



**TANGO**  
Device  
Server

# Hqps Alarm Manager User's Guide

## HqpsAlarm Class

Revision: - Author: goudard  
Implemented in C++ - CVS repository: ESRF

### Introduction:

HqpsAlarm is a polled device server who access different TACO device servers to display HQPS Alarms in the Control Room.

### Class Inheritance:

- Tango::Device\_4Impl
  - HqpsAlarm

### Class Description:

## Properties:

<b>Device Properties</b>		
<b>Property name</b>	<b>Property type</b>	<b>Description</b>
<b>TemperatureThreshold</b>	Tango::DEV_FLOAT	Temperature Alarm Threshold
<b>Storage_vibThreshold</b>	Tango::DEV_FLOAT	StorageVibration Alarm Threshold
<b>PressureThreshold</b>	Tango::DEV_DOUBLE	Pressure Alarm Threshold.
<b>Voice_state</b>	Tango::DEV_SHORT	Tell if messages must be sent to the speaker (0 or 1
<b>Voice_fault_message</b>	Tango::DEV_STRING	Message sent to the speaker when the global state is FAULT
<b>Voice_warning_message</b>	Tango::DEV_STRING	Message sent to the speaker when the global state is ALARM
<b>Inhibited</b>	Tango::DEV_SHORT	Return true if there are some inhibited devices in the list.
<b>PressureThershold_KS_running</b>	Tango::DEV_SHORT	Pressure Thershold while KS is running is different from Pressure Thershold while KS is on.

## Device Properties Default Values:

<b>Property Name</b>	<b>Default Values</b>
TemperatureThreshold	No default value
Storage_vibThreshold	No default value
PressureThreshold	No default value
Voice_state	No default value
Voice_fault_message	No default value
Voice_warning_message	No default value
Inhibited	No default value
PressureThershold_KS_running	No default value

**There is no Class properties.**

## States:

<b>States</b>	
<b>Names</b>	<b>Descriptions</b>
<b>ON</b>	No alarm detected in HQPS.
<b>ALARM</b>	Alarm detected in HQPS.
<b>FAULT</b>	Fault detected in HQPS.

## **Attributes:**

<b>Scalar Attributes</b>			
<b>Attribute name</b>	<b>Data Type</b>	<b>R/W Type</b>	<b>Expert</b>
<b>VoiceState</b>	DEV_SHORT	READ	No
<b>Inhibited</b>	DEV_SHORT	READ	No

## **Commands:**

More Details on commands....

Device Commands for Operator Level		
Command name	Argument In	Argument Out
<b>Init</b>	DEV_VOID	DEV_VOID
<b>State</b>	DEV_VOID	DEV_STATE
<b>Status</b>	DEV_VOID	CONST_DEV_STRING
<b>ControlAlarm</b>	DEV_VOID	DEV_VOID
<b>DevDisableVoice</b>	DEV_VOID	DEV_VOID
<b>DevEnableVoice</b>	DEV_VOID	DEV_VOID
<b>DevInhibit</b>	DEV_STRING	DEV_VOID
<b>DevSetFaultMessage</b>	DEV_STRING	DEV_VOID
<b>DevSetAlarmMessage</b>	DEV_STRING	DEV_VOID
<b>DevUninhibit</b>	DEV_STRING	DEV_VOID
<b>DevUninhibitAll</b>	DEV_VOID	DEV_VOID
<b>DevHistory</b>	DEV_VOID	DEV_STRING
<b>DevVoiceState</b>	DEV_VOID	DEV_SHORT
<b>ReadTemp</b>	DEV_VOID	DEV_VOID
<b>ReadVibration</b>	DEV_VOID	DEV_VOID
<b>ReadEngineVibrations</b>	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::ON
  - Tango::ALARM
  - 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::ALARM
  - 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::ALARM
  - Tango::FAULT

## 4 - ControlAlarm

- **Description:** Retrieves data from TACO devices and compute data to check if they are correct every 5 minuts.
- **Argin:**  
**DEV\_VOID** :
- **Argout:**  
**DEV\_VOID** : Argout returns HQPS State.
- **Command allowed for:**
  - Tango::ON
  - Tango::ALARM

- Tango::FAULT

- Polled every 60000 ms

## 5 - DevDisableVoice

- **Description:** Command to disable message send to CTRM (not implmented).

- **Argin:**  
**DEV\_VOID :**

- **Argout:**  
**DEV\_VOID :**

- **Command allowed for:**

- Tango::ON
- Tango::ALARM
- Tango::FAULT

## 6 - DevEnableVoice

- **Description:** Command to enable message send to CTRM (not implmented).

- **Argin:**  
**DEV\_VOID :**

- **Argout:**  
**DEV\_VOID :**

- **Command allowed for:**

- Tango::ON
- Tango::ALARM
- Tango::FAULT

## 7 - DevInhibit

- **Description:** Command to inhibit this device.

- **Argin:**  
**DEV\_STRING :** Device name to inhibit for the global state calculation

- **Argout:**  
**DEV\_VOID :**

- **Command allowed for:**

- Tango::ON

- Tango::ALARM
- Tango::FAULT

## 8 - DevSetFaultMessage

- **Description:** Command to set Fault message. (not implemented)
- **Argin:**  
**DEV\_STRING :** Message send to the speaker when the device is fault.
- **Argout:**  
**DEV\_VOID :**
- **Command allowed for:**
  - Tango::ON
  - Tango::ALARM
  - Tango::FAULT

## 9 - DevSetAlarmMessage

- **Description:** Command to set alarm message. (not implemented)
- **Argin:**  
**DEV\_STRING :**
- **Argout:**  
**DEV\_VOID :**
- **Command allowed for:**
  - Tango::ON
  - Tango::ALARM
  - Tango::FAULT

## 10 - DevUninhibit

- **Description:**
- **Argin:**  
**DEV\_STRING :** Device name to activate for the global state calculation
- **Argout:**  
**DEV\_VOID :**
- **Command allowed for:**
  - Tango::ON

- Tango::ALARM
- Tango::FAULT

## 11 - DevUninhibitAll

- **Description:**
- **Argin:**  
DEV\_VOID :
- **Argout:**  
DEV\_VOID :
- **Command allowed for:**
  - Tango::ON
  - Tango::ALARM
  - Tango::FAULT

## 12 - DevHistory

- **Description:**
- **Argin:**  
DEV\_VOID :
- **Argout:**  
DEV\_STRING :
- **Command allowed for:**
  - Tango::ON
  - Tango::ALARM
  - Tango::FAULT

## 13 - DevVoiceState

- **Description:** Return the state of the speaker
- **Argin:**  
DEV\_VOID :
- **Argout:**  
DEV\_SHORT :
- **Command allowed for:**
  - Tango::ON



- Tango::ALARM
- Tango::FAULT

## 14 - ReadTemp

- **Description:** Polled command to read clutch temperatures.
- **Argin:**  
**DEV\_VOID :**
- **Argout:**  
**DEV\_VOID :**
- **Command allowed for:**
  - Tango::ON
  - Tango::ALARM
  - Tango::FAULT
- **Polled every 10000 ms**

## 15 - ReadVibration

- **Description:** Read vibrations in TACO device server.
- **Argin:**  
**DEV\_VOID :**
- **Argout:**  
**DEV\_VOID :**
- **Command allowed for:**
  - Tango::ON
  - Tango::ALARM
  - Tango::FAULT
- **Polled every 10000 ms**

## 16 - ReadEngineVibrations

- **Description:** REad and save engine vibrations if DEVRUN
- **Argin:**  
**DEV\_VOID :**
- **Argout:**  
**DEV\_VOID :**

- **Command allowed for:**

- Tango::ON
- Tango::ALARM
- Tango::FAULT

- **Polled every 240000 ms**

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