



TANGO Device Server

SR Tune monitor selector
User's Guide

SRTuneSelector Class

Revision: - Author: vedder
Implemented in C++ - CVS repository: ESRF

Introduction:

This class is used to select a Tune H/V value and store it into data collector.

Class Inheritance:

- Tango::Device_4Impl
  - SRTuneSelector

Properties:

Device Properties		
Property name	**Property type**	**Description**
SourceList	Array of string	This is a 4 field entry list containing all device/attribute that can be selected as source for Tune value.
Dc_destination	Tango::DEV_STRING	Data collector signal destination.
Dc_storage_period	Tango::DEV_USHORT	This is the delay between two storage in data collector. Unit is seconds.

Device Properties Default Values:

Property Name	**Default Values**
SourceList	No default value
Dc_destination	No default value
Dc_storage_period	5

There is no Class properties.

States:

States	
Names	**Descriptions**
ON	
FAULT	

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
SourceIndex: This is the index into the sourceList property of the entry that will be used as source for tune value.	DEV_SHORT	READ_WRITE	No
SourceAttribute: This is the attribute used as source. It's a confort attribute.	DEV_STRING	READ	No
DataCollectorName: This is the property dc_destination value mapped into this attribute. This is a 'confort' attribute.	DEV_STRING	READ	No
Tune: This 'confort' attribute contains the value read on the source, that will be stored into data collector.	DEV_DOUBLE	READ	No

Commands:

More Details on commands....

Device Commands for Operator Level		
Command name	**Argument In**	**Argument Out**
Init	DEV_VOID	DEV_VOID
State	DEV_VOID	DEV_STATE
Status	DEV_VOID	CONST_DEV_STRING

1 - Init

- **Description:** This commands re-initialise a device keeping the same network connection.
  After an Init command executed on a device, it is not necessary for client to re-connect to the device.
  This command first calls the device *delete_device()* method and then execute its *init_device()* method.
  For C++ device server, all the memory allocated in the *nit_device()* method must be freed in the *delete_device()* method.
  The language device desctructor automatically calls the *delete_device()* method.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  DEV_VOID : none.

- **Command allowed for:**
  - Tango::ON
  - Tango::FAULT

2 - State

- **Description:** This command gets the device state (stored in its *device_state* data member) and returns it to the caller.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  DEV_STATE : State Code

- **Command allowed for:**
  - Tango::ON
  - Tango::FAULT

3 - Status

- **Description:** This command gets the device status (stored in its *device_status* data member) and returns it to the caller.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  CONST_DEV_STRING : Status description

- **Command allowed for:**
  - Tango::ON
  - Tango::FAULT