-station index]<br>Constraints are:<br>0 < start channel < 376<br>start channel must be a multiple of 8<br>8 < number of channels < 48<br>1 < beam index < 48<br>1 < sub-station index < 8 | | |
| | | | | type | *array* | | |
| | | | | items | type | *array* | |
| | | | | | items | type | *integer* |
| | | | • **antenna_weights** | Antenna weights.<br>Maximum array size = 512 (=256 antennas x2 pols per sub-array beam).<br>Antennas signals can be weighted to modify the station beam, varying from 0.0 for full exclusion to potentially 256.0 for an antenna contribution compensated for the number of antennas in the beam. This value is an amplitude multiplier added to that antenna signal before adding into the sum.<br>Weights apply to all channels assigned to a beam. | | |
| | | | | type | *array* | | |
| | | | | items | type | *number* | |
| | | | • **phase_centre** | Phase centre offset for the station beam, in metres.<br>The reference position for station phase must be modified to reflect antenna weighting and their contribution to the station beam. This offset can be can considered the desired centre of mass for the station.<br>Constraints: array size = 2 -20 < phase centre value < 20 | | |
| | | | | type | *array* | | |
| | | | | items | type | *number* | |
| | | | • **target** | Target position for the sub-array beam.<br>Only drift scan targets are currently implemented by MCCS, hence only azimuth and elevation are specified. | | |
| | | | | type | *object* | | |
| | | | | properties | | | |
| | | | | • **reference_frame** | Co-ordinate system.<br>Must be HORIZON for drift scan. | | |
| | | | | | type | *string* | |
| | | | | • **target_name** | Name of target. | | |
| | | | | | type | *string* | |
| | | | | • **az** | Pointing azimuth in degrees. | | |
| | | | | | type | *number* | |
| | | | | • **el** | Pointing elevation in degrees. | | |
| | | | | | type | *number* | |
| | | | | additionalProperties | False | | |
| | | | additionalProperties | False | | | |
| | additionalProperties | False | | | | | |

Continued on next page

Table 1 – continued from previous page

| | | |
|---|---|---|
| • tmc | TMC configuration specification. | |
| | type | *object* |
| | properties | |
| | • **scan_duration** | Scan duration in seconds.<br>Value must be >= 0.0 |
| | | type *number* |
| | additionalProperties | False |
| additionalProperties | False | |

## 20.2 Low TMC configure 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-tmc-configure/1.0",
    "mccs": {
        "stations": [{
            "station_id": 1
        }, {
            "station_id": 2
        }],
        "subarray_beams": [{
            "subarray_beam_id": 1,
            "station_ids": [1, 2],
            "update_rate": 0.0,
            "channels": [
                [0, 8, 1, 1],
                [8, 8, 2, 1],
                [24, 16, 2, 1]
            ],
            "antenna_weights": [1.0, 1.0, 1.0],
            "phase_centre": [0.0, 0.0],
            "target": {
                "system": "HORIZON",
                "name": "DriftScan",
                "az": 180.0,
                "el": 45.0
            }
        }]
    },
    "tmc": {
        "scan_duration": 10.0
    }
}
```

| | | |
|---|---|---|
| https://schema.skatelescope.org/ska-low-tmc-configure/1.0 | | |
| type | *object* | |
| properties | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | |
| | type | *string* |
| • transaction_id | A transaction id specific to the command | |
| | type | *string* |

Continued on next page

Table 2 – continued from previous page

| | | | | | |
|---|---|---|---|---|---|
| • **mccs** | MCCS configuration specification. | | | | |
| | type | *object* | | | |
| | properties | | | | |
| | • **stations** | IDs of the MCCS stations to configure.<br>Maximum array size = 512, the maximum number of MCCS stations. | | | |
| | | type | *array* | | |
| | | items | type | *object* | |
| | | | properties | | |
| | | | • **station_id** | MCCS Station ID.<br>Each ID must be between 1 and 512. | |
| | | | | type | *integer* |
| | | | additionalProperties | False | |
| | • **subarray_beams** | MCCS sub-array beam configuration. | | | |
| | | type | *array* | | |
| | | items | type | *object* | |
| | | | properties | | |
| | | | • **subarray_beam_id** | ID of MCCS sub-array beam to configure.<br>ID must be an integer between 1 and 48. | |
| | | | | type | *integer* |
| | | | • **station_ids** | IDs of MCCS stations within this sub-array beam to configure.<br>Array size must be less than 512, the maximum number of MCCS stations.<br>Each item in the list must be an integer between 1 and 512. | |
| | | | | type | *array* |
| | | | | items | type | *integer* |
| | | | • **update_rate** | Update rate for pointing information.<br>Value must be 0.0 or greater.<br>TODO: clarify whether this is specified as a frequency or as a cadence, plus units. | |
| | | | | type | *number* |
| | | | • **channels** | Channel block configurations.<br>Each item in the list is a channel block configuration, each specified as a list of 4 numbers as follows:<br>[start channel, number of channels, beam index, sub-station index]<br>Constraints are:<br>0 < start channel < 376<br>start channel must be a multiple of 8<br>8 < number of channels < 48<br>1 < beam index < 48<br>1 < sub-station index < 8 | |
| | | | | type | *array* | |
| | | | | items | type | *array* |
| | | | | | items | type | *integer* |

Continued on next page

Table 2 – continued from previous page

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | • **antennaWeights** | Antenna weights.<br>Maximum array size = 512 (=256 antennas x2 pols per sub-array beam).<br>Antennas signals can be weighted to modify the station beam, varying from 0.0 for full exclusion to potentially 256.0 for an antenna contribution compensated for the number of antennas in the beam. This value is an amplitude multiplier added to that antenna signal before adding into the sum.<br>Weights apply to all channels assigned to a beam. | | |
| | | | | type | *array* | |
| | | | | items | type | *number* |
| | | | • **phase_centre** | Phase centre offset for the station beam, in metres.<br>The reference position for station phase must be modified to reflect antenna weighting and their contribution to the station beam. This offset can be can considered the desired centre of mass for the station.<br>Constraints: array size = 2 -20 < phase centre value < 20 | | |
| | | | | type | *array* | |
| | | | | items | type | *number* |
| | | | • **target** | Target position for the sub-array beam.<br>Only drift scan targets are currently implemented by MCCS, hence only azimuth and elevation are specified. | | |
| | | | | type | *object* | |
| | | | | properties | | |
| | | | | • **system** | Co-ordinate system.<br>Must be HORIZON for drift scan. | |
| | | | | | type | *string* |
| | | | | • **name** | Name of target. | |
| | | | | | type | *string* |
| | | | | • **az** | Pointing azimuth in degrees. | |
| | | | | | type | *number* |
| | | | | • **el** | Pointing elevation in degrees. | |
| | | | | | type | *number* |
| | | | | additionalProperties | False | |
| | | | additionalProperties | False | | |
| | | additionalProperties | False | | | |
| | • tmc | TMC configuration specification. | | | | |
| | | type | *object* | | | |
| | | properties | | | | |
| | | • **scan_duration** | Scan duration in seconds.<br>Value must be >= 0.0 | | | |
| | | | type | *number* | | |
| | | additionalProperties | False | | | |
| additionalProperties | | False | | | | |

# CHAPTER 21

# ska-low-tmc-releaseresources

## 21.1 Low TMC resource release 2.0

Example JSON.

```
{
    "interface": "https://schema.skao.in/ska-low-tmc-releaseresources/2.0",
    "transaction_id": "txn-....-00001",
    "subarray_id": 1,
    "release_all": true
}
```

| https://schema.skao.int/ska-low-tmc-releaseresources/2.0 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | |
| | type | *string* |
| • transaction_id | A transaction id specific to the command | |
| | type | *string* |
| • **subarray_id** | ID of the sub-array which should release resources. | |
| | type | *integer* |
| • **release_all** | true to release all resources, false to release only the resources defined in this payload.<br>Note: partial resource release for SKA LOW is not implemented and the identification of the resources to release is not yet part of the schema. | |
| | type | *boolean* |
| additionalProperties | False | |

## 21.2 Low TMC resource release 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-tmc-releaseresources/1.0",
    "subarray_id": 1,
    "release_all": true
}
```

| https://schema.skatelescope.org/ska-low-tmc-releaseresources/1.0 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | |
| | type | *string* |
| • transaction_id | A transaction id specific to the command | |
| | type | *string* |
| • **subarray_id** | ID of the sub-array which should release resources. | |
| | type | *integer* |
| • **release_all** | true to release all resources, false to release only the resources defined in this payload.<br>Note: partial resource release for SKA LOW is not implemented and the identification of the resources to release is not yet part of the schema. | |
| | type | *boolean* |
| additionalProperties | False | |

# CHAPTER 22

# ska-low-tmc-scan

## 22.1 Low TMC scan 2.0

Example JSON.

```
{
    "interface": "https://schema.skao.in/ska-low-tmc-scan/2.0",
    "transaction_id": "txn-....-00001",
    "scan_id": 1
}
```

| https://schema.skao.int/ska-low-tmc-scan/2.0 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | |
| | type | *string* |
| • transaction_id | A transaction id specific to the command | |
| | type | *string* |
| • **scan_id** | Scan ID to associate with the data.<br>The scan ID and SBI ID are used together to uniquely associate the data taken with the telescope configuration in effect at the moment of observation. | |
| | type | *integer* |
| additionalProperties | False | |

## 22.2 Low TMC scan 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-tmc-scan/1.0",
    "scan_id": 1
}
```


| https://schema.skatelescope.org/ska-low-tmc-scan/1.0 | | |
|---|---|---|
| type | *object* | |
| properties | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | |
| | type | *string* |
| • transaction_id | A transaction id specific to the command | |
| | type | *string* |
| • **scan_id** | Scan ID to associate with the data.<br>The scan ID and SBI ID are used together to uniquely associate the data taken with the telescope configuration in effect at the moment of observation. | |
| | type | *integer* |
| additionalProperties | False | |

# CHAPTER 23

# ska-low-tmc-assignedresources

## 23.1 Low TMC assigned resources 1.0

Example JSON.

```
{
    "interface": "https://schema.skatelescope.org/ska-low-tmc-assignedresources/1.0",
    "mccs": {
        "subarray_beam_ids": [1],
        "station_ids": [
            [1, 2]
        ],
        "channel_blocks": [3]
    }
}
```

| | | | | |
|---|---|---|---|---|
| https://schema.skatelescope.org/ska-low-tmc-assignedresources/1.0 | | | | |
| type | *object* | | | |
| properties | | | | |
| • **interface** | URI of JSON schema applicable to this JSON payload. | | | |
| | type | *string* | | |
| • **mccs** | Specification of the MCCS resources allocated to this sub-array. | | | |
| | type | *object* | | |
| | properties | | | |
| | • **subarray_beam_ids** | IDs of the MCCS sub-array beams allocated to this subarray. Each ID must be between 1 and 48, the maximum number of sub-array beams. | | |
| | | type | *array* | |
| | | items | type | *integer* |
| | • **station_ids** | IDs of MCCS stations allocated to each MCCS sub-array beam. Each ID must be between 1 and 512, the maximum number of stations. | | |
| | | type | *array* | |
| | | items | type | *array* |
| | | | items | type | *integer* |
| | • **channel_blocks** | Number of channel blocks allocated per sub-array beam. Maximum number of channel blocks = 48. | | |
| | | type | *array* | |
| | | items | type | *integer* |
| | additionalProperties | False | | |
| additionalProperties | False | | | |

CHAPTER 24

# Project-name documentation HEADING

These are all the packages, functions and scripts that form part of the project.

- *Telescope Model Public API*

# Python Module Index

## S

# Index