



**TANGO Device Server**

# Check possible motor collisions User's Guide

# CollisionsSensor Class

**Revision: - Author: tere29**
**Implemented in C++**

## Introduction:

Check possible collisions before allowing the movement of a motor.

## Class Inheritance:

- Tango::Device_4Impl
  - CollisionsSensor

## Properties:

| Device Properties | | |
|---|---|---|
| **Property name** | **Property type** | **Description** |
| **P09Dptransctrans** | Tango::DEV_DOUBLE | Maximum allowed distance between ptrans and ctrans |
| **P09Vptrans** | Tango::DEV_DOUBLE | ptrans value for a different allowed distance between mu and nu. |
| **P09Vctrans** | Tango::DEV_DOUBLE | ctrans value for a different allowed distance between mu and nu. |
| **P09Dnumu** | Tango::DEV_DOUBLE | nu-mu value has to be bigger |
| **P09Dnumu1** | Tango::DEV_DOUBLE | mu-nu bigger if ptrans bigger than p09vptrans |
| **P09Dnumu2** | Tango::DEV_DOUBLE | mu-nu bigger if ptrans smaller than p09vptrans |
| **P09Dnumu3** | Tango::DEV_DOUBLE | mu-nu bigger if ctrans bigger than p09vctrans |
| **P09Dnumu4** | Tango::DEV_DOUBLE | mu-nu bigger if ctrans smaller than p09vctrans |

Device Properties Default Values:

| **Property Name** | **Default Values** |
|---|---|
| P09Dptransctrans | 0 |
| P09Vptrans | 0 |
| P09Vctrans | 0 |
| P09Dnumu | 0 |
| P09Dnumu1 | 0 |
| P09Dnumu2 | 0 |
| P09Dnumu3 | 0 |
| P09Dnumu4 | 0 |

**There is no Class properties.**

# States:

| States | |
|---|---|
| **Names** | **Descriptions** |
| **ON** | Server in normal state. |

# 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 |
| **CheckCollision** | DEVVAR_DOUBLESTRINGARRAY | 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

## 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

## 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

## 4 - CheckCollision

- **Description:** Check if the movement to the new position for the given motor is possible without collisions. Returns 0 or 1 depending if the movement can or not be done.

- **Argin:**
  **DEVVAR_DOUBLESTRINGARRAY** : Beamline identification (string), motor device name, new motor position

- **Argout:**
  **DEV_LONG** : 0 -> allows movement, 1 -> movement not allowed

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