TANGO
Device
Server

Backlight Controller
User's Guide

Backlight Class

Revision: release_1_1_1 - Author: leclercq
Implemented in C++

Introduction:

A [Backlight] device acts as brightness control for objects observed by a video acquisition system. Its has an abstract valid range of [0..10]. This device must be linked to a MAO device (i.e. a SignleShotDIO device).

Properties:

Device Properties		
Property name	**Property type**	**Description**
MAODeviceName	Tango::DEV_STRING	Name of the underlying MAO device [MAO_6208 or compatible - no default value]
MAODeviceChannel	Tango::DEV_SHORT	The channel of MAO generic device actually connected to the backlight system [no default value]
MinBrightnessVoltage	Tango::DEV_DOUBLE	The voltage associated to brigtness level 0 [in Volts - defaults to 0]
MaxBrightnessVoltage	Tango::DEV_DOUBLE	The voltage associated to brigtness level 10 [in Volts - defaults to 10]
InitialBrightness	Tango::DEV_SHORT	Initial value (this value will be set at device startup [defaults to 0]

Device Properties Default Values:

Property Name	Default Values
MAODeviceName	No default value
MAODeviceChannel	No default value
MinBrightnessVoltage	No default value
MaxBrightnessVoltage	No default value
InitialBrightness	No default value

There is no Class properties.

States:

States	
Names	**Descriptions**
FAULT	The underlying MAO device is in FAULT state or unreachable
RUNNING	Device is up and ready to accept requests

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
brigthness: the brigthness value	DEV_SHORT	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
SetBrightness	DEV_USHORT	DEV_VOID
GetBrightness	DEV_VOID	DEV_USHORT
GetDependencies	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:**
  - Tango::FAULT
  - 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::FAULT
  - 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::FAULT
  - Tango::RUNNING

4 - SetBrightness

- **Description:**

- **Argin:**
  DEV_USHORT : Required brightness [0..10]

- **Argout:**
  DEV_VOID : N/A

- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING

5 - GetBrightness

- **Description:**

- **Argin:**
  DEV_VOID : N/A

- **Argout:**
  DEV_USHORT : Current brightness [0..10]

- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING

6 - GetDependencies

- **Description:** Returns the list of devices on which this device depends

- **Argin:**
  DEV_VOID : N/A

- **Argout:**
  DEVVAR_STRINGARRAY : The list of devices on which this device depends

- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING

ESRF - Software Engineering Group

TANGO
Device
Server

Backlight Controller
User’s Guide

Backlight Class

Revision: release_1_1_1 - Author: leclercq
Implemented in C++

Introduction:

A [Backlight] device acts as brightness control for objects observed by a video acquisition system. Its has an abstract valid range of [0..10]. This device must be linked to a MAO device (i.e. a SignleShotDIO device).

Properties:

Device Properties		
Property name	**Property type**	**Description**
MAODeviceName	Tango::DEV_STRING	Name of the underlying MAO device [MAO_6208 or compatible - no default value]
MAODeviceChannel	Tango::DEV_SHORT	The channel of MAO generic device actually connected to the backlight system [no default value]
MinBrightnessVoltage	Tango::DEV_DOUBLE	The voltage associated to brigtness level 0 [in Volts - defaults to 0]
MaxBrightnessVoltage	Tango::DEV_DOUBLE	The voltage associated to brigtness level 10 [in Volts - defaults to 10]
InitialBrightness	Tango::DEV_SHORT	Initial value (this value will be set at device startup [defaults to 0]

Device Properties Default Values:

Property Name	Default Values
MAODeviceName	No default value
MAODeviceChannel	No default value
MinBrightnessVoltage	No default value
MaxBrightnessVoltage	No default value
InitialBrightness	No default value

There is no Class properties.

States:

States	
Names	**Descriptions**
FAULT	The underlying MAO device is in FAULT state or unreachable
RUNNING	Device is up and ready to accept requests

Attributes:

Scalar Attributes			
Attribute name	**Data Type**	**R/W Type**	**Expert**
brigthness: the brigthness value	DEV_SHORT	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
SetBrightness	DEV_USHORT	DEV_VOID
GetBrightness	DEV_VOID	DEV_USHORT
GetDependencies	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:**
  - Tango::FAULT
  - 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::FAULT
  - 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::FAULT
  - Tango::RUNNING

4 - SetBrightness

- **Description:**

- **Argin:**
  DEV_USHORT : Required brightness [0..10]

- **Argout:**
  DEV_VOID : N/A

- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING

5 - GetBrightness

- **Description:**

- **Argin:**
  DEV_VOID : N/A

- **Argout:**
  DEV_USHORT : Current brightness [0..10]

- **Command allowed for:**
    - Tango::FAULT
    - Tango::RUNNING

6 - GetDependencies

- **Description:** Returns the list of devices on which this device depends

- **Argin:**
  DEV_VOID : N/A

- **Argout:**
  DEVVAR_STRINGARRAY : The list of devices on which this device depends

- **Command allowed for:**
    - Tango::FAULT
    - 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

Backlight Controller
Device Commands Description
Backlight Class

Revision: release_1_1_1 - Author: leclercq

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::FAULT
  - 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::FAULT
  - 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::FAULT
  - Tango::RUNNING

4 - SetBrightness

- **Description:**

- **Argin:**
  DEV_USHORT : Required brightness [0..10]

- **Argout:**
  DEV_VOID : N/A

- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING

5 - GetBrightness

- **Description:**

- **Argin:**
  DEV_VOID : N/A

- **Argout:**
  DEV_USHORT : Current brightness [0..10]

- **Command allowed for:**
  - Tango::FAULT

  - Tango::RUNNING

6 - GetDependencies

- **Description:** Returns the list of devices on which this device depends
- **Argin:**
  DEV_VOID : N/A
- **Argout:**
  DEVVAR_STRINGARRAY : The list of devices on which this device depends
- **Command allowed for:**
  - Tango::FAULT
  - Tango::RUNNING

ESRF - Software Engineering Group