



TANGO Device Server

Highest level of the RF Control system. User's Guide

RFSrrf Class

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

Introduction:

Class for the SRRF application, collecting data from various transmitters to compute RF switches.

Class Identification:

- **Contact :** at esrf.fr - vedder
- **Class Family :** RadioFrequency (Specific Specific)
- **Platform :** All Platforms
- **Bus :** Not Applicable

Class Inheritance:

- Tango::Device_4Impl
  - RFSrrf

Properties:

Class Properties		
Property name	**Property type**	**Description**
Tra1DeviceName	Tango::DEV_STRING	Transmitter 1 Tango device name.
Tra2DeviceName	Tango::DEV_STRING	Transmitter 2 Tango device name.
Tra3DeviceName	Tango::DEV_STRING	Transmitter 3 Tango device name.
Bypass_Tra1_DeviceName	Tango::DEV_STRING	This is the taco device name of the server that handle RF Bypass, when TRA1 is delivering RF for 4 Cavities.
Bypass_Tra2_DeviceName	Tango::DEV_STRING	This is the taco device name of the server that handle RF Bypass, when TRA2 is delivering RF for 4 Cavities.
Tra1_fast_logger	Tango::DEV_STRING	
Tra1_slow_logger	Tango::DEV_STRING	
Tra2_fast_logger	Tango::DEV_STRING	
Tra2_slow_logger	Tango::DEV_STRING	
Tra3_fast_logger	Tango::DEV_STRING	
Tra3_slow_logger	Tango::DEV_STRING	
Cav14_fast_logger	Tango::DEV_STRING	
Cav14_slow_logger	Tango::DEV_STRING	
Cav12DeviceName	Tango::DEV_STRING	Device name for cavity 12
Cav34DeviceName	Tango::DEV_STRING	
Cav56DeviceName	Tango::DEV_STRING	

Class Properties Default Values:

Property Name	**Default Values**</td>
Tra1DeviceName	sr/rf-tra/tra1
Tra2DeviceName	sr/rf-tra/tra2
Tra3DeviceName	sr/rf-tra/tra3
Bypass_Tra1_DeviceName	sr/rf-wagd/tra1-byp
Bypass_Tra2_DeviceName	sr/rf-wagd/byp
Tra1_fast_logger	SR/RF-LOGGER/TRA1-FAST
Tra1_slow_logger	SR/RF-LOGGER/TRA1-SLOW
Tra2_fast_logger	SR/RF-LOGGER/TRA2-FAST
Tra2_slow_logger	SR/RF-LOGGER/TRA2-SLOW
Tra3_fast_logger	SR/RF-LOGGER/TRA3-FAST
Tra3_slow_logger	SR/RF-LOGGER/TRA3-SLOW
Cav14_fast_logger	SR/RF-LOGGER/CAV14-FAST
Cav14_slow_logger	SR/RF-LOGGER/CAV14-SLOW
Cav12DeviceName	sr/rf-cavi/cav12
Cav34DeviceName	sr/rf-cavi/cav34
Cav56DeviceName	sr/rf-cavi/cav56

There is no Device Properties.

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
Switch1	DEV_LONG	READ	No
Switch2	DEV_LONG	READ	No
Switch3	DEV_LONG	READ	No
Switch4	DEV_LONG	READ	No
Switch5	DEV_LONG	READ	No
Switch6	DEV_LONG	READ	No
ByPassState	DEV_STATE	READ	No
DataLogging	DEV_STATE	READ	No
GlobalCavityVoltage	DEV_DOUBLE	READ_WRITE	No
Tra1Connected	DEV_BOOLEAN	READ	No
Tra2Connected	DEV_BOOLEAN	READ	No
Tra3Connected	DEV_BOOLEAN	READ	No
Tra1Settable	DEV_BOOLEAN	READ	No
Tra2Settable	DEV_BOOLEAN	READ	No
Tra3Settable	DEV_BOOLEAN	READ	No
Operation	DEV_BOOLEAN	READ	No
BypassWaveGuideDeviceName	DEV_STRING	READ	No

Spectrum Attributes			
Attribute name	**Data Type**	**X Data Length**	**Expert**
TangoSubDevicesStates	DEV_STATE	6	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
OnOperation	DEV_BOOLEAN	DEV_VOID

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:**

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:**

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:**

4 - OnOperation

- **Description:** This command tells if the RF is in operation mode. It will interact with tra1Settable, tra2Settable, tra3Settable. On operation transmitters used are not settable.

- **Argin:**
  DEV_BOOLEAN :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**