



TANGO
Device
Server

SamplesManager User's Guide

SamplesManager Class

Revision: release_1_0_6 - Author: jean_coquet
Implemented in C++ - CVS repository: tango-ds

Introduction:

this device has got mainly 2 fonctionnalities: - the communication with the ControlBox device server, in order to get X & Z motor positions data - the storage of a Samples List. Each element of the list contains some information about one sample DON'T FORGET to create Proxy Name properties and the "orderedSampleList property" used by a client application

Class Inheritance:

- Tango::Device_4Impl
 - SamplesManager

Class Description:

this device has got mainly 2 fonctionnalities: - the communication with the ControlBox device server, in order to get X & Z motor positions data - the storage of a Samples List. Each element of the list contains some information about one sample

Properties:

Device Properties		
Property name	Property type	Description
MotorXProxyName	Tango::DEV_STRING	name of the X-GalilAxis device proxy
MotorZProxyName	Tango::DEV_STRING	name of the Z-GalilAxis device proxy
SampleListConfiguration	Array of string	each element of the list is a sample configuration. One configuration is consists of 6 fields according to the following pattern: sample name, rack number,cell number, position in cell(a,b,c,d,e), X position, Y position
OrderedSampleList	Array of string	ordered list by sample name

Device Properties Default Values:

Property Name	Default Values
MotorXProxyName	none
MotorZProxyName	none
SampleListConfiguration	No default value
OrderedSampleList	No default value

There is no Class properties.

States:

States	
Names	Descriptions
MOVING	at least one motor is moving
STANDBY	the motors are stopped
FAULT	Communication to the ControlBox is broken or at least one motor is in FAULT state
ALARM	at least one motor is in ALARM state
OFF	the motors are swithed off
UNKNOWN	at least one motor is in unexpected state

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
xPosition: X motor position	DEV_DOUBLE	READ_WRITE	No
zPosition: Z motor position	DEV_DOUBLE	READ_WRITE	No
currentSample: current sample or ----- if no sample positionned	DEV_STRING	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
MoveSample	DEV_STRING	DEV_VOID
AddSample	DEV_STRING	DEV_VOID
DeleteSample	DEV_STRING	DEV_VOID
GetSampleConfiguration	DEV_STRING	DEVVAR_STRINGARRAY
CleanCurrentConfiguration	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::MOVING
- Tango::STANDBY
- Tango::FAULT
- Tango::ALARM
- Tango::OFF
- Tango::UNKNOWN

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::MOVING
- Tango::STANDBY
- Tango::FAULT
- Tango::ALARM
- Tango::OFF
- Tango::UNKNOWN

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::MOVING
- Tango::STANDBY

- Tango::FAULT
- Tango::ALARM
- Tango::OFF
- Tango::UNKNOWN

4 - MoveSample

- **Description:** insert the sample into the beam by sending moving orders of the X & Z motors
- **Argin:**
DEV_STRING : sample name
- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**
 - Tango::MOVING
 - Tango::STANDBY
 - Tango::ALARM
 - Tango::OFF
 - Tango::UNKNOWN

5 - AddSample

- **Description:** add a sample config in the list regarding to the following pattern:
name,rack_number,cell_number_pos_in_cell,X_position,Y_position (commas are necessary)
- **Argin:**
DEV_STRING : sample name,rack number,cell number,position in cell(a,b,c,d,e), X position, Y position
- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**
 - Tango::MOVING
 - Tango::STANDBY
 - Tango::FAULT
 - Tango::ALARM
 - Tango::OFF
 - Tango::UNKNOWN

6 - DeleteSample

- **Description:** delete one sample from the list
- **Argin:**
DEV_STRING : sample name
- **Argout:**
DEV_VOID : nothing
- **Command allowed for:**
 - Tango::MOVING
 - Tango::STANDBY
 - Tango::FAULT
 - Tango::ALARM
 - Tango::OFF
 - Tango::UNKNOWN

7 - GetSampleConfiguration

- **Description:** position in cell(a,b,c,d,e), X position, Y position
- **Argin:**
DEV_STRING : sample name
- **Argout:**
DEVVAR_STRINGARRAY : sample name, rack number; cell number; position in cell(a,b,c,d,e), X position, Y position
- **Command allowed for:**
 - Tango::MOVING
 - Tango::STANDBY
 - Tango::FAULT
 - Tango::ALARM
 - Tango::OFF
 - Tango::UNKNOWN

8 - CleanCurrentConfiguration

- **Description:** Clean the sample list
- **Argin:**
DEV_VOID : nothing
- **Argout:**
DEV_VOID : nothing

● **Command allowed for:**

- Tango::MOVING
- Tango::STANDBY
- Tango::FAULT
- Tango::ALARM
- Tango::OFF
- Tango::UNKNOWN

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