



**TANGO
Device
Server**

Tango Device Server User's Guide

ExperimentalFrame Class

**Revision: release_1_3_4 - Author: poirier
Implemented in C++**

Introduction:

This device stores the data of the experimental context

Class Inheritance:

- Tango::Device_3Impl
 - ExperimentalFrame

Properties:

**There is no Class properties.
There is no Device Properties.**

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
BeamlineLeader	DEV_STRING	READ_WRITE	No
Type	DEV_STRING	READ_WRITE	No
CommentConditions	DEV_STRING	READ_WRITE	No
CommentSample	DEV_STRING	READ_WRITE	No
Beamline	DEV_STRING	READ_WRITE	No
Group	DEV_STRING	READ_WRITE	No
ProjectCode	DEV_STRING	READ_WRITE	No
UserList	DEV_STRING	READ_WRITE	No
LocalContact	DEV_STRING	READ_WRITE	No
LogBookRef	DEV_STRING	READ_WRITE	No
currentConfig	DEV_STRING	READ	No

Spectrum Attributes			
Attribute name	Data Type	X Data Length	Expert
typeList	DEV_STRING	1000	No
beamlineLeaderList	DEV_STRING	1000	No
localContactList	DEV_STRING	1000	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
LoadConfig	DEV_STRING	DEV_VOID
SaveConfig	DEV_VOID	DEV_VOID
SaveConfigAs	DEV_STRING	DEV_VOID
DeleteConfig	DEV_STRING	DEV_VOID
GetConfigList	DEV_VOID	DEVVAR_STRINGARRAY

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:**

4 - LoadConfig

- **Description:**
- **Argin:**
DEV_STRING : configuration name
- **Argout:**
DEV_VOID :
- **Command allowed for:**

5 - SaveConfig

- **Description:**
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

6 - SaveConfigAs

- **Description:**
- **Argin:**
DEV_STRING : configuration name
- **Argout:**
DEV_VOID :
- **Command allowed for:**

7 - DeleteConfig

- **Description:**
- **Argin:**
DEV_STRING : configuration name
- **Argout:**
DEV_VOID :

- **Command allowed for:**

8 - GetConfigList

- **Description:** Returns the configurations list
- **Argin:**
DEV_VOID :
- **Argout:**
DEVVAR_STRINGARRAY :
- **Command allowed for:**

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

Tango Device Server

Device Commands Description

ExperimentalFrame Class

Revision: release_1_3_4 - Author: poirier

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:**

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:**

4 - LoadConfig

- **Description:**
- **Argin:**
DEV_STRING : configuration name
- **Argout:**
DEV_VOID :
- **Command allowed for:**

5 - SaveConfig

- **Description:**
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

6 - SaveConfigAs

- **Description:**
- **Argin:**
DEV_STRING : configuration name
- **Argout:**

DEV_VOID :

- **Command allowed for:**

7 - DeleteConfig

- **Description:**
- **Argin:**
DEV_STRING : configuration name
- **Argout:**
DEV_VOID :
- **Command allowed for:**

8 - GetConfigList

- **Description:** Returns the configurations list
- **Argin:**
DEV_VOID :
- **Argout:**
DEVVAR_STRINGARRAY :
- **Command allowed for:**

ESRF - Software Engineering Group