SKA1 TM TMC-OSO INTERFACE CONTROL DOCUMENT

TM Number T0000-0000-ICD-012
SKAO Number SKA-TEL-TM-0000297
Context ..... TM.SE
Revision 01
Primary Author ..... Subhrojyoti Roy Chaudhuri
Date 2018-06-29
Document Classification ..... FOR PROJECT USE ONLY
Status ..... Approved

Name	Designation	Affiliation	Signature	
Custodian:				
Subhrojyoti Roy Chaudhuri	TELMGT Lead	TRDDC	Subhrojyoti Roy Chaudhuri Subhrojyoti Roy Chaudhuri (Jul 11, 2018)	
			Date:	Jul 11, 2018
Following Party:				
Alan Bridger	OBSMGT Lead	UKATC/STFC	Alan Bridger Alan Bridger (Jul 10, 2018)	
			Date:	Jul 10, 2018
Approved by:				
Vinod Sathe	Project Manager	TCS	VMS Vinod Sathe (Jul 10, 2018)	
			Date:	Jul 10, 2018
Released by:				
LRS Brederode	TM Configuration Manager	SKA SA	Ray Brederode Ray Brederode (Jul 10, 2018)	
			Date:	Jul 10, 2018

Name	Designation	Affiliation
Author(s)		
Lochan Babani	TELMGT Software Design Engineer	TCS
Apurva Patkar	TELMGT Software Design Engineer	TCS
Vatsal Trivedi	TELMGT Software Design Engineer	TCS
Stewart Williams	OBSMGT Software Design Engineer	UKATC/STFC

DOCUMENT HISTORY

Revision	Date of Issue	Engineering Change Number	Comments
A	2018-05-30	NA	Initial draft of the ICD created in response to Jira tickets TMCDR-215, TMCDR-236, TMCDR-604 and TMCDR-745 (Regarding the maturity of the TMC-OSO interface in section #14 in TMC SAD Rev 01) and circulated for TM and SKAO review.
01	2018-06-29	NA	Updated as per comments received after TM and SKAO review. Approved for CDR closure.

DOCUMENT SOFTWARE

	Package	Version	File Name
Word processor	Google Docs	-	TMC-OSO Interface
	Microsoft Word	Word 2013	T0000-0000-ICD-012-01_TMC_OSO_ICD.docx
Block Diagrams	Microsoft PowerPoint	PowerPoint 2013	T0000-0000-ICD-012-01_TMC_OSO_ICD.pptx
Other			

ORGANISATION DETAILS

Name	National Centre for Radio Astrophysics
Registered Address	National Centre for Radio Astrophysics Tata Institute of Fundamental Research, Pune University Campus, Post Bag 3 , Ganeshkhind, Pune – 411007, Maharashtra, India
Phone	Tel: +91 20 25719000, +91 20 25719111 Fax: +91 20 25692149
Website	www.ncra.tifr.res.in

Table of Contents

List of Figures

List of Tables

1 Introduction

This document defines the interface between the TMC Components (such as the Central Node, a Subarray Node, and the Resource Manager) and OSO components (such as the Observation Scheduling Tool and Observation Execution Tool) for TM Mid.

Scope of this document is to identify the content exchanged between the TMC components and OSO components and to describe method for data exchange, valid command sequences, and constraints imposed by TMC and OSO architecture (if any). This document describes TANGO API (TANGO commands, TANGO attributes, TANGO Pipes) implemented by the top level TMC components accessed by OSO components during normal operations. A command, an attribute, or a pipe exposed by a TMC component may not be needed by an OSO component. Such commands, attributes, and pipes are marked with the blue background in the respective tables.

2 References

2.1 Applicable Documents

N.A.

2.2 Reference Documents

The following documents are referenced in this document. In the event of conflict between the contents of the referenced documents and this document, this document shall take precedence.

[RD1] 000-000000-010, SKA1 Control System Guidelines, Rev 01, Dated 05 May 2017

[RD2] The TANGO Control System Manual, Version 9.2, Dated 14 Jan 2016

[RD3] Object Management Group, Common Object Request Broker Architecture (CORBA) Specification: Part 1 - Interfaces, Version 3.3.

[RD4] P Hintjens, ØMQ Reference Guide, viewed 17-08-2016, http://zguide.zeromq.org/page:all

[RD5] 000-000000-012, SKA1 TANGO Naming Conventions, Rev 01, Dated 05 May 2017

[RD6] 000-000000-011, SKA1 TANGO Developers Guideline, Rev 01, Dated 05 May 2017

[RD7] Clements et al, Documenting Software Architectures - Views and Beyond, Addison Wesley, Second Edition

[RD8] 300-000000-021, SKA1 MID CSP to TM Interface Control Document, Rev 03, Dated 6 Mar 18

[RD9] T0000-0000-AR-017, TMC Software Architecture Document, Rev 02, Dated 29 Jun 2018

[RD10] SKA-TEL-SKO-0000860, SKA Observing Control, Rev B

[RD11] 300-000000-002, SKA1 MID SDP - CSP - Interface Control Document, Rev 04, Dated 21 May 2018

[RD12] T4000-0000-AR-002, OSO Software Architecture Document, Rev 02, Dated 29 Jun 2018

[RD13] 300-000000-029, SKA1 Interface Control Document SDP to TM MID, Rev 03, Dated 17 Jan 2018

[RD14] Time scales and formats supported by astropy.time, http://docs.astropy.org/en/stable/time/

3 Acronyms List

Table 1: List of Acronyms

Term	Definition
API	Application Programming Interface
CBF	Correlator Beamformer
CDR	Critical Design Review
CN	Central Node
CSP	Central Signal Processor
FSP	Frequency Slice Processor
GUI	Graphical User Interface
ICD	Interface Control Document
NA	Not Applicable
OET	Observation Execution Tool
OSO	Observatory Science Operation
PSS	Pulsar Search Engine
PST	Pulsar Timing Engine

Term	Definition
SAD	Software Architecture Document
SB	Scheduling Block
SCM	SKA Control Model
SDP	Science Data Processor
SKA	Square Kilometre Array
SN	Subarray Node
TACO	Telescope and Accelerator Controlled with Objects
TAI	International Atomic Time
TANGO	TACO Next Generation Object system
TBD	To Be Determined
TM	Telescope Manager
TMC	Telescope Monitor and Control
UTC	Coordinated Universal Time

4 Interface Implementation Description

4.1 Interface Overview

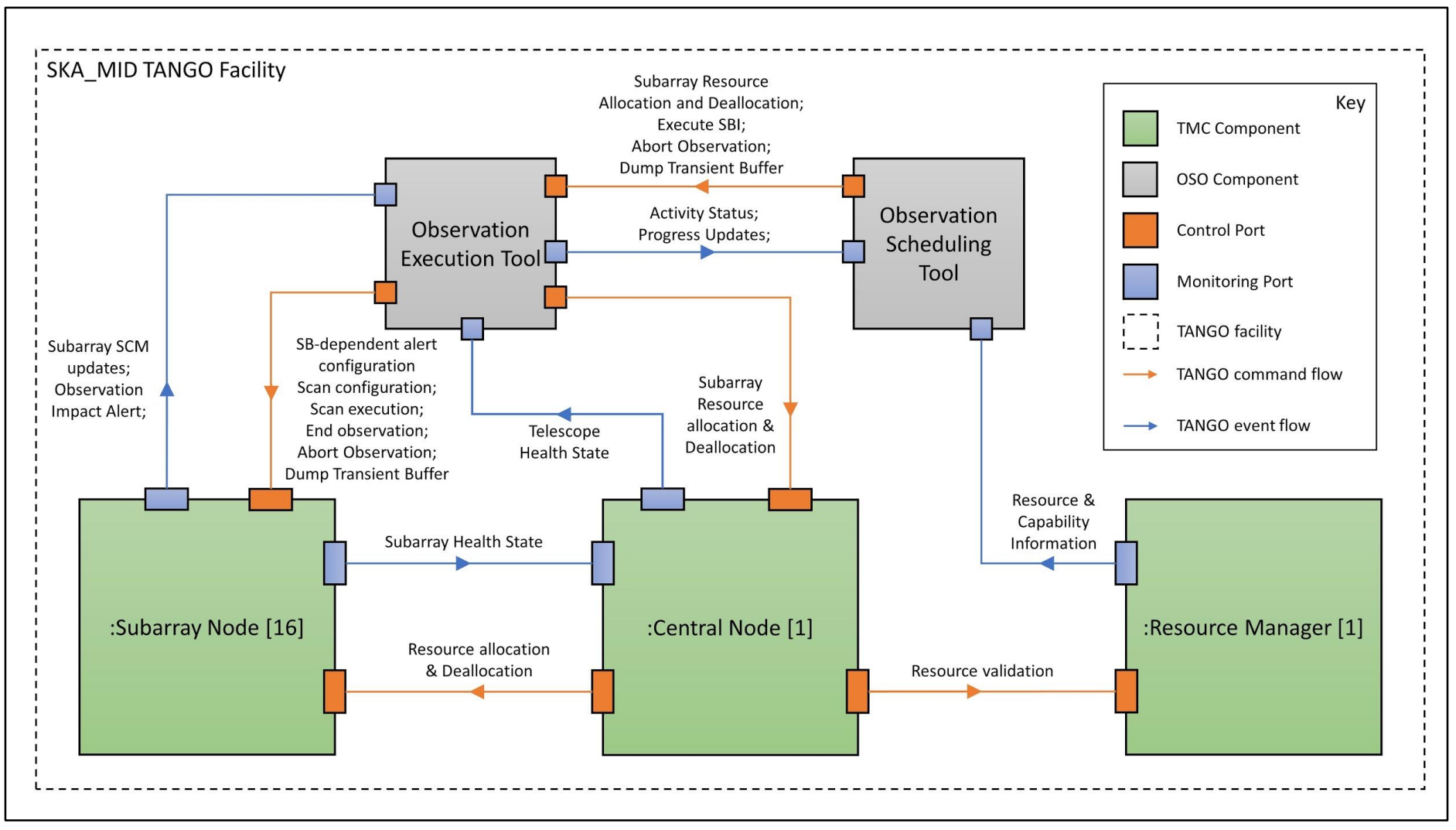


Figure 1: TMC - OSO Interface Overview

Figure 1 depicts the interface between OSO and TMC[1]. The two subsystems, TMC and OSO, interact during observation scheduling and execution. The role of TMC components is to provide the interface, and OSO components are the consumers of the interface.

For observation scheduling, OST obtains information regarding health state of resources, resources availability, capability health state and availability status (idle or in-use) from the Resource Manager. For conducting observations, OET interacts with the Central Node and a Subarray Node. The Central Node provides an interface to allocate/deallocate resources for observations whereas a Subarray Node provides interface for observation execution.

[1] The figure also depicts the interface between OST and OET. Both, OST and OET, are OSO applications and so their internal interface is documented in OSO SAD [RD12], and not in this TMC-OSO ICD.

Allocation/Deallocation of resources to a Subarray Node always happen via the Central Node[2]. Once the resources are allocated to a Subarray, OET interacts directly with the corresponding Subarray Node for subsequent observation control (such as to configure a scan, execute a scan, end an SB, and abort the observation). For additional details, refer Section 4 in Volume II of [RD9].

The subsequent sections describe the TANGO APIs implemented by the Central Node, a Subarray Node, and the Resource Manager.

4.2 TANGO Facility Context

OSO components as well as TMC components reside in the same TANGO facility, i.e. the SKA_MID Telescope Facility. The details of the SKA_MID facility TANGO Configuration Database are:

1. Host: `ska_mid`
2. Port: 10000

4.3 Central Node TANGO Device Interface

Central Node is the top - level TANGO Device in M&C hierarchy. Primary responsibilities of the Central Node are:

1. To allocate/deallocate resources to/from a Subarray Node as per the instructions from the Observation Execution Tool.
2. To distribute and coordinate the control within the Telescope for operations such as startup, shutdown, power management, and so on.
3. To implement the operator instructions for safety critical and non-critical interventions (stow all antennas, inhibit antennas, power down sensitive components on power and cooling failures)
4. To derive and report the aggregated Telescope Health State based on the information (such as Element Health State, Capabilities Health State) reported by Elements.
5. To derive and report Telescope Availability and Telescope Utilization Factor.

The interface is between a TANGO[3] [RD2] Client and a TANGO Device. The TANGO Device exposes TANGO commands and attributes via the interface to clients. Central Node implements the standard TANGO API, aligned with the [RD1]. The roles of the interfacing components are:

- TANGO Clients: OET, Telescope UI
- TANGO Device: Central Node

The client uses requests to read or write TANGO device attributes, and to invoke TANGO device commands. Synchronous and asynchronous communication relating to TANGO device attributes and commands are based on the CORBA [RD3] architecture.

Event-based communication (refer [RD2], Section 4.6) is implemented with the ØMQ library (refer [RD4]). The TANGO client is the event subscriber, and the TANGO device is the event publisher.

Communication via pipes between a client and the TANGO device is synchronous: a client, for a specific pipe, requests the data and blocks until the TANGO device sends the data via the pipe (refer [RD1] par. 4.5).

4.3.1 Central Node TANGO Device Name

The Central Node TANGO Device name is `ska_mid/tm_central/central_node` (in accordance with [RD5]).

4.3.2 Central Node TANGO Device Properties

SKA Project wide properties mandated for TANGO Devices are specified in [RD6]. Central Node device properties are documented in Table 2.

Table 2: Central Node TANGO Device Properties

Property Name	Property Description
SkaLevel	1
CentralAlarmHandler	Name of the Central Alarm Handler TANGO Device. Required to create Proxy
TMAlarmHandler	Name of the TM Alarm Handler TANGO Device. Required to create Proxy
TMMidSubarrayNodes	List of TM_Mid Subarray TANGO Device Names
NumDishes	Number of dishes commissioned.
DishLeafNodePrefix	Prefix used to obtain list of dish leaf node devices deployed from the TANGO database. Value of this property → *ska_mid/tm_leaf_node/d*
CspMasterLeafNode	TANGO Device Name of the CSP Master Leaf Node. Required to create Proxy.
SdpMasterLeafNode	TANGO Device Name of the SDP Master Leaf Node. Required to create Proxy.

[2] Alternative design approach is that OET instructs Subarray Node directly to allocate or release resources. This architectural decision will be re-visited during the prototyping phase (to be done post CDR) or construction phase.

[3] Where this document refers to TANGO, TANGO 9 is implied.

4.3.3 Central Node TANGO Device States and Modes

Central Node implements the standard set of state and mode attributes defined by the SKA Control Model (refer [RD1], Section 5.8). Table 3 describes state and mode implementation and indicates compliance with the SKA Control Model. Column **A** indicates Access rights, i.e. whether the attributes is Read-Write (R/W) or Read-Only (R/O). Column **M** indicates whether the attribute is Memorized (Y) or not (N). The value of the Memorized attribute is preserved over restart and shut-down. These indicators are exposed as TANGO attributes by the Central Node.

Table 3: Central Node TANGO Device States and Modes

Attribute	Range	A	M	Description and Comments
healthState		R/O	N	Central Node derives the Telescope Health State from the *TM Health State* and the *Health State* of the Observation Resource Units (as reported by other Elements to TM) (refer TM_REQ_336). **Note:** TM Monitor, a component of TM Services, derives and reports *TM Health State*.
	~~UNKNOWN~~			UNKNOWN Telescope Health State is never reported by the Central Node. However, Telescope UI may report the Telescope Health State as UNKNOWN when it is not able to communicate with the Central Node TANGO Device.
	OK			Telescope functionalities are available for use.
	DEGRADED			Telescope functionalities are partially available. It is calculated based on number or percent of available dishes, the number or percent of the baselines available, by sensitivity or some other criterion. The exact criterion will be worked out during the construction phase.
	FAILED			Telescope is unable to perform core functionality. Difference between FAILED and DEGRADED will be clearly identified during the construction phase.
adminMode		R/W	Y	Set by the Observatory operations to indicate availability of Telescope for science observing.
	ONLINE			Telescope is available for science observing.
	MAINTENANCE			Indicates that entire Telescope is not available for science observing due to upgrades, scheduled maintenance, off-line calibration, observations for engineering/commissioning, software updates or testing.
	~~OFFLINE~~			Not applicable.
	~~NOT_FITTED~~			Not applicable.
	RESERVED			Reported by a STANDBY Central Node TANGO Device.
state		R/O	N	Reports the operational state of the Central Node.
	INIT			INIT state is reported by Central Node when it is performing initialization. In this state, Central Node is not fully operational.
	ON			Upon successful initialization, the Central Node transits to ON state. In this state, Central Node is fully operational.
	~~STANDBY~~			Used to indicate that an entity is in LOW_POWER state. It is not applicable to the Central Node as it is a software component.
	ALARM			State is set to ALARM by the TANGO core when at least one attribute has quality factor ATTR_WARN or ATTR_ALARM and TANGO Device state is ON. When the ALARM quality factor for all attributes is NORMAL, the TANGO core sets the TANGO Device state back to ON.
	FAULT			State is set to FAULT when an unrecoverable error is detected which most probably requires human intervention to recover, rendering the device unusable. This may not be applicable to the Central Node TANGO Device. Additional analysis will be done during the construction phase.
	UNKNOWN			UNKNOWN state is never reported by the Central Node TANGO Device. However, Telescope UI may report the Central Node State as UNKNOWN when it is not able to communicate with the Central Node TANGO Device.
obsMode	-	-	-	It specifies the observation mode. Not applicable for Telescope as a whole, implemented per Subarray.
obsState	-	-	-	It specifies the observation state. Not applicable for Telescope as a whole, implemented per Subarray.
simulationMode	FALSE, TRUE	R/W	Y	If TRUE, Central Node operates as a Simulator from the point of view of monitor and control. Commands are accepted, but are not sent to the lower level devices (such as Subarray Node, Leaf Nodes) in the control hierarchy.

Attribute	Range	A	M	Description and Comments
controlMode	`REMOTE`, ~~`LOCAL`~~	R/O	Y	Implemented as read-only and always reported as `REMOTE`; Value `LOCAL` is not supported.
testMode	`NORMAL`, `Custom values`	R/W	Y	Test modes will be identified and documented during the Construction Phase.

4.3.4 Central Node TANGO Device Commands

Central Node implements:

1. Standard set of TANGO Device commands as described in [RD2], Section 6.4.2.1 and appendix A.7.
2. Standard set of SKA TANGO Device commands as described in [RD6], Section 10.2.
3. Command specific to the Central Node as described in Table 4.

Note: The TANGO pattern suggests to reflect the successful command completion by changing/updating attributes and/or states, whereas non-compliance or failure is reported by raising exceptions (not via output arguments).

Table 4: Central Node TANGO Device Commands

Name	Input Type	Input Arguments	Output Type	Description
ReleaseResources	DevString (JSON String)	JSON string comprises of: 1. SubarrayID 2. List of Receptors, 3. List of PSS Beams, 4. List of PST Beams, 5. List of VLBI Beams 6. ReleaseALL 7. TBD	DevVoid	This command is to release resources from a Subarray. The resource details are sent in a JSON format (described in the subsequent section).
AssignResources	DevString (JSON String)	JSON string comprises of: 1. SubarrayID 2. List of Receptors, 3. List of PSS Beams, 4. List of PST Beams, 5. List of VLBI Beams 6. TBD	DevVoid	This command is to assign resources in a Subarray. The resource details are sent in a JSON format (described in the subsequent section).
StowAntennas	DevStringArray	List of Receptors to be stowed	DevVoid	This command is to stow the specified receptors. This command is useful when an operator wants to stow a group of dishes.
InhibitAntennas	DevStringArray	List of Receptors to be inhibited	DevVoid	This command is to inhibit movement of the specified receptors. This command is useful when an operator wants to inhibit movement of a group of dishes.
StandByTelescope	DevVoid	N.A.	DevVoid	This command is to bring the Telescope into a `STANDBY` state (i.e. Low Power State)
StartUpTelescope	DevVoid	N.A.	DevVoid	This command is to bring the Telescope into `ON` state from the `STANDBY` state.

4.3.4.1 JSON structure of input arguments

Table 5: JSON Structure of Input Arguments

Sr. #	Command Name	JSON structure of input arguments

1	AssignResources	`{`
		`'subarrayID': 4,`
		`'dish':{`
		`'receptorIDList': [1,2,3,...,197]`
		`},`
		`'csp.pss':{`
		`'numPSSBeams':1500,`
		`'capabilityIDList': [1,2,...,1500]`
		`},`
		`'csp.pst':{`
		`'numPSTBeams':16,`
		`'capabilityIDList': [1,2,...,16]`
		`},`
		`'csp.VLBI':{`

Sr. #	Command Name	JSON structure of input arguments
		'numVLBIBeams':4, 'capabilityIDList': [1,2,...,4] } }

2	ReleaseResources	{
		'subarrayID': 4,
		'dish':{
		'receptorIDList': [1,2,3,...,197]
		},
		'csp.pss':{
		'numPSSBeams':1500,
		'capabilityIDList': [1,2,...,1500]
		},
		'csp.pst':{
		'numPSTBeams':16,
		'capabilityIDList': [1,2,...,16]
		},
		'csp.VLBI':{
		'numVLBIBeams':4,
		'capabilityIDList': [1,2,...,4]
		},
		'releaseALL': True
		}

4.3.4.1.1 Description of JSON members

Description of JSON members used in the input argument to *AssignResources* and *ReleaseResources* command is provided in Table 6.

Table 6: JSON Members used in *AssignResources* and *ReleaseResources* commands

Sr. #	JSON Member	Data type	Optional	Description
1	subarrayID	integer	N	Unique Identifier of a Subarray
2	dish	object	Y	JSON object that contains a list of receptors to be added/removed to/from a Subarray
3	receptorIDList	array	Y	A list of receptors to be added/removed to/from a Subarray
4	csp.pss	object	Y	JSON object that contains a number of PSS beams and list of PSS beam IDs to be added/removed to/from a Subarray
5	numPSSBeams	integer	Y	Number of PSS beams. Maximum value: 1500
6	capabilityIDList	array	Y	A list of PSS/PST/VLBI beam IDs to be added/removed from a Subarray
7	csp.pst	object	Y	JSON object that contains a number of PST beams and list of PST beam IDs to be added/removed to/from a Subarray.
8	numPSTBeams	integer	Y	Number of PST beams. Maximum value: 16
9	csp.VLBI	object	Y	JSON object that contains a number of VLBI beams and list of VLBI beam IDs to be added/removed to/from a Subarray
10	numVLBIBeams	integer	Y	Number of VLBI beams. Maximum value: 4
11	releaseALL	boolean	N	Indicates whether all resources allocated to Subarray should be released.

The proposed JSON structure allows the allocation/release of resources to/from a Subarray to be additive. The resources can be added or removed without tearing down the Subarray. It is possible to allocate dish1 and dish2 in two separate step to a Subarray as shown below:

```
# allocation step 1
CentralNode.AssignResources({
  'subarrayID': 4,
  'dish': {
    'receptorIDList': [1]
  }
})
# allocation step 2
CentralNode.AssignResources({
  'subarrayID': 4,
  'dish': {
```

```
    'receptorIDList': [2],
  }
})
```

VCCs do not need to be included in the resource allocation command. There is a one-to-one mapping between a Receptor and a VCC. TMC informs CSP about the receptors being used in a Subarray. CSP automatically allocates corresponding VCCs to the respective Subarray.

FSP can be shared across Subarrays, provided those Subarrays use it for the same FSP Function Mode. CSP offers maximum flexibility in that FSP modes and sub-array mappings can change at scan boundaries without having to end the SB to reallocate resources (refer CSP-TM ICD [RD8]). Hence, both allocation and configuration of FSPs is done on a per-scan basis via the *Configure* command on a Subarray Node (see Section 4.4.4.1), and not via the *AssignResources* command.[4] That is, other resources such as Dishes, PSS/PST/VLBI Beams are allocated in *AssignResources* command but, FSPs are configured and 'allocated' during *Configure* command.

The downside is that using this flexibility to the full would introduce significant complexity in execution and scheduling. For execution, the *Configure* command should be synchronised across Subarrays. That means, if one Subarray changes FSP mode, the other Subarrays using that FSP have to be ready for that change. Scheduling should also be aware of the FSP usage per scan, so it could fit other observations that match FSP usage alongside. Hence, while the TMC interface exposes the full flexibility of CSP FSPs, to minimise complexity at scheduling and execution end, OSO may choose to model FSP allocation at the level of SBs rather than per scan. For instance, scripts might only be allowed to change FSP mode if no other Subarrays are using them.

4.3.5 Central Node TANGO Device Attributes

Central Node Implements:

1. The standard set of TANGO Device attributes as described in [RD2], Appendix B.6;
2. The standard set of SCM attributes as defined in the Section 4.3.3 ;
3. Attributes specific to the Central Node as described in Table 7.

Note: The list of attributes specified in Table 7 is a preliminary list, the complete set of attributes will be worked out in the construction phase.

Table 7: Central Node TANGO Device Attributes

Name	Attribute Type	Data Type	Read/Write	Description	Attribute Properties
telescopeHealthState	Scalar	DevEnum	R	Overall Telescope Health State based on the aggregated Health State of all the Subarrays.	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN
subarray*N*HealthState	Scalar	DevEnum	R	Health State of an individual Subarray. A forwarded attribute of Subarray Node *N*.	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN
telescopeAvailability	Scalar	DevFloat	R	The ratio between the Schedulable Resources currently available to perform observational experiments (i.e. its core mission) and the total number of Schedulable Resource's commissioned (i.e. accepted for operational use) and therefore including those that are faulty, in maintenance or not ready. (Refer TM_REQ_33). **Note:** The ratio will be expressed as a fraction of 100 (i.e. in percentage)	Unit: N.A. Min value: 0.00 Max value: 100.00
telescopeUtilizationFactor	Scalar	DevFloat	R	The ratio between the Schedulable Resource's currently performing observational experiments (i.e. its core mission) and the Schedulable Resources available to perform observational experiments. (Refer TM_REQ_32) **Note:** The ratio will be expressed as a fraction of 100 (i.e. in percentage)	Unit: N.A. Min value: 0.00 Max value: 100.00
versionInfo	Scalar	DevString	R	Version information of TANGO device.	Unit: N.A.

[4] This would be modelled internally by OSO and would have no impact on the ICD.

4.3.6 Central Node TANGO Device Events

Implementation of event subscription is described in [RD2], Section 4.6. The pipe event type can be used on the TANGO device side application code to push data through a pipe. The TANGO heartbeat mechanism is described in [RD2], Section 7.4.1.

Specific events for the Central Node TANGO Device against each TANGO attribute and pipes are specified in Table 8.

Table 8: Central Node TANGO Device Events

Attribute Name	Change	Periodic	Archive	Pushed by Device		
				Archive	Change	Data Ready
operatingState	Absolute: [1][5] Relative: N.A.	Default[6]	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
telescopeHealthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
adminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
telescopeAvailability	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
telescopeUtilizationFactor	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
versionInfo	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.

4.3.7 Central Node TANGO Device Pipes

Central Node does not implement pipes. Table 9 is part of the documentation template.

Table 9: Central Node TANGO Device Pipes

Label	Description	Read/Write	Device pipe blob			
			Name	Value		
				Type	Data Type	Content

[5] Change and Archive Event Thresholds are automatically managed by TANGO for attributes of DevState, DevBoolean, DevEnum and DevString data types.

[6] Default value for Periodic event is 1000 ms.

4.4 Subarray TANGO Device Interface

Subarray Node is primary point of monitoring and control with respect to an Observation. It coordinates the configuration and execution of a scan as per the instructions from the Observation Execution Tool.

The interface is between a TANGO [RD2] Client and a TANGO Device. The TANGO Device exposes TANGO commands and attributes via the interface to clients. Subarray Node implements the standard TANGO API, aligned with the [RD1]. The roles of the interfacing components are:

- TANGO Clients: OET, Central Node, Telescope UI
- TANGO Device: Subarray Node

The client uses requests to read or write TANGO device attributes, and to invoke TANGO device commands. Synchronous and asynchronous communication relating to TANGO device attributes and commands are based on the CORBA [RD3] architecture. Event-based communication (refer [RD2], Section 4.6) is implemented with the ØMQ library (refer [RD4]). The TANGO client is the event subscriber, and the TANGO device is the event publisher.

Command based communication via pipes between the client and the TANGO device is synchronous: the client, for a specific pipe, requests the data and blocks until the TANGO device sends the data via the pipe (refer [RD1] par. 4.5).

4.4.1 Subarray TANGO Device Name

The Subarray TANGO Device name is `ska_mid/tm_subarray_node/1` (in accordance with [RD5]).

4.4.2 Subarray TANGO Device Properties

SKA Project wide properties mandated for TANGO Devices are specified in [RD6]. Subarray Node device properties are documented in Table 10.

Table 10: Subarray TANGO Device Properties

Property Name	Property Description
SkaLevel	2
DishLeafNodePrefix	Prefix used to obtain list of dish leaf node devices deployed from the TANGO database. Value of this property → *ska_mid/tm_leaf_node/d*
CspSubarrayLeafNode	TANGO Device Name of the corresponding CSP Subarray Leaf Node
SdpSubarrayLeafNode	TANGO Device Name of the corresponding SDP Subarray Leaf Node

4.4.3 Subarray TANGO Device States and Modes

Subarray Node implements the standard set of state and mode attributes defined by the SKA Control Model (refer [RD1], Section 5.8). Table 11 describes state and mode implementation and indicates compliance with the SKA Control Model. Column **A** indicates Access rights, i.e. whether the attributes is Read-Write (R/W) or Read-Only (R/O). Column **M** indicates whether the attribute is Memorized (Y) or not (N). The value of the Memorized attribute is preserved over restart and shut-down. These indicators are exposed as TANGO attributes by Subarray Node.

Table 11: Subarray TANGO Device States and Modes

Attribute	Range	A	M	Description and Comments
adminMode (Enum)		R/W	Y	Set by the Central Node or an Operator.
	`ONLINE`			The Subarray can be used for scientific observing.
	`MAINTENANCE`			The Subarray is not to be used for scientific observing, but can be used for testing and commissioning.
	`OFFLINE`			The Subarray is not to be used; adminMode=`OFFLINE`, state=`DISABLE`.
	~~`NOT_FITTED`~~			Not applicable.
	~~`RESERVED`~~			Not applicable.
state (Enum)		R/O	N	Indicates the operational state of a Subarray Node.
	`INIT`			The Subarray is being initialized.
	`OFF`			The Subarray is ‘empty’; receptors have not been allocated to the Subarray.
	`ON`			At least one receptor is allocated to the Subarray.

Attribute	Range	A	M	Description and Comments
	ALARM			The Quality Factor for at least one attribute is outside the pre-defined ALARM limits. Some functionalities may not be available.
	DISABLE			The Subarray is administratively disabled (adminMode = OFFLINE).
	FAULT			An unrecoverable fault has been detected. The Subarray is not available for use; maintainer/operator intervention is required.
	UNKNOWN			Subarray Node is unresponsive, e.g. due to loss of communication. This state is not directly reported by a Subarray Node, but may be reported by parent entity such as Central Node or Telescope UI.
healthState (Enum)		R/O	N	Indicates the health status of a Subarray. It is determined by the aggregating the health state of resources and capabilities allocated to a Subarray.
	OK			The health state of Subarray is OK and it can be used for observation.
	DEGRADED			Subarray can be used for observation, but with degraded functionality. Health State of a Subarray can be DEGRADED based on number or percent of available dishes, the number or percent of the baselines available, or some other criterion. The exact criteria will be worked out during the construction phase.
	FAILED			Subarray cannot be used for observation. Difference between FAILED and DEGRADED will be clearly identified during the construction phase.
	UNKNOWN			The health state of a Subarray cannot be determined. UNKNOWN health state is not directly reported by a Subarray Node, but may be reported by parent entity such as Central Node or Telescope UI.
obsState (Enum)		R/O	N	The Subarray Observing State indicates status related to scan configuration and execution.
	IDLE			Input data is not being processed and output products are not being generated.
	CONFIGURING			Transient state entered when a command to *configure* the Subarray is received. The Subarray leaves this state when configuration is completed.
	READY			A Subarray automatically enters into READY when scan configuration is complete; re-configuration has been completed; this implies that a Subarray is ready to start a scan.
	SCANNING			A Subarray is processing input data and generating output products. The parameters that require updates during the scan are being updated.
	ABORTED			A Subarray transitions to this state when a command Abort is received. In this state (re-)configuration, delay tracking, execution of a scan and any other on-going processing functions are stopped.
	FAULT			A Subarray transitions to this state when an unrecoverable error that requires operator intervention has been detected.
obsMode (Enum)		R/O	N	It is set as a part of the configuration of the Subarrays and capabilities and never set directly. This attribute can be multi-valued, i.e. more than one observing mode can be active in the same Subarray at the same time. For a commensal observation, a Subarray could be doing simultaneous pulsar search and pulsar timing, or simultaneous continuum imaging and pulsar timing, etc.
	IDLE			The obsMode shall be reported as IDLE when obsState is IDLE; else, it will report the appropriate value. More than one observing mode can be applicable to a Subarray for a commensal observation.
	IMG_CONTINUUM			
	IMG_SPECTRAL_LINE			
	IMG_ZOOM			
	PULSAR_SEARCH			
	TRANSIENT_SEARCH_FAST			
	TRANSIENT_SEARCH_SLOW			
	PULSAR_TIMING			

Attribute	Range	A	M	Description and Comments
	VLBI			
simulationMode (Enum)	TRUE, FALSE	R/W	Y	If TRUE, Subarray Node operates as a Simulator from the point of view of monitor and control. Commands are accepted, but are not sent to the lower level devices (Leaf Nodes) in the control hierarchy.

4.4.4 Subarray Node TANGO Device Commands

Subarray Node implements the standard set of commands as specified in:

1. Standard set of TANGO Device commands as described in [RD2], Section 6.4.2.1 and appendix A.7.
2. Standard set of SKA TANGO Device commands as described in [RD6], Section 10.2.
3. Command specific to a Subarray Node as described in the in Table 12.

Note: The TANGO pattern suggests to reflect the successful command completion by changing/updating attributes and/or states, whereas non-compliance or failure is reported by raising exceptions (not via output arguments).

Table 12: Subarray Node TANGO Device Commands

Name	Input Type	Input Arguments	Output Type	Allowed in Observing State(s)	Description
ReleaseResources	DevString (JSON String)	JSON string comprises of: 1. List of Receptors, 2. List of PSS Beams, 3. List of PST Beams, 4. List of VLBI Beams 5. ReleaseALL 6. TBD	DevVoid	IDLE	This command is to release resources from a Subarray. The resource details are sent in JSON format (described in the subsequent section). This command is invoked by Central Node, and not intended for use by OSO components.
AssignResources	DevString (JSON String)	JSON string comprises of: 1. List of Receptors, 2. List of PSS Beams, 3. List of PST Beams, 4. List of VLBI Beams 5. TBD	DevVoid	IDLE	This command is to assign resources to a Subarray. The resource details are sent in JSON format (described in the subsequent section). This command is invoked by Central Node, and not intended for use by OSO components.
Configure	DevString (JSON String)	A big JSON string comprising of scan configuration for TMC, Dish, CSP, and SDP	DevVoid	CONFIGURING, READY	Command has a single string parameter, which is a JSON string containing the configuration information. One of the items in the JSON string is scanID. On receipt of this command, a Subarray Node fans out the Configure command to allocated Dishes, CSP Subarray Controller, and SDP controller via respective leaf nodes; It also transits to obsState = CONFIGURING. The attribute configurationProgress is exposed to show the progress. When system is configured, i.e. configuratinProgress = 100, Subarray node transits to obsState = READY.
Abort	DevVoid	N.A.	DevVoid	CONFIGURING, READY, SCANNING	This is an asynchronous command that can be sent at any time. Subarray immediately transitions to the ABORTED state which is one in which the system is in a safe configuration and not generating observational data.
Scan	DevString (JSON String)	JSON string comprises of: 1. Start time, 2. Time format 3. Time scale 4. Scan duration 5. Auto Transition	DevVoid	READY	This function schedules a scan for execution on a subarray. Command has a parameter which indicates the time (TAI) at which the Scan will start. Subarray transitions to obsState = SCANNING, when the execution of a scan starts.
EndScan	DevVoid	N.A.	DevVoid	SCANNING	This can be either an automatic or an externally triggered transition after the scanning completes normally.
Reset	DevVoid	N.A.	DevVoid	ABORTED, FAULT	Command has no parameter, and is used to exit the ABORTED state. Subarray transitions to IDLE state. Resources are not affected by the Reset command: the Subarray still keeps the assigned resources but requires configuration.
EndSB	DevVoid	N.A.	DevVoid	READY	Command has no parameter. This command is used to indicate the end of the Scheduling Block.

Name	Input Type	Input Arguments	Output Type	Allowed in Observing State(s)	Description
					Resources, allocated to a Subarray, are not automatically released. Subarray transitions to `IDLE`.
OffloadTransient DataCaptureBuffer	DevString (JSON String)	JSON string comprises of: 1. Transient Epoch time, 2. Time format 3. Time scale	DevVoid	`SCANNING`	Used to trigger off-loading of the content of the Transient Buffer.

4.4.4.1 JSON structure of Input arguments

Table 13: JSON Structure of Input Arguments

Sr. #	Command Name	JSON structure of input arguments

1	AssignResources	`{`
		`'dish':{`
		`'receptorIDList': [1,2,3,...,197]`
		`},`
		`'csp.pss':{`
		`'numPSSBeams':1500,`
		`'capabilityIDList': [1,2,...,1500]`
		`},`
		`'csp.pst':{`
		`'numPSTBeams':16,`
		`'capabilityIDList': [1,2,...,16]`
		`},`
		`'csp.VLBI':{`
		`'numVLBIBeams':4,`
		`'capabilityIDList': [1,2,..,4]`
		`}`
		`}`

2	ReleaseResources	`{`
		`'dish':{`
		`'receptorIDList': [1,2,3,...,197]`
		`},`
		`'csp.pss':{`
		`'numPSSBeams':1500,`
		`'capabilityIDList': [1,2,...,1500]`
		`},`
		`'csp.pst':{`
		`'numPSTBeams':16,`
		`'capabilityIDList': [1,2,...,16]`
		`},`
		`'csp.VLBI':{`
		`'numVLBIBeams':4,`
		`'capabilityIDList': [1,2,...,4]`
		`},`
		`'releaseALL': False`
		`}`

3	Scan	`{`
		`'startTime': '2000-01-01T00:00:00.000',  # Time at which scan should start`
		`'timeFormat': 'isot',  # Specifies how an instant of time is represented.`
		`'timeScale': 'TAI'  # A specification for measuring time.`
		`'scanDuration': 10,  # scan duration in seconds`
		`'autoTransition': False  # automatic transition of Subarray to READY state`
		`}`
		
		It is decided to use time formats and time scales supported by astropy.time [RD14] package. astropy.time package supports time scales (e.g. UTC, TAI, UT1, TDB) and time representations (e.g. JD, MJD, ISO 8601) that are used in astronomy.

4	OffloadTransient DataCaptureBuffer	`{`
		`# Transient epoch time`
		`'transientEpochTime': '2000-01-01T00:00:00.000',`
		`'timeFormat': 'isot',  # Specifies how an instant of time is represented.`
		`'timeScale': 'TAI'  # A specification for measuring time.`
		`}`
		
		It is decided to use time formats and time scales supported by astropy.time [RD14] package. astropy.time package supports time scales (e.g. UTC, TAI, UT1, TDB) and time representations (e.g. JD, MJD, ISO 8601) that are used in astronomy.

Sr. #	Command Name	JSON structure of input arguments
5	Configure[7]	

```
{
      'pointing': {
            # Pattern represents Scan Type. Possible values -- siderealTrack,
#nonSiderealTrack, driftScan, fivePointScan, wideAreaMapping
            'pattern': 'siderealTrack',
            'pointingParameters': {

                  # example parameters for sidereal track
                  'siderealTrack': {
                        'ra': 143.245,
                        'dec': 47.2661,

                        # name of the source
                        'name': 'NGC5023'
                  },

                  # example parameters for non-sidereal track
                  'nonSiderealTrack': {
                        'positionTripletList': [
                              { 'ra': 143.245, 'dec': 47.2601, 'time': '2000-01-
01T00:00:00.000' },
                              { 'ra': 143.235, 'dec': 47.2361, 'time': '2000-01-
01T00:01:00.000' },
                              { 'ra': 143.205, 'dec': 47.2261, 'time': '2000-01-
01T00:03:00.000' }
                        ]
                  },

                  # example parameters for fivePointScan
                  'fivePointScan': {
                        'ra': 143.245,
                        'dec': 47.2661,

                        # name of the source
                        'name': 'NGC5023',

                        # 5-point offset in arcsecs
                        'offset': 12.0
                  },

                  # example parameters for wide-area mapping, a.k.a. OTF mapping
                  'wideAreaMapping': {
                        # scanning pattern e.g.,RASTER,CROSS_RASTER,ROTATED_RASTER,
                        # LISSAJOUS, SPIRAL, etc.
                        'rasterPattern': 'RASTER',

                        # the following are example parameters for a simple
                        # rectangular RASTER configuration

                        # list of ra/decs defining an enclosed area to be mapped.
                        # Usually rectangular, but a list allows for
                        # irregularly-shaped mapping areas
                        'mapArea': [[202.4958, -48.320], [205.3770, -48.320],
                                    [205.3770, -47.439], [202.4958, -47.439]]

                        # separation between raster rows (arcsecs)
                        'rasterSpacing': 25.2,

                        # scanning velocity (arcsecs per sec)
                        'rasterScanRate’: 10.0,
                  }
            }
      },
      'dish': {
            'receiverBand': '5a',
            'centreFreq': 10000,
            'bandwidth': 2000
      },

      'tmc': {
            # Example alert thresholds which would result in an Observation Impact
            # Alert being issued if the threshold is breached.

            # Raise an alert if the wind speed > 50km/h. This configuration
            # parameter is intended as an operator alert for observations that
```

[7] JSON structure for input arguments to Configure command will evolve further during the construction phase.

Sr. #	Command Name	JSON structure of input arguments

```
            # require high pointing precision and are affected by high wind speeds.
            # This parameter is distinct from any safety-related thresholds.
            'maxWindSpeed': 50.0,

            # Threshold for the maximum number of receptors that are allowed to fail
            # before an alert is issued. Limits can be specified as an integer or
            # as a percent. Values containing a percent sign (%) specify the
            # fraction of receptors in the sub-array that can fail before an alert
            # is triggered.
            'maxReceptorFailure': '10',

            # The operator/scientist may request that some calibrations are
            # calculated, but not applied. The example below requests that pointing
            # model updates are not applied until further notice or the end of the
            # SB, whichever comes first.
            'applyPointingModelUpdates': true,
      },

      'csp': {

            'correlation': {
                  'freqChannels': ,
                  'numZoomWindows': 4,        # Number of Zoom Windows
                  'polarizationParameters': {
                        'XX': ,
                        'YY': ,
                  },
                  'fsp1': {    # configurations are provided per FSP
                        'receptorIDList':,  # (optional) list of sub-array receptors
to use for correlation
                        'freqBand':,  # frequency band ([1,2,3,4,5a,5b])
                        'frequencySliceId':,
                        'bandwidth':,
                        'centreFreq':,  # centre frequency of zoom window (KHz)
                        'tInt':,  #integration time for correlation products (ms)
                         'chanAvgMap':,
                  },
                  'fsp2': {
                     ...
                  },
                   ...
            },

            'pulsarSearch': {
                  'numBeams': ,                    # Number of beams
                  'freqBand': ,                    # Frequency Band
                  'searchFreqBand': ,              # Search Frequency Band
                  'fspIDList':,
                  'frequencySliceIDList':,
                  'tObs': ,                        #Observing Time
                  'tSampling': ,                   #Sampling Interval
                  'polarizationProductsRequired': true, # Whether polarization
products required
                  'skyCoordinates': ,
                  'pulseSearchControlParameters': ,

            },
            'pulsarTiming': {
                  'numBeams': ,               # Number of beams
                  'freqBand': ,               # Frequency Band
                  'fspIDList':,
                  'frequencySliceIDList':,
                  'tObs': ,                   # Observing Time
                  'tSampling': ,              # Sampling Interval
                  'pulsarTime': ,             # Pulsar Time
                  'pulsarTimingPeriod': ,          #Pulsar Timing Period
                  'skyCoordinates': ,
            },
            'VLBI': {
                  'channelWidth': ,                # Beam-channel bandwidth in MHz
                  'nChannels': ,                   # Number of Channels
                  'polarizationCorrection': ,
                  'numBeams': ,                    # Number of beams
                  'polarizationType': ,
                  'polarizationProductsRequired': true,
                  'wordFormat': ,                  #Word Format 2, 4, 8 bit
                  'freqResolution': ,                   # Frequency Resolution
                  'RFICorrection': ,
```

Sr. #	Command Name	JSON structure of input arguments

```
                    'internalGainAdjustment': ,

                    'fsp1': {
                         'freqBand':,  # frequency band ([1,2,3,4,5a,5b])
                         'frequencySliceID':,
                         'tInt':,  #integration time
                    },

                    'fsp2': {
                         ...
                    },
                    ...
               },
          },
          'sdp'[8]: {
               'dataCaptureConfiguration': {

               },
               'postProcessingConfiguration': {

               },
               'reProcessingConfiguration': {

               },
          },
}
```

4.4.4.1.1 Description of JSON members

JSON members used in the input argument to *AssignResources* and *ReleaseResources* command are explained in Section 4.4.4.1. JSON members used in the input argument for *Configure* and *Scan* commands are described in Table 14. Column **Modes** indicates the observing modes that require a particular JSON member.

Key for column **Modes**:
A: Continuum Imaging Observing Mode
B: Dynamic Spectrum Observing Mode
C: Imaging Transient Search Observing Mode
D: Pulsar Timing Observing Mode
E: Pulsar Search Observing Mode
F: Spectral Line Imaging Observing Mode
G: VLBI Observing Mode

Table 14: JSON members for AssignResources and ReleaseResources on Subarray Node

Name	Data Type	Modes	Description
dish	object	All	JSON object that contains Dish specific configuration
tmc	object	All	JSON object that contains TMC specific configuration
csp	object	All	JSON object that contains CSP specific configuration
sdp	object	All	JSON object that contains SDP specific configuration
pointing	object	All	JSON object that contains all the configuration required for pointing
pointingParameters	object	All	JSON object that contains parameters for different pointing patterns
correlation	object	A, F	JSON object that contains parameters for Correlation mode
pulsarSearch	object	E	JSON object that contains parameters for Pulsar Search mode
pulsarTiming	object	D	JSON object that contains parameters for Pulsar Timing mode
VLBI	object	G	JSON object that contains parameters for VLBI mode
siderealTrack	object	All	JSON object that contains pointing parameters for sidereal tracking
nonSiderealTrack	object	All	JSON object that contains pointing parameters for non-sidereal tracking
fivePointScan	object	All	JSON object that contains pointing parameters for five point scanning
wideAreaMapping	object	All	JSON object that contains pointing parameters for wide area mapping

[8] TM - SDP ICD Rev 3 [RD13] does not contain information about the configuration parameters to be sent to SDP. Hence, the structure of the JSON object containing the SDP specific configuration is not worked out. It will be updated when information is available in the future version of TM - SDP ICD.

Name	Data Type	Modes	Description
fspN	object	A, F, G	JSON object that contains configuration for the specified FSP
centreFreq	number	All	Centre Frequency In case of B, D, E, G it will be center frequency per beam In case of A, F it will be center frequency per zoom window
bandwidth	number	All	In case of B, D, E, G it will be bandwidth per beam In case of C, it will be channel bandwidth In case of A, F it will be bandwidth per zoom window
freqChannels	number	A	Number of frequency channels
channelWidth	number	G	Channel width
nChannels	number	A, C, F, G	Number of channels In case of A, F it will be number of channels per zoom window
numZoomWindows	number	A, F	Number of Zoom windows (Upto 4)
polarizationParameters		A	Polarization Parameters
polarizationCorrection	number	G	Polarization Correction required
tInt	number	A, C, F	Visibility Integration period, in ms
numBeams	number	B, D, E, G	Number of Beams
polarization(s)		B	
polarizationType	enum	G	Polarization type required (CIrcular or Linear)
freqBand	enum	B, C, D, E, F, G	Frequency Band
searchFreqBand	enum	E	Search Frequency Band
tObs	number	B, D, E	Observation Time
tSampling	number	B, D, E	Sampling Interval
spectralResolution	number	B	
temporalResolution	number	B	
polarizationProductsRequired	boolean	B, C, E, F, G	Whether polarization products are required
imagingRFIMask		C	
imagingTransientDetectionThresholds		C	
pulsarTime	number	D	
pulsarTimingPeriod	number	D	
skyCoordinates	number	D, E	Sky Coordinates
pulseSearchControlParameters		E	
wordFormat	number	G	Word Format (2,4,8 bit)
freqResolution	number	G	Frequency Resolution In case of G it is frequency resolution per VLBI beam.
rfiCorrection	number	G	
internalGainAdjustment	number	G	Internal gain adjustment required
pattern	enum	All	Type of pointing pattern (Possible values - siderealTrack, nonSiderealTrack, driftScan, fivePointScan, wideAreaMapping)
capabilityIDList	array		List of capabilities to be added
ra, dec	number		Right Ascension, Declination coordinates
offset	number		Distance between points in arcsecs
rasterScanRate	number	All	Raster Scan Rate in degree/min or degree/sec
frequencySliceID	number	A,G	Identifies the 200MHz frequency slice to be used for specified FSP
frequencySliceIDList	array	D,E	Identifies the 200MHz frequency slices to be used for specified FSPs

Name	Data Type	Modes	Description
fspIDList	array	D,E	The FSPs to be used for beamforming; the number of FSPs should be the same as the number of Frequency Slices mentioned in frequencySliceIDList.
chanAvgMap			Channel averaging per 744 channels. (Table of 20 entries, mapping channel ID: averaging factor)
maxReceptorFailure	number	N.A.	Threshold for the maximum number of receptors that are allowed to fail before an alert is issued. Limits can be specified as an integer or as a percent. Values containing a percent sign (%) specify the fraction of receptors in the Subarray that can fail before an alert is triggered.
applyPointingModelUpdates	boolean	N.A.	Whether pointing model updates are required. In some cases, the scientists may want to run an observation that SDP uses to derive new pointing model parameters, but TMC does not apply the new values.
rasterSpacing	number	All	Separation between raster rows (arcsecs)
startTime	string	All	Time at which scan should start
timeFormat	string	All	Specifies how an instant of time is represented (See formats supported by astropy.time [RD14]).
timeScale	string	All	Time scale is a specification for measuring time. (See time scales supported by astropy.time [RD14]).
scanDuration	number	All	Scan duration in seconds
autoTransition	boolean	All	This option specifies Subarray to automatically transition back to READY state when the receptors have completed their raster pattern. This would be more acc urate than trying to calculate a scan duration for raster scan. It can also be used for non-sidereal track. That is, Subarray Node would also transition back to READY when the last time/ra/dec triplet of an ephemeris has been observed. It can also be used to support an experimental mapping pattern that isn't (yet) implemented by TMC. Experimental mapping pattern could be encoded as an ephemeris.

4.4.4.1.2 Configure command JSON structure

The proposed JSON structure for the *Configure* command contains `pointing`, `dish`, `tmc`, `csp`, and `sdp` as Level 0 key. These keys contain the pointing specific configuration, Dish specific configurations, TMC specific configurations, CSP specific configuration, and SDP specific configuration for a scan respectively.

This JSON structure allows to send full configuration as well as delta configuration. For example, OET sends a full (aggregated) configuration to Subarray Node at the starting of the SB by invoking *Configure* command. However, for subsequent scans, only delta configuration could be sent where only the parameters that need to be changed are sent in JSON format.

```
#Set up the configuration for the next scan. Keep the resource configuration from the last scan, but
#change the pointing RA, DEC, and pattern.
SubArrayNode.configure({
 'pointing': {
   'pattern': 'fivePointScan',  # do a five-point scan
   'ra': 143.245,             # RA centre
   'dec': -32.25325,          # dec centre
   'offset': 12.0             # 5-point offset in arcsecs
 }
})
```

TMC doesn't blindly pass configurations provided by the OET to the respective Elements; it could also parse, modify, and/or take actions based on those values (setting up pointing models by parsing ra/dec commands, etc.).

In some cases, especially during the Commissioning phase, OET may want to configure a particular Dish in a different way. This can be achieved in two way. First, OET sends command directly to the corresponding dish leaf node. The proposed architecture does not prevent it. But it is not recommended. Second, a JSON key inside the *Configure* command is used to specify configuration for a particular dish. OET creates a delta configuration and sends it to Subarray Node. This aspect will be further analysed during the construction phase. Below is a possible example:

```
# set RA on a single dish instance
SubArrayNode.configure({
```

```
  'dish.instance1': {
    'ra': 143.245  # RA in degrees
  }
})
```

4.4.4.1.2.1 Commensal Aspect

As stated in Section 4.5 of Volume II of [RD9], the TMC architecture remains same while executing commensal observation. OET combines the resource configurations required for the different observations in the same JSON string. For an Imaging and Pulsar Search commensal observation, `csp` key will include both `correlation` and `pulsarSearch` child keys containing the CSP configuration for imaging and pulsar search observation.

4.4.5 Subarray TANGO Device Attributes

Subarray Node implements:

1. The standard set of TANGO Device attributes as described in [RD2], Appendix B.6;
2. The standard set of SCM attributes as defined in the Section 4.4.3 ;
3. Subarray TANGO Device-Specific Attributes as described in Table 15.

Note: The list of parameters specified below is the preliminary list, the complete set of attributes will be provided in construction.

Table 15: Subarray TANGO Device Attributes

Name	Attribute Type	Data Type	Read/Write	Description	Attribute Properties
scanID	Scalar	DevString	R	Provided as a part of scan configuration.	Unit: N.A. Min value: N.A. Max value: N.A. Min alarm: N.A. Max alarm: N.A. Min warning: N.A. Max warning: N.A.
sbID	Scalar	DevString	R	Scheduling Block ID.	Unit: N.A. Min value: N.A. Max value: N.A. Min alarm: N.A. Max alarm: N.A. Min warning: N.A. Max warning: N.A.
configurationProgress	Scalar	DevFloat	R	It is updated periodically as the configuration progresses. configurationProgress reaching 100% (and obsState transitions to `READY`) indicates configuration completion.	Unit: N.A. Min Value: 0 Max Value: 100.00
configurationDelayExpected	Scalar	DevFloat	R	The expected time to transition from obsState in `CONFIGURING` to `READY`.	Unit: seconds Min value: 0
configurationDelayMeasured	Scalar	DevFloat	R	The actual/measured time to transition from obsState in `CONFIGURING` to `READY`.	Unit: seconds Min value: 0
obsImpactAlert[9]	Scalar	DevBoolean	R	Attribute exposed by a Subarray Node during the conditions affecting the observation. OET subscribes to it.	Unit: N.A. Min value: N.A. Max value: N.A. Min alarm: N.A. Max alarm: N.A. Min warning: N.A. Max warning: N.A.
receptorIDList	Spectrum	DevUShort	R	List of Receptor IDs	Unit: N.A. Min value: 1 Max value: 197 Min alarm: N.A. Max alarm: N.A. Min warning: N.A. Max warning: N.A.
numPSSBeams	Scalar	DevUShort	R	Number of PSS beams	Unit: N.A. Min value: 0 Max value: 1500
listPSSBeamID	Spectrum	DevUShort	R	List of PSS beam IDs	Unit: N.A.

[9] Details of an Observation Impact Alert is provided via TANGO Pipe named obsImpactInfo (See Section 4.4.7). The number and type of SB-dependent alarms will evolve with time. Usage of a TANGO Pipe to provide the details of an Observation Impact Alert allows evolution of SB-dependent alarms without changing the interface, i.e. only the structure of Pipe will change.

Name	Attribute Type	Data Type	Read/Write	Description	Attribute Properties
					Min value: N.A. Max value: N.A. Min alarm: N.A. Max alarm: N.A. Min warning: N.A. Max warning: N.A.
numPSTBeams	Scalar	DevUShort	R	Number of PST beams	Unit: N.A. Min value: 0 Max value: 16
listPSTBeamID	Spectrum	DevUShort	R	List of PST beam IDs	Unit: N.A. Min value: 1 Max value: 16 Min alarm: N.A. Max alarm: N.A. Min warning: N.A. Max warning: N.A.
numVLBIBeams	Scalar	DevUShort	R	Number of VLBI beams	Unit: N.A. Min value: 0 Max value: 4
listVLBIBeamID	Spectrum	DevUShort	R	List of VLBI beam IDs	Unit: N.A. Min value: 1 Max value: 4 Min alarm: N.A. Max alarm: N.A. Min warning: N.A. Max warning: N.A.
cbfSdpRate	Scalar	DevFloat	R	Output data rate on CSP_Mid.CBF to SDP_Mid link. Formula to calculate it, is mentioned in [RD11] .	Unit: GB/s Min value: 0 Max value: 772.30
pssSdpRate	Scalar	DevFloat	R	Output data rate on CSP_Mid.PSS to SDP_Mid link. Formula to calculate it, is mentioned in [RD11] .	Unit: GB/s Min value: 0 Max value: 60.56
pstSdpRate	Scalar	DevFloat	R	Output data rate on CSP_Mid.PST to SDP_Mid link. Formula to calculate it, is mentioned in [RD11] .	Unit: GB/s Min value: 0 Max value: 9.44
versionInfo	Scalar	DevString	R	Version information of TANGO device.	Unit: N.A.

4.4.6 Subarray TANGO Device Events

Implementation of event subscription is described in [RD2], Section 4.6. The pipe event type can be used on the TANGO device side application code to push data through a pipe. The TANGO heartbeat mechanism is described in [RD2], Section 7.4.1.

Specific events for a Subarray Node TANGO Device against each TANGO attribute and pipes are specified in Table 16.

Table 16: Subarray TANGO Device Events

Attribute Name	Change	Monitoring	Archive	Pushed by Device		
				Archive	Change	Data Ready
adminMode	Absolute: [1][10] Relative: N.A.	Default[11]	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
state	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
healthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A.	N.A.	N.A.	N.A.

[10] Change and Archive Event Thresholds are automatically managed by TANGO for attributes of DevState, DevBoolean, DevEnum and DevString data types.

[11] Default value for Periodic event is 1000 ms.

Attribute Name	Change	Monitoring	Archive	Pushed by Device		
				Archive	Change	Data Ready
			Period: 3600000			
obsState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
obsMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
simulationMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
configurationProgress	Absolute: 1.00 Relative: N.A.	Default	Absolute: 1.00 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
configurationDelayExpected	Absolute: 5.00 Relative: N.A.	Default	Absolute: 5.00 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
configurationDelayMeasured	Absolute: 5.00 Relative: N.A.	Default	Absolute: 5.00 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
obsImpactAlert	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorIDList	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
numPSSBeams	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
listPSSBeamID	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
numPSTBeams	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
listPSTBeamID	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
numVLBIBeams	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
listVLBIBeamID	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
cbfSdpRate	Absolute: 0.001 Relative: N.A.	Default	Absolute: 0.001 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
pssSdpRate	Absolute: 0.001 Relative: N.A.	Default	Absolute: 0.001 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
pstSdpRate	Absolute: 0.001 Relative: N.A.	Default	Absolute: 0.001 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
versionInfo	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.

4.4.7 Subarray TANGO Device Pipes

Table 17 shows the implementation of TANGO pipe on Subarray Node for details related to Observation Impact alerts.

Table 17: Subarray TANGO Device Pipes

Label	Description	Read/ Write	Device pipe blob			
			Name	Value		
				Type	Data Type	Content
obsImpactInfo	Provides details on the cause of the SB-dependent observation Impact Alert.	Read	windSpeedAlert	Scalar	DevBoolean	Indicates that the wind speed is above the specified threshold. Observations that require high pointing precision may get affected.
			receptorFailureAlert	Scalar	DevBoolean	Indicates that the number of receptors failed is above the threshold for the maximum number of receptors that are allowed to fail.
			failedReceptorList	Spectrum	DevUShort	List of failed receptor IDs
			failedFSPList	Spectrum	DevUShort	List of failed FSP IDs

4.5 Resource Manager TANGO Device Interface

Resource Manager is central to Telescope Operations from the point of view of resource management and maintaining the resource matrix. Primary responsibilities of the Resource Manager are:

1. To provide the view of resource and capability availability that includes current, up-to-date availability of all systems.
2. To update resource and Capability health status, number of Capabilities, their availability in resource matrix whenever their status changes or whenever they are allocated to a sub-array.

The interface is between a TANGO [RD2] Client and a TANGO Device. The TANGO Device exposes TANGO commands and attributes via the interface to clients. Central Node implements the standard TANGO API, aligned with the [RD1]. The roles of the interfacing components are:

- TANGO Clients: OST, Telescope GUI
- TANGO Device: Resource Manager

The client uses requests to read or write TANGO device attributes, and to invoke TANGO device commands. Synchronous and asynchronous communication relating to TANGO device attributes and commands are based on the CORBA [RD3] architecture. Event-based communication (refer [RD2], Section 4.6) is implemented with the ØMQ library (refer [RD4]). The TANGO client is the event subscriber, and the TANGO device is the event publisher.

Command based communication via pipes between the client and the TANGO device is synchronous: the client, for a specific pipe, requests the data and blocks until the TANGO device sends the data via the pipe (refer [RD1] par. 4.5).

4.5.1 Resource Manager TANGO Device Name

The Resource Manager TANGO Device name is `ska_mid/tm_central/resource_manager` (in accordance with [RD5]).

4.5.2 Resource Manager TANGO Device Properties

SKA Project wide properties mandated for TANGO Devices are specified in [RD6]. Resource Manager device properties are documented in Table 18.

Table 18: Resource Manager TANGO Device Properties

Property Name	Property Description
SkaLevel	2
DishLeafNodePrefix	Prefix used to obtain list of dish leaf node devices deployed from the TANGO database. Value of this property → *ska_mid/tm_leaf_node/d*
CspMasterLeafNode	TANGO Device Name of the CSP Master Leaf Node. Required to create Proxy.
SdpMasterLeafNode	TANGO Device Name of the SDP Master Leaf Node. Required to create Proxy.

4.5.3 Resource Manager TANGO Device States and Modes

Resource Manager implements the standard set of state and mode attributes defined by the SKA Control Model (refer [RD1], Section 5.8). Table 19 describes state and mode implementation and indicates compliance with the SKA Control Model. Column **A** indicates Access rights, i.e. whether the attributes is Read-Write (R/W) or Read-Only (R/O). Column **M** indicates whether the attribute is Memorised (Y) or not (N). The value of the Memorised attribute is preserved over restart and shut-down. These indicators are exposed as TANGO attributes by the Central Node.

Table 19: Resource Manager TANGO Device States and Modes

Attribute	Range	A	M	Description and Comments
healthState		R/O	N	Health Status of the Resource manager. It indicates ability of resource manager to maintain resources and capabilities availability information.
	~~UNKNOWN~~			UNKNOWN health State is never reported by the Resource Manager. However, Telescope UI may report the Resource Manager Health State as UNKNOWN when it is not able to communicate with the Resource Manager TANGO Device.
	OK			Health of the resource manager is OK. It provides full functionality of maintaining and reporting (on request) resource and capability availability information.
	~~DEGRADED~~			Not applicable.
	FAILED			Resource manager is unable to maintain and report resources and capabilities availability information.
adminMode		R/W	Y	Set by Operator.
	ONLINE			Resource Manager can be used for management of resources and capabilities.
	~~MAINTENANCE~~			Not applicable.
	OFFLINE			Resource Manager is not to be used; adminMode = OFFLINE, state = DISABLE.

Attribute	Range	A	M	Description and Comments
	~~NOT_FITTED~~			Not Applicable.
	~~RESERVED~~			Not Applicable.
state		R/O	N	Indicates operational state of Resource Manager.
	INIT			Resource Manager is being initialized.
	ON			Resource Manager is available for management of resources and capabilities.
	~~STANDBY~~			Not applicable.
	ALARM			The Quality Factor for at least one attribute is outside the pre-defined ALARM limits. Some functionalities may not be available.
	DISABLE			Resource Manager is administratively disabled. [adminMode = OFFLINE]
	FAULT			An unrecoverable fault has been detected. The Resource Manager is not available for use; Operator intervention may be required.
	UNKNOWN			UNKNOWN state is never reported by the Resource Manager TANGO Device. However, Telescope UI may report the Resource Manager State as UNKNOWN when it is not able to communicate with the Resource Manager TANGO Device.
obsMode	-	-	-	Not applicable.
obsState	-	-	-	Not applicable.
simulationMode	FALSE, TRUE	R/W	Y	If TRUE, Resource Manager operates as a simulator.
controlMode	REMOTE, ~~LOCAL~~	R/O	Y	Implemented as read-only and always reported as REMOTE; Value LOCAL is not supported.
testMode	NORMAL, Custom values	R/W	Y	Test modes will be identified and documented during the Construction Phase.

4.5.4 Resource Manager TANGO Device Commands

Resource Manager implements:

1. Standard set of TANGO Device commands as described in [RD2], Section 6.4.2.1 and appendix A.7.
2. Standard set of SKA TANGO Device commands as described in [RD6], Section 10.2.
3. Command specific to the Resource Manager as described in the Table 20.

Note: The TANGO pattern suggests to reflect the successful command completion by changing/updating attributes and/or states, whereas non-compliance or failure is reported by raising exceptions (not via output arguments).

Table 20: Resource Manager TANGO Device Commands

Name	Input Type	Output Type	Description
GetReceptorAvailability	DevVarUShortArray	DevVarBooleanArray	Command to obtain availability of specified receptors. Input Argument: List of Receptor IDs Output Argument: List consisting of availability of each Receptor specified in the input.
GetVCCAvailability	DevVarUShortArray	DevVarBooleanArray	Command to obtain availability of specified VCCs. Input Argument: List of VCC IDs Output Argument: List consisting of availability of each VCC specified in the input.
GetPSSBeamsAvailability	DevVarUShortArray	DevVarBooleanArray	Command to obtain availability of specified PSS beams. Input Argument: List of PSS Beam IDs Output Argument: List consisting of availability[12] of each PSS Beam ID specified in the input.

[12] Availability of Search beam is determined without considering whether the FSP is available to produce it. It is assumed that the number of FSPs required to produce requested Search beams will be configured in PSS mode as a part of scan configuration.

Name	Input Type	Output Type	Description
GetPSTBeamsAvailability	DevVarUShortArray	DevVarBooleanArray	Command to obtain availability of specified PST beams. Input Argument: List of PST Beam IDs Output Argument: List consisting of availability[13] of each PST Beam ID specified in the input.
GetVLBIBeamsAvailability	DevVarUShortArray	DevVarBooleanArray	Command to obtain availability of specified VLBI beams. Input Argument: List of VLBI Beam IDs Output Argument: List consisting of availability[14] of each VLBI Beam ID specified in the input.
GetReceptorBandAvailability	DevVarUShortArray	DevVarBooleanArray	Command to obtain availability of an observing band for specified receptors. Input Argument: InArg[0] = Band ID InArg[1] = Receptor ID InArg[n] = Receptor ID Output Argument: List consisting availability of an observing band for each receptor specified in the input.
GetVCCBandAvailability	DevVarUShortArray	DevVarBooleanArray	Command to obtain availability of an observing band for specified VCCs. Input Argument: InArg[0] = Band ID InArg[1] = VCC ID InArg[n] = VCC ID Output Argument: Output Argument: List consisting availability of an observing band for each VCC specified in the input.
GetSubarrayReceptorList	DevUShort	DevVarUShortArray	Command to obtain receptors allocated in the specified Subarray. Input Argument: Subarray ID Output Argument: List of receptors allocated in the specified Subarray
GetSubarrayPssBeamsList	DevUShort	DevVarUShortArray	Command to obtain PSS beams allocated in the specified Subarray. Input Argument: Subarray ID Output Argument: List of PSS beams allocated in the specified Subarray
GetSubarrayPstBeamsList	DevUShort	DevVarUShortArray	Command to obtain PST beams allocated in the specified Subarray. Input Argument: Subarray ID Output Argument: List of PST beams allocated in the specified Subarray
GetSubarrayVlbiBeamsList	DevUShort	DevVarUShortArray	Command to obtain VLBI beams allocated in the specified Subarray. Input Argument: Subarray ID Output Argument: List of VLBI beams allocated in the specified Subarray
GetSubarrayFSPUsageInfo	DevUShort	DevVarLongStringArray	Command to obtain information on the Function Mode for each FSP w.r.t. a Subarray. Input Argument: Subarray ID Output Argument: Information on function mode for each FSP being used in the specified Subarray. Output argument is a structure with a sequence of *long* and sequence of *string*. Sequence of *long* indicates the FSP IDs and Sequence of *string* indicates the corresponding FSP function modes. OutArgLValue[0] = FSP ID OutArgLValue[1] = FSP ID …. OutArgLValue[n] = FSP ID OutArgLValue[0] = FSP Function Mode OutArgLValue[1] = FSP Function Mode …. OutArgLValue[n] = FSP Function Mode

4.5.5 Resource Manager TANGO Device Attributes

Resource Manager Implements:

[13] Availability of Timing beam does not consider whether the FSP is available to produce it. It is assumed that the number of FSPs required to produce requested Timing beams will be configured in PST mode as a part of scan configuration.

[14] Availability of VLBI beam does not consider whether the FSP is available to produce it. It is assumed that the number of FSPs required to produce requested VLBI beams will be configured in VLBI mode as a part of scan configuration.

1. The standard set of TANGO Device attributes as described in [RD2], Appendix B.6;
2. The standard set of SCM attributes as defined in the Section 4.5.3 ;
3. Attributes specific to the Resource Manager as described in Table 21.

Note: The list of attributes specified below is a preliminary list, the complete set of attributes will be worked out in the construction phase.

Resource Manager TANGO Device Attributes model the information captured in Resource Matrix (refer Resource and Capability Management View in [RD9]).

Table 21: Resource Manager TANGO Device Attributes

Name	Attribute Type	Data Type	Read/ Write	Description	Attribute Properties
receptorList	Spectrum	DevUShort	R	List of all the receptor IDs.	Unit: N.A. Min value: 1 Max value: 197
receptorHealthState	Spectrum	DevEnum	R	List consisting of Health State of each receptor.	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN
receptorAdminMode	Spectrum	DevEnum	R	List consisting of Admin Mode of each receptor.	Unit: N.A. Min value: 1 Max value: 5 Enum Labels: ONLINE, OFFLINE, MAINTENANCE, NOT_FITTED, RESERVED
receptorState	Spectrum	DevEnum	R	List consisting of Operating State of each receptor.	Unit: N.A. Min value: 1 Max value: 7 Enum Labels: ON, ALARM, INIT, STANDBY, DISABLE, FAULT, UNKNOWN
receptorSubarrayMembership	Spectrum	DevUShort	R	List consisting of Subarray Membership of each receptor.	Unit: N.A. Min value: 0 Max value: 16
receptorAvailability	Spectrum	DevBoolean	R	List depicting the availability of each receptor for an observation. Availability of each receptor is computed by Resource Manager as per the rule mentioned in Resource and Capability Management View.	Unit: N.A. Min value: N.A. Max value: N.A.
receptorBandHealthState	Image	DevEnum	R	Matrix depicting health state of each Band capability for each receptor. A row corresponds to a Band and Column corresponds to a Receptor.	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN
receptorBandAdminMode	Image	DevEnum	R	Matrix depicting Admin Mode of each Band capability for each receptor. A row corresponds to a Band and Column corresponds to a Receptor.	Unit: N.A. Min value: 1 Max value: 5 Enum Labels: ONLINE, OFFLINE, MAINTENANCE, NOT_FITTED, RESERVED
receptorBandState	Image	DevEnum	R	Matrix depicting State of each Band capability for each receptor. A row corresponds to a Band and Column corresponds to a Receptor.	Unit: N.A. Min value: 1 Max value: 7 Enum Labels: ON, ALARM, INIT, OFF, DISABLE, FAULT,

Name	Attribute Type	Data Type	Read/ Write	Description	Attribute Properties
					UNKNOWN
receptorBandAvailability	Image	DevBoolean	R	Matrix depicting health state of each Band capability for each receptor. A row corresponds to a Band and Column corresponds to a Receptor.	Unit: N.A. Min value: N.A. Max value: N.A.
vccList	Spectrum	DevUShort	R	List of all the VCC IDs.	Unit: N.A. Min value: 1 Max value: 198
vccHealthState	Spectrum	DevEnum	R	List consisting of Health State of each VCC.	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN
vccAdminMode	Spectrum	DevEnum	R	List consisting of Admin Mode of each VCC.	Unit: N.A. Min value: 1 Max value: 5 Enum Labels: ONLINE, OFFLINE, MAINTENANCE, NOT_FITTED, RESERVED
vccState	Spectrum	DevEnum	R	List consisting of Operating State of each VCC.	Unit: N.A. Min value: 1 Max value: 7 Enum Labels: ON, ALARM, INIT, STANDBY, DISABLE, FAULT, UNKNOWN
vccSubarrayMembership	Spectrum	DevUShort	R	List consisting of Subarray Membership of each VCC.	Unit: N.A. Min value: 0 Max value: 16
vccAvailability	Spectrum	DevBoolean	R	List depicting the availability of each VCC for an observation. Availability of each VCC is computed by Resource Manager as per the rule mentioned in Resource and Capability Management View.	Unit: N.A. Min value: N.A. Max value: N.A.
vccBandHealthState	Image	DevEnum	R	Matrix depicting health state of each Band capability for each VCC. A row corresponds to a Band and Column corresponds to a VCC.	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN
vccBandAdminMode	Image	DevEnum	R	Matrix depicting Admin Mode of each Band capability for each VCC. A row corresponds to a Band and Column corresponds to a VCC.	Unit: N.A. Min value: 1 Max value: 5 Enum Labels: ONLINE, OFFLINE, MAINTENANCE, NOT_FITTED, RESERVED
vccBandState	Image	DevEnum	R	Matrix depicting State of each Band capability for each VCC. A row corresponds to a Band and Column corresponds to a VCC.	Unit: N.A. Min value: 1 Max value: 7 Enum Labels: ON, ALARM, INIT, OFF, DISABLE, FAULT, UNKNOWN
vccBandAvailability	Image	DevBoolean	R	Matrix depicting health state of each Band capability for each VCC. A row corresponds to a Band and Column corresponds to a VCC.	Unit: N.A. Min value: N.A. Max value: N.A.

Name	Attribute Type	Data Type	Read/ Write	Description	Attribute Properties
fspList	Spectrum	DevUShort	R	List of all the FSP IDs.	Unit: N.A. Min value: 1 Max value: 27
fspHealthState	Spectrum	DevEnum	R	List consisting of Health State of each FSP.	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN
fspAdminMode	Spectrum	DevEnum	R	List consisting of Admin Mode of each FSP.	Unit: N.A. Min value: 1 Max value: 5 Enum Labels: ONLINE, OFFLINE, MAINTENANCE, NOT_FITTED, RESERVED
fspState	Spectrum	DevEnum	R	List consisting of Operating State of each FSP.	Unit: N.A. Min value: 1 Max value: 7 Enum Labels: ON, ALARM, INIT, STANDBY, DISABLE, FAULT, UNKNOWN
fspModeCorr	Image	DevBoolean	R	Matrix depicting usage of FSPs in Correlation mode across Subarrays. FSPs are columns, Subarrays are rows.	Unit: N.A. Min value: N.A. Max value: N.A.
fspModePSS	Image	DevUShort	R	Matrix depicting usage of FSPs in PSS mode across Subarrays. FSPs are columns, Subarrays are rows.	Unit: N.A. Min value: 0 Max value: 192
fspModePST	Image	DevUShort	R	Matrix depicting usage of FSPs in PST mode across Subarrays. FSPs are columns, Subarrays are rows.	Unit: N.A. Min value: 0 Max value: 16
fspModeVLBI	Image	DevUShort	R	Matrix depicting usage of FSPs in VLBI mode across Subarrays. FSPs are columns, Subarrays are rows.	Unit: N.A. Min value: 0 Max value: 2
fspModePSSUnusedBeams	Spectrum	DevUShort	R	Depicts number of PSS beams that can still be produced by an FSP being used in PSS mode, i.e. number of unused PST beams for each FSP being used in PST mode.	Unit: N.A. Min value: 0 Max value: 192
fspModePSTUnusedBeams	Spectrum	DevUShort	R	Depicts number of PST beams that can still be produced by an FSP being used in PST mode, i.e. number of unused PST beams for each FSP being used in PST mode.	Unit: N.A. Min value: 0 Max value: 16
fspModeVLBIUnusedBeams	Spectrum	DevUShort	R	Depicts number of PSS beams that can still be produced by an FSP being used in VLBI mode, i.e. number of unused PST beams for each FSP being used in PST mode.	Unit: N.A. Min value: 0 Max value: 20
searchBeamList	Spectrum	DevUShort	R	List of all the Search Beam IDs	Unit: N.A. Min value: 1 Max value: 1500
searchBeamHealthState	Spectrum	DevEnum	R	List consisting of Health State of each Search Beam	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN
searchBeamAdminMode	Spectrum	DevEnum	R	List consisting of Admin Mode of each Search Beam	Unit: N.A. Min value: 1

Name	Attribute Type	Data Type	Read/ Write	Description	Attribute Properties
					Max value: 5 Enum Labels: ONLINE, OFFLINE, MAINTENANCE, NOT_FITTED, RESERVED
searchBeamState	Spectrum	DevEnum	R	List consisting of Operating State of each Search Beam	Unit: N.A. Min value: 1 Max value: 7 Enum Labels: ON, ALARM, INIT, OFF, DISABLE, FAULT, UNKNOWN
searchBeamSubarrayMembership	Spectrum	DevUShort	R	List consisting of Subarray Membership of each Search Beam	Unit: N.A. Min value: 0 Max value: 16
searchBeamAvailability	Spectrum	DevBoolean	R	List depicting the availability of each Search Beam for an observation. Availability of each Search Beam is computed by Resource Manager as per the rule mentioned in Resource and Capability Management View.	Unit: N.A. Min value: N.A. Max value: N.A.
timingBeamList	Spectrum	DevUShort	R	List of all the Timing Beam IDs	Unit: N.A. Min value: 1 Max value: 16
timingBeamHealthState	Spectrum	DevEnum	R	List consisting of Health State of each Timing Beam	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN
timingBeamAdminMode	Spectrum	DevEnum	R	List consisting of Admin Mode of each Timing Beam	Unit: N.A. Min value: 1 Max value: 5 Enum Labels: ONLINE, OFFLINE, MAINTENANCE, NOT_FITTED, RESERVED
timingBeamState	Spectrum	DevEnum	R	List consisting of Operating State of each Timing Beam	Unit: N.A. Min value: 1 Max value: 7 Enum Labels: ON, ALARM, INIT, OFF, DISABLE, FAULT, UNKNOWN
timingBeamSubarrayMembership	Spectrum	DevUShort	R	List consisting of Subarray Membership of each Timing Beam	Unit: N.A. Min value: 0 Max value: 16
timingBeamAvailability	Spectrum	DevBoolean	R	List depicting the availability of each Timing Beam for an observation. Availability of each Timing Beam is computed by Resource Manager as per the rule mentioned in Resource and Capability Management View.	Unit: N.A. Min value: N.A. Max value: N.A.
vlbiBeamList	Spectrum	DevUShort	R	List of all the VLBI Beam IDs	Unit: N.A. Min value: 1
vlbiBeamHealthState	Spectrum	DevEnum	R	List consisting of Health State of each VLBI Beam	Unit: N.A. Min value: 1 Max value: 4 Enum Labels: OK, DEGRADED, FAILED, UNKNOWN

Name	Attribute Type	Data Type	Read/ Write	Description	Attribute Properties
vlbiBeamAdminMode	Spectrum	DevEnum	R	List consisting of Admin Mode of each VLBI Beam	Unit: N.A. Min value: 1 Max value: 5 Enum Labels: ONLINE, OFFLINE, MAINTENANCE, NOT_FITTED, RESERVED
vlbiBeamState	Spectrum	DevEnum	R	List consisting of Operating State of each VLBI Beam	Unit: N.A. Min value: 1 Max value: 7 Enum Labels: ON, ALARM, INIT, OFF, DISABLE, FAULT, UNKNOWN
vlbiBeamSubarrayMembership	Spectrum	DevUShort	R	List consisting of Subarray Membership of each VLBI Beam	Unit: N.A. Min value: 0 Max value: 16
vlbiBeamAvailability	Spectrum	DevBoolean	R	List depicting the availability of each VLBI Beam for an observation. Availability of each Timing Beam is computed by Resource Manager as per the rule mentioned in Resource and Capability Management View.	Unit: N.A. Min value: N.A. Max value: N.A.
cbfSdpSubarrayLinkRate	Spectrum	DevFloat	R	List consisting of output data rate of each Subarray on CSP_Mid.CBF to SDP_Mid link	Unit: GB/s Min value: 0 Max value: 772.30
cbfSdpTotalLinkRate	Scalar	DevFloat	R	Total output data rate on CSP_Mid.CBF to SDP_Mid link	Unit: GB/s Min value: 0 Max value: 772.30
pssSdpSubarrayLink	Spectrum	DevFloat	R	List consisting of output data rate of each Subarray on CSP_Mid.PSS to SDP_Mid link	Unit: GB/s Min value: 0 Max value: 60.56
pssSdpTotalLinkRate	Scalar	DevFloat	R	Total output data rate on CSP_Mid.PSS to SDP_Mid link	Unit: GB/s Min value: 0 Max value: 60.56
pstSdpSubarrayLink	Spectrum	DevFloat	R	List consisting of output data rate of each Subarray on CSP_Mid.PSS to SDP_Mid link	Unit: GB/s Min value: 0 Max value: 9.44
pstSdpTotalLinkRate	Scalar	DevFloat	R	Total output data rate on CSP_Mid.PST to SDP_Mid link	Unit: GB/s Min value: 0 Max value: 9.44
versionInfo	Scalar	DevString	R	Version information of TANGO device.	Unit: N.A.

4.5.6 Resource Manager TANGO Device Events

Implementation of event subscription is described in [RD2], Section 4.6. The pipe event type can be used on the TANGO device side application code to push data through a pipe. The TANGO heartbeat mechanism is described in [RD2], Section 7.4.1.

Specific events for the Resource Manager TANGO Device against each TANGO attribute and pipes are specified in Table 22.

Table 22: Resource Manager TANGO Device Events

Attribute Name	Change	Periodic	Archive	Pushed by Device		
				Archive	Change	Data Ready
operatingState	Absolute: [1][15] Relative: N.A.	Default[16]	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.

[15] Change and Archive Event Thresholds are automatically managed by TANGO for attributes of DevState, DevBoolean, DevEnum and DevString data types.

[16] Default value for Periodic event is 1000 ms.

Attribute Name	Change	Periodic	Archive	Pushed by Device		
				Archive	Change	Data Ready
adminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
healthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorList	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorHealthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorAdminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorSubarrayMembership	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorAvailability	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorBandHealthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorBandAdminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorBandState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
receptorBandAvailability	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vccList	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vccHealthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vccAdminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vccState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vccSubarrayMembership	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vccAvailability	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vccBandHealthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A.	N.A.	N.A.	N.A.

Attribute Name	Change	Periodic	Archive	Pushed by Device		
				Archive	Change	Data Ready
			Period: 3600000			
vccBandAdminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vccBandState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vccBandAvailability	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspList	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspHealthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspAdminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspModeCorr	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspModePSS	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspModePST	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspModeVLBI	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspModePSSUnusedBeams	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspModePSTUnusedBeams	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
fspModeVLBIUnusedBeams	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
searchBeamList	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
searchBeamHealthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
searchBeamAdminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
searchBeamState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.

Attribute Name	Change	Periodic	Archive	Pushed by Device		
				Archive	Change	Data Ready
searchBeamSubarrayMembership	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
searchBeamAvailability	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
timingBeamList	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
timingBeamHealthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
timingBeamAdminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
timingBeamState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
timingBeamSubarrayMembership	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
timingBeamAvailability	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vlbiBeamList	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vlbiBeamHealthState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vlbiBeamAdminMode	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vlbiBeamState	Absolute: [1] Relative: N.A.	Default	Absolute: [1] Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vlbiBeamSubarrayMembership	Absolute: 1 Relative: N.A.	Default	Absolute: 1 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
vlbiBeamAvailability	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
cbfSdpSubarrayLinkRate	Absolute: 0.001 Relative: N.A.	Default	Absolute: 0.001 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
cbfSdpTotalLinkRate	Absolute: 0.001 Relative: N.A.	Default	Absolute: 0.001 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
pssSdpSubarrayLink	Absolute: 0.001 Relative: N.A.	Default	Absolute: 0.001 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
pssSdpTotalLinkRate	Absolute: 0.001 Relative: N.A.	Default	Absolute: 0.001 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
pstSdpSubarrayLink	Absolute: 0.001 Relative: N.A.	Default	Absolute: 0.001 Relative: N.A.	N.A.	N.A.	N.A.

Attribute Name	Change	Periodic	Archive	Pushed by Device		
				Archive	Change	Data Ready
			Period: 3600000			
pstSdpTotalLinkRate	Absolute: 0.001 Relative: N.A.	Default	Absolute: 0.001 Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.
versionInfo	Absolute: N.A. Relative: N.A.	Default	Absolute: N.A. Relative: N.A. Period: 3600000	N.A.	N.A.	N.A.

4.5.7 Resource Manager TANGO Device Pipes

Resource Manager does not implement pipes. Table 23 is part of the documentation template.

Table 23: Resource Manager TANGO Device Pipes

Label	Description	Read/Write	Device pipe blob			
			Name	Value		
				Type	Data Type	Content

5 Appendix A: Software Engineering Institute Software Interface Documentation Recommendations

This section describes how recommendations from [RD7], Section 7.3, “A Standard Organisation for Interface Documentation” are addressed by this document.

5.1. Interface Identity

The interface between OSO components and TMC components is based on the TANGO framework. This document describes the data exchange between the OSO components and TMC components in terms of the constituent interfaces of the TANGO Devices on TMC side, that make up the interface between OSO and TMC.

Each TANGO Device exposed via this interface has a unique name in accordance with [RD5] . The TANGO Device name is specified for each TMC TANGO Device described in this document.

5.2. Resources

Resources (such as commands, attributes, properties) are exposed via TANGO Devices. The following subsections describe how resource syntax, semantics, and error handling are addressed.

5.2.1. Resource Syntax

TANGO command syntax is described by: name, input type, output type, and description.

TANGO attribute syntax is described by: name, attribute type (scalar, array, etc.), data type (TANGO primitive type), format (for representation on GUI), read/write, description (enumerated types are defined here).

TANGO attribute properties are described by: attribute unit, min value, max value, warning min & max values, alarm min and max values.

5.2.2. Resource Semantics

The purpose of each TANGO Device is described with a statement in relevant sections.

For each TANGO command the document specifies: description of the command, and in which states/modes the command is applicable/allowed.
For instance, client sends command to device; device returns immediately; while command is executed, either an exception is sent to client, or some attribute value is updated to reflect state change. The following statement is added to the TANGO commands sections: responses are not used to notify the originator of a command regarding the failure to execute the command; the TANGO Device uses exceptions for that purpose.

Description is provided for each TANGO attribute.

5.3. Error Handling

Failures will be reported based on the FMECA analysis. These are monitoring points that:

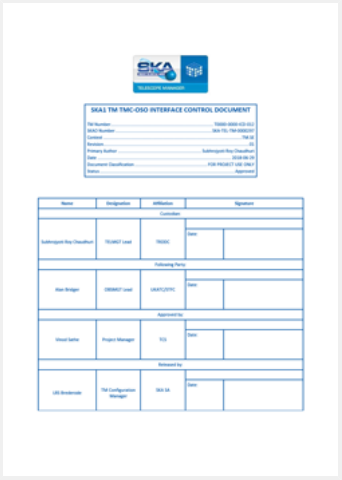
- identify occurrence of failures,
- predict occurrence of failures (i.e. monitoring hard drive usage to predict failure),
- indicate usage as input to planning of preventive maintenance tasks aid in fault-finding.

5.4. Variability

It is envisaged that the OSO - TMC interface will remain constant at a framework level, since the narrow interface for the specific TANGO version, except perhaps a change from TANGO Version 9 to Version 10. Changes may be expected if/when SKA is updated to support additional observing modes.

5.5. Quality attribute characteristics

After Subarray resource allocation is complete, OET sends instructions directly to a Subarray Node, rather than via the Central Node. The alternative is to preserve strict hierarchy, such that the Central Node is involved in relaying every OET command. This would (a) make the central node a single point of failure for all observations, (b) increase central node workload, without value addition since it will merely relay OET commands (c) increase latency (number of hops) and reduce reliability (more potential points of failure). To avoid this, it is decided that after Subarray resource allocation is complete, OET sends instructions directly to a Subarray Node.



T0000-0000-ICD-012-01_TMC_OSO_ICD

Adobe Sign Document History 07/11/2018

Created:	07/10/2018
By:	Ray Brederode (ray.brederode@ska.ac.za)
Status:	Signed
Transaction ID:	CBJCHBCAABAAt-xudISTMwsoDwWlknaQuYj6jHmu26Go

"T0000-0000-ICD-012-01_TMC_OSO_ICD" History

Document created by Ray Brederode (ray.brederode@ska.ac.za)
07/10/2018 - 7:32:51 AM PDT- IP address: 196.24.39.242

Document emailed to Ray Brederode (ray.brederode@ska.ac.za) for signature
07/10/2018 - 7:34:23 AM PDT

Document emailed to Subhrojyoti Roy Chaudhuri (subhrojyoti.ska@gmail.com) for signature
07/10/2018 - 7:34:23 AM PDT

Document emailed to Vinod Sathe (vinod@ncra.tifr.res.in) for signature
07/10/2018 - 7:34:23 AM PDT

Document emailed to Alan Bridger (alan.bridger@stfc.ac.uk) for signature
07/10/2018 - 7:34:24 AM PDT

Document e-signed by Ray Brederode (ray.brederode@ska.ac.za)
Signature Date: 07/10/2018 - 7:34:40 AM PDT - Time Source: server- IP address: 196.24.39.242

Document viewed by Alan Bridger (alan.bridger@stfc.ac.uk)
07/10/2018 - 7:43:58 AM PDT- IP address: 195.194.120.166

Document e-signed by Alan Bridger (alan.bridger@stfc.ac.uk)
Signature Date: 07/10/2018 - 7:44:27 AM PDT - Time Source: server- IP address: 195.194.120.166

Document viewed by Vinod Sathe (vinod@ncra.tifr.res.in)
07/10/2018 - 8:50:04 AM PDT- IP address: 66.249.82.76

Document e-signed by Vinod Sathe (vinod@ncra.tifr.res.in)
Signature Date: 07/10/2018 - 8:50:49 AM PDT - Time Source: server- IP address: 42.106.198.119

Document viewed by Subhrojyoti Roy Chaudhuri (subhrojyoti.ska@gmail.com)
07/10/2018 - 10:41:31 AM PDT- IP address: 64.233.173.14

Document viewed by Subhrojyoti Roy Chaudhuri (subhrojyoti.ska@gmail.com)
07/11/2018 - 0:17:23 AM PDT- IP address: 64.233.173.17

Document e-signed by Subhrojyoti Roy Chaudhuri (subhrojyoti.ska@gmail.com)
Signature Date: 07/11/2018 - 0:19:18 AM PDT - Time Source: server- IP address: 42.106.199.139

Signed document emailed to Ray Brederode (ray.brederode@ska.ac.za), Subhrojyoti Roy Chaudhuri (subhrojyoti.ska@gmail.com), Alan Bridger (alan.bridger@stfc.ac.uk), and Vinod Sathe (vinod@ncra.tifr.res.in)
07/11/2018 - 0:19:18 AM PDT