



**TANGO
Device
Server**

TANGO Device Server User's Guide

HqpsGlobalState Class

**Revision: tata - Author: goudard
Implemented in Python**

Introduction:

This class imports hqps devices and calculate a global state for HQPS. The device imported are :
- sys/hqps-rotating-machine/xy * x = {4,5,6,7,8,9,10} * y = {a,b} - sys/hqps-cb/d31 -
sys/hqps-trans/ntr It also calculate the number of active rotating machines.

Class Inheritance:

- PyTango.Device_4Impl
 - HqpsGlobalState

Class Description:

This class imports hqps devices and calculate a global state for HQPS. The device imported are : -
sys/hqps-rotating-machine/xy * x = {4,5,6,7,8,9,10} * y = {a,b} - sys/hqps-cb/d31 - sys/hqps-trans/ntr
It also calculate the number of active rotating machines.

Properties:

Device Properties		
Property name	Property type	Description
Devices	Tango::DEVVAR_STRINGARRAY	Low level devices
AccumulatorDevices	Tango::DEVVAR_STRINGARRAY	List of devices names for accumulators.
AlternatorDevices	Tango::DEVVAR_STRINGARRAY	List of devices names for alternators.

Device Properties Default Values:

Property Name	Default Values
Devices	No default value
AccumulatorDevices	No default value
AlternatorDevices	No default value

There is no Class properties.

States:

States	
Names	Descriptions
ON	
OFF	When sys/hqps-cb/d31 is close.
FAULT	When a rotating is fault or when neutral tranformer is fault.
ALARM	When a rotating machine is ALARM or when neutral tranformer is ALARM.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
Available_rotating_machine	DEV_SHORT	READ	No
HqpsActivePower	DEV_SHORT	READ	No
PGUTActivePower	DEV_FLOAT	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

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

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

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

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