


TANGO Device Server

Mostab Regulation User's Guide

MostabRegulation Class

Revision: release_1_1_1 - Author: le
Implemented in C++

Introduction:

Class Inheritance:

- Tango::Device_3Impl
  - MostabRegulation

Properties:

Device Properties		
Property name	**Property type**	**Description**
IntensityProxy	Tango::DEV_STRING	The device on which the Machine current can be read.
IntensityAttributeName	Tango::DEV_STRING	The name of the attribute from which the Machine current will be read.
MAODeviceProxy	Tango::DEV_STRING	Name of the MAO Device on which the computed value will applied on.
MaoChannelName	Tango::DEV_STRING	The channel name on which the computed value will be applied
TDLDeviceProxy	Tango::DEV_STRING	Name of the TDL device
TDLAttributeStateName	Tango::DEV_STRING	Name of TDL device attribute to read to get the TDL state.

Device Properties Default Values:

Property Name	**Default Values**
IntensityProxy	no/intensityProxy/defined
IntensityAttributeName	noAttr/intensityName/defined
MAODeviceProxy	noMAO/deviceProxy/defined
MaoChannelName	noMAO/AttrName/defined
TDLDeviceProxy	noTDL/device/defined
TDLAttributeStateName	noTDL/attrState/defined

There is no Class properties.

States:

States	
Names	**Descriptions**
STANDBY	The TDL is closed so the device doesn’t compute and apply value.
RUNNING	The TDL is opened so the device computes and applies value on the MAO channel.

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
in1Mostab: The compute value to be written to one channel of the MAO board.	DEV_DOUBLE	READ	No
timeCycle: Delta time between two cycles.	DEV_LONG	READ_WRITE	No
constant: Constant added to the current	DEV_DOUBLE	WRITE	No
factor: Scale factor applied to the current to get a value between 0-10V	DEV_DOUBLE	WRITE	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
On	DEV_VOID	DEV_VOID
Stop	DEV_VOID	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:**
  - Tango::STANDBY
  - Tango::RUNNING

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::STANDBY
  - Tango::RUNNING

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::STANDBY
  - Tango::RUNNING

4 - On

- **Description:** Starts cycles
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_VOID : no argout
- **Command allowed for:**
  - Tango::STANDBY
  - Tango::RUNNING

5 - Stop

- **Description:** Stops cycles.
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_VOID : no argout
- **Command allowed for:**

- Tango::STANDBY
- Tango::RUNNING

ESRF - Software Engineering Group




TANGO Device Server

Mostab Regulation User's Guide

MostabRegulation Class

Revision: release_1_1_1 - Author: le
Implemented in C++

Introduction:

Class Inheritance:

- Tango::Device_3Impl
  - MostabRegulation

Properties:

Device Properties		
Property name	**Property type**	**Description**
IntensityProxy	Tango::DEV_STRING	The device on which the Machine current can be read.
IntensityAttributeName	Tango::DEV_STRING	The name of the attribute from which the Machine current will be read.
MAODeviceProxy	Tango::DEV_STRING	Name of the MAO Device on which the computed value will applied on.
MaoChannelName	Tango::DEV_STRING	The channel name on which the computed value will be applied
TDLDeviceProxy	Tango::DEV_STRING	Name of the TDL device
TDLAttributeStateName	Tango::DEV_STRING	Name of TDL device attribute to read to get the TDL state.

Device Properties Default Values:

Property Name	**Default Values**
IntensityProxy	no/intensityProxy/defined
IntensityAttributeName	noAttr/intensityName/defined
MAODeviceProxy	noMAO/deviceProxy/defined
MaoChannelName	noMAO/AttrName/defined
TDLDeviceProxy	noTDL/device/defined
TDLAttributeStateName	noTDL/attrState/defined

There is no Class properties.

States:

States	
Names	**Descriptions**
STANDBY	The TDL is closed so the device doesn’t compute and apply value.
RUNNING	The TDL is opened so the device computes and applies value on the MAO channel.

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
in1Mostab: The compute value to be written to one channel of the MAO board.	DEV_DOUBLE	READ	No
timeCycle: Delta time between two cycles.	DEV_LONG	READ_WRITE	No
constant: Constant added to the current	DEV_DOUBLE	WRITE	No
factor: Scale factor applied to the current to get a value between 0-10V	DEV_DOUBLE	WRITE	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
On	DEV_VOID	DEV_VOID
Stop	DEV_VOID	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:**
  - Tango::STANDBY
  - Tango::RUNNING

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::STANDBY
  - Tango::RUNNING

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::STANDBY
  - Tango::RUNNING

4 - On

- **Description:** Starts cycles
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_VOID : no argout
- **Command allowed for:**
  - Tango::STANDBY
  - Tango::RUNNING

5 - Stop

- **Description:** Stops cycles.
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_VOID : no argout
- **Command allowed for:**

- Tango::STANDBY
- Tango::RUNNING

ESRF - Software Engineering Group

Frame Alert

This document is designed to be viewed using the frames feature. If you see this message, you are using a non-frame-capable web client.
Link to Non-frame version.




TANGO Device Server

Mostab Regulation
Device Commands Description
MostabRegulation Class

Revision: release_1_1_1 - Author: le

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::STANDBY
  - Tango::RUNNING

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::STANDBY
  - Tango::RUNNING

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::STANDBY
  - Tango::RUNNING

4 - On

- **Description:** Starts cycles

- **Argin:**
  DEV_VOID : no argin

- **Argout:**
  DEV_VOID : no argout

- **Command allowed for:**
  - Tango::STANDBY
  - Tango::RUNNING

5 - Stop

- **Description:** Stops cycles.

- **Argin:**
  DEV_VOID : no argin

- **Argout:**
  DEV_VOID : no argout

- **Command allowed for:**
  - Tango::STANDBY

- Tango::RUNNING
