



**TANGO
Device
Server**

Tango Device Server User's Guide

EnumeratedAttribute Class

**Revision: 1.8 - Author: katyho
Implemented in Java**

Introduction:

This device predefines an enumerated list of values for a simple scalar attribute. Its tango interface is dynamically created from the properties `attributeLabelList` and `attributeEnumeratedValues`. The dynamically created attributes are boolean. For example : `EnumeratedLabelList="isInserted", "isExtracted"` `EnumeratedValueList = "0","100"` Then 2 attributes are created : `isInserted` and `isExtracted`. When `isInserted` is set to true then the controlled attribute is set to 0 When `isExtracted` is set to true then the controlled attribute is set to 100

Class Inheritance:

- `fr.esrf.TangoDs.Device_Impl`
 - `EnumeratedAttribute`

Properties:

Device Properties		
Property name	Property type	Description
EnumeratedLabelList	Array of string	The label of the created attributes
EnumeratedValueList	Tango::DEVVAR_DOUBLEARRAY	The enumerated values for corresponding labels defined in EnumeratedLabelList
AttributeName	string	The name of the controlled attribute
EnumeratedStateList	Array of string	the enumerated states for corresponding label defined in EnumeratedLabelList
EnumeratedAccuracyList	Tango::DEVVAR_DOUBLEARRAY	The list of the accuracy corresponding to the EnumeratedLabelList. Accuracy is expressed in percent.
StopCommandFullName	string	If a the attribute can be stop during the writing instruction.
SelectedAttributeName	string	The name of the attribute equals to true

Device Properties Default Values:

Property Name	Default Values
EnumeratedLabelList	No default value
EnumeratedValueList	No default value
AttributeName	No default value
EnumeratedStateList	No default value
EnumeratedAccuracyList	No default value
StopCommandFullName	No default value
SelectedAttributeName	No default value

There is no Class properties.

Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
version	DEV_STRING	READ	No
State	DEV_STRING	READ	No
selectedAttributeName	DEV_STRING	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
ActivateAttribute	DEV_STRING	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 *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 - ActivateAttribute

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

EnumeratedAttribute Class

Revision: 1.8 - Author: katyho

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 - ActivateAttribute

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

ESRF - Software Engineering Group