



TANGO
Device
Server

Linac cooling User's Guide

LinacCooling Class

Revision: LinacCooling-Release_1_0_0 - Author: vedder
Implemented in C++ - CVS repository: ESRF

Introduction:

This class will manage the linac cooling.

Class Inheritance:

- Tango::Device_4Impl
 - LinacCooling

Class Description:

Properties:

Device Properties		
Property name	Property type	Description
BuncherTemp_adc	Tango::DEV_STRING	ADC device to get the buncher temperature.
Daresbury	Tango::DEV_STRING	Name of the Daresbury device.
Interlocks_list	Array of double	Interlocks list.
PbuncherTemp_adc	Tango::DEV_STRING	ADC device to get the prebuncher temperature.
Pulsetime	Tango::DEV_LONG	Duration in milliseconds of the pulse to do on the pulsed relays.
SectionTemp_adc	Tango::DEV_STRING	ADC device to get the section temperature.
Stateonnum	Tango::DEV_DOUBLE	Interlock number used to determine if the device is on or not. (ex 21.1)
WaterTemp_adc	Tango::DEV_STRING	ADC device to get the water temperature.

Device Properties Default Values:

Property Name	Default Values
BuncherTemp_adc	No default value
Daresbury	No default value
Interlocks_list	No default value
PbuncherTemp_adc	No default value
Pulsetime	No default value
SectionTemp_adc	No default value
Stateonnum	No default value
WaterTemp_adc	No default value

There is no Class properties.

States:

States	
Names	Descriptions
ON	*
OFF	*
FAULT	*
UNKNOWN	

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
WaterTemp: Water Temperature	DEV_DOUBLE	READ	No
PBuncherTemp: PreBuncher Temperature	DEV_DOUBLE	READ	No
BuncherTemp: Buncher Temperature	DEV_DOUBLE	READ	No
SectionTemp: Section Temperature	DEV_DOUBLE	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
Reset	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::OFF
 - Tango::FAULT
 - 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::ON
 - Tango::OFF
 - Tango::FAULT
 - 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::ON
- Tango::OFF
- Tango::FAULT
- Tango::UNKNOWN

4 - Reset

- **Description:**

- **Argin:**

DEV_VOID :

- **Argout:**

DEV_VOID :

- **Command allowed for:**

- Tango::ON
- Tango::OFF
- Tango::FAULT
- 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