



**TANGO**  
Device  
Server

# **Libera BPM for transfer line User's Guide**

## **LiberaPulsedBPM Class**

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

### **Introduction:**

A class to get position and current from a libera device. it is a client of the BPMLibera server. It extracts the beam pulse from the DD acquisition buffer of a libera and get the positions at the time of the peak.

### **Class Inheritance:**

- Tango::Device\_4Impl
  - LiberaPulsedBPM

## Properties:

| Device Properties     |                   |                                          |
|-----------------------|-------------------|------------------------------------------|
| Property name         | Property type     | Description                              |
| <b>MAPerUnit</b>      | Tango::DEV_DOUBLE | Electrode value to mA conversion factor. |
| <b>ValidThreshold</b> | Tango::DEV_DOUBLE | Current validity Threshold               |
| <b>LiberaDevice</b>   | Tango::DEV_STRING | Name of the Libera device                |
| <b>CurrentOffset</b>  | Tango::DEV_DOUBLE | Offset used to calculate the current     |

### Device Properties Default Values:

| Property Name  | Default Values   |
|----------------|------------------|
| MAPerUnit      | No default value |
| ValidThreshold | No default value |
| LiberaDevice   | No default value |
| CurrentOffset  | 0                |

**There is no Class properties.**

## Attributes:

| Scalar Attributes    |             |            |        |
|----------------------|-------------|------------|--------|
| Attribute name       | Data Type   | R/W Type   | Expert |
| <b>Current</b>       | DEV_DOUBLE  | READ       | No     |
| <b>MaxIndex</b>      | DEV_LONG    | READ       | No     |
| <b>HorizontalPos</b> | DEV_DOUBLE  | READ       | No     |
| <b>VerticalPos</b>   | DEV_DOUBLE  | READ       | No     |
| <b>AGC</b>           | DEV_BOOLEAN | READ_WRITE | No     |
| <b>Attenuation</b>   | DEV_DOUBLE  | READ_WRITE | 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:**

## 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:**

### 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 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