



TANGO
Device
Server

Motor control of the FMB Oxford Double Crystal Monochromator User's Guide

FMBOxfDCMMotor Class

Revision: - Author: tere29
Implemented in C++

Introduction:

Class for controlling a single motor of the Double Crystal Monochromator from FMB Oxford

Class Inheritance:

- Tango::Device_4Impl
 - FMBOxfDCMMotor

Properties:

Device Properties		
Property name	Property type	Description
FMBOxfDCMCtrlDevice	Tango::DEV_STRING	FMBOxfDCMCtrl server for RS232 interface
SimulationMode	Tango::DEV_ULONG	0 -> real mode, 1 -> simulation mode
AxisNb	Tango::DEV_LONG	Axis number
Version	Tango::DEV_LONG	0 -> old , 1 -> new

Device Properties Default Values:

Property Name	Default Values
FMBOxfDCMCtrlDevice	No default value
SimulationMode	0
AxisNb	No default value
Version	No default value

There is no Class properties.

States:

States	
Names	Descriptions
ON	
MOVING	
FAULT	

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
Position: Position in degrees for motor 1, and mm for motors 3 and 4	DEV_DOUBLE	READ_WRITE	No
AxisName	DEV_STRING	READ	No
PositionCts: Position in counts	DEV_DOUBLE	READ	No
UnitCalibration: Offset factor for the position (in units: degrees for Bragg motor, mm for perpendicular and paralel motors).	DEV_DOUBLE	READ_WRITE	No
Conversion: the conversion factor, counts = conversion*units	DEV_DOUBLE	READ_WRITE	No
UnitLimitMax: the upper motor limit	DEV_DOUBLE	READ_WRITE	No
UnitLimitMin: the lower limit of a motor	DEV_DOUBLE	READ_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
Calibrate	DEV_DOUBLE	DEV_LONG

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::MOVING
- 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::MOVING
- 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::MOVING
- Tango::FAULT

4 - Calibrate

- **Description:** Calibrate the motor: current position is calibrated to be the value given as an argument

- **Argin:**

DEV_DOUBLE : Value to be calibrated

- **Argout:**

DEV_LONG : Completion status

● **Command allowed for:**

- Tango::ON
- Tango::MOVING
- Tango::FAULT

TANGO is an open source project hosted by :
SOURCEFORGE.NET®

Core and Tools : CVS repository on tango-cs project
Device Servers : CVS repository on tango-ds project