



**TANGO**  
Device  
Server

# **RF Acquisition in Continuous AI mode User's Guide**

## **RFPoStMortem Class**

**Revision: release\_2\_0\_3 - Author: elattaoui  
Implemented in C++**

### **Introduction:**

Perform continuous analog input acquisition simultaneously on four channels. The device is available for boards SAI\_2005 and SAI\_2010.

### **Class Inheritance:**

- Tango::Device\_3Impl
  - RFPoStMortem

## Properties:

<b>Device Properties</b>		
<b>Property name</b>	<b>Property type</b>	<b>Description</b>
<b>BoardType</b>	Tango::DEV_USHORT	The board type ( the possible values are SAI_2005, SAI_2010).
<b>BoardNum</b>	Tango::DEV_USHORT	The number of the board in the cPCI chassis (between 0 and 7).
<b>InputRange0</b>	Tango::DEV_USHORT	The input range for channel 0 ( possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25 ).
<b>InputRange1</b>	Tango::DEV_USHORT	The input range for channel 1 ( possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25 ).
<b>InputRange2</b>	Tango::DEV_USHORT	The input range for channel 2 ( possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25 ).
<b>InputRange3</b>	Tango::DEV_USHORT	The input range for channel 3 ( possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25 ).
<b>Frequency</b>	Tango::DEV_DOUBLE	The acquisition frequency in Hertz.
<b>BufferDepth</b>	Tango::DEV_ULONG	The buffer depth in seconds for one channel.
<b>OverrunStrategy</b>	Tango::DEV_STRING	The overrun strategy (the strategy adopted in case of data lost). The possible values are NOTIFY, ABORT, TRASH, RESTART, IGNORE.
<b>TriggerMode</b>	Tango::DEV_STRING	The trigger mode. It is possible to use a start trigger or a stop trigger or no trigger. The possible values are START, STOP, NONE.
<b>TriggerSource</b>	Tango::DEV_USHORT	To choose analog (ATRIG) or digital trigger(DTRIG).
<b>ATRIGSelection</b>	Tango::DEV_USHORT	To select the type of analog trigger. The trigger can be generated when the analog trigger passes below a level or above a level. The possible values are BELOWand ABOVE.
<b>ATRIGLevel</b>	Tango::DEV_DOUBLE	The level of the analog trigger in volts.
<b>DTRIGPolarity</b>	Tango::DEV_USHORT	The digital level polarity. It can be detected on rising edge or falling edge. The possible values are RISING or FALLING.
<b>SamplesAfterTrigger</b>	Tango::DEV_DOUBLE	Number of samples to acquire after trigger Default : 10000
<b>HistoricBufferDepth</b>	Tango::DEV_ULONG	Buffer depth (in ms) of the acquisition history. Default : 1000 ms

Device Properties Default Values:

<b>Property Name</b>	<b>Default Values</b>
BoardType	No default value
BoardNum	No default value
InputRange0	No default value
InputRange1	No default value
InputRange2	No default value
InputRange3	No default value
Frequency	No default value
BufferDepth	No default value
OverrunStrategy	IGNORE
TriggerMode	No default value
TriggerSource	No default value
ATRIGSelection	No default value
ATRIGLevel	No default value
DTRIGPolarity	No default value
SamplesAfterTrigger	10000
HistoricBufferDepth	1000

**There is no Class properties.**

### **States:**

<b>States</b>	
<b>Names</b>	<b>Descriptions</b>
<b>UNKNOWN</b>	
<b>STANDBY</b>	
<b>RUNNING</b>	

## Attributes:

<b>Scalar Attributes</b>			
Attribute name	Data Type	R/W Type	Expert
<b>dataCounter:</b> This attribute is incremented each time a buffer is received (one buffer is received for the 4 channels).	DEV_LONG	READ	No
<b>overrunCounter:</b> May be not used, because the overrun strategy will be IGNORE.	DEV_LONG	READ	No
<b>errorCounter:</b> Errors can occur while the acquisition is running	DEV_LONG	READ	No
<b>triggerTimestamp:</b> Timestamp of the received trigger	DEV_STRING	READ	No
<b>timeoutCounter:</b> Counter of timeout(s)	DEV_LONG	READ	No

<b>Spectrum Attributes</b>			
Attribute name	Data Type	X Data Length	Expert
<b>acqHighResolutionCh1:</b> Buffer which contains the last data value stored with a high resolution	DEV_DOUBLE	5000000	No
<b>acqHighResolutionCh2:</b> Buffer which contains the last data value stored with a high resolution	DEV_DOUBLE	5000000	No
<b>acqHighResolutionCh3:</b> Buffer which contains the last data value stored with a high resolution	DEV_DOUBLE	5000000	No
<b>acqHighResolutionCh4:</b> Buffer which contains the last data value stored with a high resolution	DEV_DOUBLE	5000000	No

## Commands:

More Details on commands....

<b>Device Commands for Operator Level</b>		
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>Start</b>	DEV_VOID	DEV_VOID
<b>Stop</b>	DEV_VOID	DEV_VOID

## Device Commands for Expert Level Only

Command name	Argument In	Argument Out
Calibrate	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::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

### 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::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

### 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::UNKNOWN

- Tango::STANDBY
- Tango::RUNNING

## 4 - Start

- **Description:** Start acquisition.
- **Argin:**  
DEV\_VOID :
- **Argout:**  
DEV\_VOID :
- **Command allowed for:**
  - Tango::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

## 5 - Stop

- **Description:** Stop acquisition.
- **Argin:**  
DEV\_VOID :
- **Argout:**  
DEV\_VOID :
- **Command allowed for:**
  - Tango::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

## 6 - Calibrate (for expert only)

- **Description:** Calibrate hardware.
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**
  - Tango::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

---

**ESRF - Software Engineering Group**



**TANGO**  
Device  
Server

# **RF Acquisition in Continuous AI mode User's Guide**

## **RFPoortMortem Class**

**Revision: release\_2\_0\_3 - Author: elattaoui  
Implemented in C++**

### **Introduction:**

Perform continuous analog input acquisition simultaneously on four channels. The device is available for boards SAI\_2005 and SAI\_2010.

### **Class Inheritance:**

- Tango::Device\_3Impl
  - RFPoortMortem



## Properties:

<b>Device Properties</b>		
<b>Property name</b>	<b>Property type</b>	<b>Description</b>
<b>BoardType</b>	Tango::DEV_USHORT	The board type ( the possible values are SAI_2005, SAI_2010).
<b>BoardNum</b>	Tango::DEV_USHORT	The number of the board in the cPCI chassis (between 0 and 7).
<b>InputRange0</b>	Tango::DEV_USHORT	The input range for channel 0 ( possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25 ).
<b>InputRange1</b>	Tango::DEV_USHORT	The input range for channel 1 ( possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25 ).
<b>InputRange2</b>	Tango::DEV_USHORT	The input range for channel 2 ( possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25 ).
<b>InputRange3</b>	Tango::DEV_USHORT	The input range for channel 3 ( possible values are B_10, B_5, B_2_5, B_1_25, U_10, U_5, U_2_5, U_1_25 ).
<b>Frequency</b>	Tango::DEV_DOUBLE	The acquisition frequency in Hertz.
<b>BufferDepth</b>	Tango::DEV_ULONG	The buffer depth in seconds for one channel.
<b>OverrunStrategy</b>	Tango::DEV_STRING	The overrun strategy (the strategy adopted in case of data lost). The possible values are NOTIFY, ABORT, TRASH, RESTART, IGNORE.
<b>TriggerMode</b>	Tango::DEV_STRING	The trigger mode. It is possible to use a start trigger or a stop trigger or no trigger. The possible values are START, STOP, NONE.
<b>TriggerSource</b>	Tango::DEV_USHORT	To choose analog (ATRIG) or digital trigger(DTRIG).
<b>ATRIGSelection</b>	Tango::DEV_USHORT	To select the type of analog trigger. The trigger can be generated when the analog trigger passes below a level or above a level. The possible values are BELOWand ABOVE.
<b>ATRIGLevel</b>	Tango::DEV_DOUBLE	The level of the analog trigger in volts.
<b>DTRIGPolarity</b>	Tango::DEV_USHORT	The digital level polarity. It can be detected on rising edge or falling edge. The possible values are RISING or FALLING.
<b>SamplesAfterTrigger</b>	Tango::DEV_DOUBLE	Number of samples to acquire after trigger Default : 10000
<b>HistoricBufferDepth</b>	Tango::DEV_ULONG	Buffer depth (in ms) of the acquisition history. Default : 1000 ms

Device Properties Default Values:

<b>Property Name</b>	<b>Default Values</b>
BoardType	No default value
BoardNum	No default value
InputRange0	No default value
InputRange1	No default value
InputRange2	No default value
InputRange3	No default value
Frequency	No default value
BufferDepth	No default value
OverrunStrategy	IGNORE
TriggerMode	No default value
TriggerSource	No default value
ATRIGSelection	No default value
ATRIGLevel	No default value
DTRIGPolarity	No default value
SamplesAfterTrigger	10000
HistoricBufferDepth	1000

**There is no Class properties.**

### **States:**

<b>States</b>	
<b>Names</b>	<b>Descriptions</b>
<b>UNKNOWN</b>	
<b>STANDBY</b>	
<b>RUNNING</b>	

## Attributes:

<b>Scalar Attributes</b>			
<b>Attribute name</b>	<b>Data Type</b>	<b>R/W Type</b>	<b>Expert</b>
<b>dataCounter:</b> This attribute is incremented each time a buffer is received (one buffer is received for the 4 channels).	DEV_LONG	READ	No
<b>overrunCounter:</b> May be not used, because the overrun strategy will be IGNORE.	DEV_LONG	READ	No
<b>errorCounter:</b> Errors can occur while the acquisition is running	DEV_LONG	READ	No
<b>triggerTimestamp:</b> Timestamp of the received trigger	DEV_STRING	READ	No
<b>timeoutCounter:</b> Counter of timeout(s)	DEV_LONG	READ	No

<b>Spectrum Attributes</b>			
<b>Attribute name</b>	<b>Data Type</b>	<b>X Data Length</b>	<b>Expert</b>
<b>acqHighResolutionCh1:</b> Buffer which contains the last data value stored with a high resolution	DEV_DOUBLE	5000000	No
<b>acqHighResolutionCh2:</b> Buffer which contains the last data value stored with a high resolution	DEV_DOUBLE	5000000	No
<b>acqHighResolutionCh3:</b> Buffer which contains the last data value stored with a high resolution	DEV_DOUBLE	5000000	No
<b>acqHighResolutionCh4:</b> Buffer which contains the last data value stored with a high resolution	DEV_DOUBLE	5000000	No

## Commands:

More Details on commands....

<b>Device Commands for Operator Level</b>		
<b>Command name</b>	<b>Argument In</b>	<b>Argument Out</b>
<b>Init</b>	DEV_VOID	DEV_VOID
<b>State</b>	DEV_VOID	DEV_STATE
<b>Status</b>	DEV_VOID	CONST_DEV_STRING
<b>Start</b>	DEV_VOID	DEV_VOID
<b>Stop</b>	DEV_VOID	DEV_VOID

## Device Commands for Expert Level Only

Command name	Argument In	Argument Out
Calibrate	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::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

### 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::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

### 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::UNKNOWN

- Tango::STANDBY
- Tango::RUNNING

## 4 - Start

- **Description:** Start acquisition.
- **Argin:**  
DEV\_VOID :
- **Argout:**  
DEV\_VOID :
- **Command allowed for:**
  - Tango::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

## 5 - Stop

- **Description:** Stop acquisition.
- **Argin:**  
DEV\_VOID :
- **Argout:**  
DEV\_VOID :
- **Command allowed for:**
  - Tango::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

## 6 - Calibrate (for expert only)

- **Description:** Calibrate hardware.
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**
  - Tango::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

---

**ESRF - Software Engineering Group**

# Frame Alert

This document is designed to be viewed using the frames feature. If you see this message, you are using a non-frame-capable web client.  
[Link to Non-frame version.](#)



TANGO  
Device  
Server

---

# RF Acquisition in Continuous AI mode Device Commands Description RFPostMortem Class

Revision: release\_2\_0\_3 - Author: elattaoui

## 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 *init\_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::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

## 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::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

### 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::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

### 4 - Start

- **Description:** Start acquisition.
- **Argin:**  
**DEV\_VOID** :
- **Argout:**  
**DEV\_VOID** :
- **Command allowed for:**
  - Tango::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

### 5 - Stop

- **Description:** Stop acquisition.
- **Argin:**  
**DEV\_VOID** :
- **Argout:**

**DEV\_VOID :**

- **Command allowed for:**
  - Tango::UNKNOWN
  - Tango::STANDBY
  - Tango::RUNNING

## **6 - Calibrate (for expert only)**

- **Description:** Calibrate hardware.
  - **Argin:**  
**DEV\_VOID :** no argin
  - **Argout:**  
**DEV\_VOID :** no argout
  - **Command allowed for:**
    - Tango::UNKNOWN
    - Tango::STANDBY
    - Tango::RUNNING
- 

**ESRF - Software Engineering Group**