



TANGO
Device
Server

Daresbury Tango Class User's Guide

Daresbury Class

Revision: release_1_0_0 - Author: le
Implemented in C++

Introduction:

A class to handle the Daresbury PSS interlock system (Initially , the hardware design has been done first by Daresbury Laboratory and redid by the ESRF later

Class Inheritance:

- Tango::Device_3Impl
 - Interlock
 - Daresbury

Properties:

Device Properties		
Property name	Property type	Description
Serial_line	Tango::DEV_STRING	Name of the serial line
Init_pscom	Tango::DEV_SHORT	Flag to force reinitialisation (This is needed on certain old daresbury board) Default is 0 => no initialisation

Device Properties Default Values:

Property Name	Default Values
Serial_line	No default value
Init_pscom	No default value

There is no Class properties.

Attributes:

Spectrum Attributes			
Attribute name	Data Type	X Data Length	Expert
Interlocks: Array of interlocks (16 interlock / short)	DEV_SHORT	1024	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
GetHistoryInfo	DEV_VOID	DEVVAR_LONGARRAY
ReadInterlockHistory	DEV_LONG	DEVVAR_SHORTARRAY
DevRead	DEV_VOID	DEV_STRING
Reset	DEV_VOID	DEV_VOID
GetInterlockDescription	DEVVAR_LONGARRAY	DEVVAR_STRINGARRAY
SetInterlockDescription	DEVVAR_STRINGARRAY	DEV_VOID
GetAllDescription	DEV_VOID	DEVVAR_STRINGARRAY
GetInterlockState	DEVVAR_LONGARRAY	DEVVAR_CHARARRAY

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 - GetHistoryInfo

- **Description:** Returns a list of known logs.

- **Argin:**
DEV_VOID :

- **Argout:**
DEVVAR_LONGARRAY : Return a list of interlock logs ,by date)

- **Command allowed for:**

5 - ReadInterlockHistory

- **Description:** Returns the states of the interlock set of the specified log

- **Argin:**
DEV_LONG : Log time , see GetHistoryInfo

- **Argout:**
DEVVAR_SHORTARRAY : Array of interlock

- **Command allowed for:**

6 - DevRead

- **Description:** Returns the response of the PSS hardware.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_STRING : Serial line reponse
- **Command allowed for:**

7 - Reset

- **Description:** Reset the interlock system. Must be implemented by the subclass.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

8 - GetInterlockDescription

- **Description:**
- **Argin:**
DEVVAR_LONGARRAY : List of Interlock addresses : [Word address,Bit address]*
- **Argout:**
DEVVAR_STRINGARRAY : Interlock description string array
- **Command allowed for:**

9 - SetInterlockDescription

- **Description:** Sets the description of the specified interlock. Note: The InterlockDefault class offers an implementation of this method and can be called directly from the subclass.

- **Argin:**
DEVVAR_STRINGARRAY : argin[0]=Word address argin[1]=Bit address argin[2]=Interlock description
- **Argout:**
DEV_VOID :
- **Command allowed for:**

10 - GetAllDescription

- **Description:**
- **Argin:**
DEV_VOID :
- **Argout:**
DEVVAR_STRINGARRAY : List of interlock description
- **Command allowed for:**

11 - GetInterlockState

- **Description:**
- **Argin:**
DEVVAR_LONGARRAY : List of Interlock addresses : [Word address,Bit address]*
- **Argout:**
DEVVAR_CHARARRAY : List of interlock state
- **Command allowed for:**

ESRF - Software Engineering Group



TANGO
Device
Server

Daresbury Tango Class User's Guide

Daresbury Class

Revision: release_1_0_0 - Author: le
Implemented in C++

Introduction:

A class to handle the Daresbury PSS interlock system (Initially , the hardware design has been done first by Daresbury Laboratory and redid by the ESRF later

Class Inheritance:

- Tango::Device_3Impl
 - Interlock
 - Daresbury

Properties:

Device Properties		
Property name	Property type	Description
Serial_line	Tango::DEV_STRING	Name of the serial line
Init_pscom	Tango::DEV_SHORT	Flag to force reinitialisation (This is needed on certain old daresbury board) Default is 0 => no initialisation

Device Properties Default Values:

Property Name	Default Values
Serial_line	No default value
Init_pscom	No default value

There is no Class properties.

Attributes:

Spectrum Attributes			
Attribute name	Data Type	X Data Length	Expert
Interlocks: Array of interlocks (16 interlock / short)	DEV_SHORT	1024	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
GetHistoryInfo	DEV_VOID	DEVVAR_LONGARRAY
ReadInterlockHistory	DEV_LONG	DEVVAR_SHORTARRAY
DevRead	DEV_VOID	DEV_STRING
Reset	DEV_VOID	DEV_VOID
GetInterlockDescription	DEVVAR_LONGARRAY	DEVVAR_STRINGARRAY
SetInterlockDescription	DEVVAR_STRINGARRAY	DEV_VOID
GetAllDescription	DEV_VOID	DEVVAR_STRINGARRAY
GetInterlockState	DEVVAR_LONGARRAY	DEVVAR_CHARARRAY

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 - GetHistoryInfo

- **Description:** Returns a list of known logs.

- **Argin:**
DEV_VOID :

- **Argout:**
DEVVAR_LONGARRAY : Return a list of interlock logs ,by date)

- **Command allowed for:**

5 - ReadInterlockHistory

- **Description:** Returns the states of the interlock set of the specified log

- **Argin:**
DEV_LONG : Log time , see GetHistoryInfo

- **Argout:**
DEVVAR_SHORTARRAY : Array of interlock

- **Command allowed for:**

6 - DevRead

- **Description:** Returns the response of the PSS hardware.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_STRING : Serial line reponse
- **Command allowed for:**

7 - Reset

- **Description:** Reset the interlock system. Must be implemented by the subclass.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

8 - GetInterlockDescription

- **Description:**
- **Argin:**
DEVVAR_LONGARRAY : List of Interlock addresses : [Word address,Bit address]*
- **Argout:**
DEVVAR_STRINGARRAY : Interlock description string array
- **Command allowed for:**

9 - SetInterlockDescription

- **Description:** Sets the description of the specified interlock. Note: The InterlockDefault class offers an implementation of this method and can be called directly from the subclass.

- **Argin:**
DEVVAR_STRINGARRAY : argin[0]=Word address argin[1]=Bit address argin[2]=Interlock description
- **Argout:**
DEV_VOID :
- **Command allowed for:**

10 - GetAllDescription

- **Description:**
- **Argin:**
DEV_VOID :
- **Argout:**
DEVVAR_STRINGARRAY : List of interlock description
- **Command allowed for:**

11 - GetInterlockState

- **Description:**
- **Argin:**
DEVVAR_LONGARRAY : List of Interlock addresses : [Word address,Bit address]*
- **Argout:**
DEVVAR_CHARARRAY : List of interlock state
- **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

Daresbury Tango Class Device Commands Description Daresbury Class

Revision: release_1_0_0 - Author: le

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 - GetHistoryInfo

- **Description:** Returns a list of known logs.
- **Argin:**
DEV_VOID :
- **Argout:**
DEVVAR_LONGARRAY : Return a list of interlock logs ,by date)
- **Command allowed for:**

5 - ReadInterlockHistory

- **Description:** Returns the states of the interlock set of the specified log
- **Argin:**
DEV_LONG : Log time , see GetHistoryInfo
- **Argout:**
DEVVAR_SHORTARRAY : Array of interlock
- **Command allowed for:**

6 - DevRead

- **Description:** Returns the response of the PSS hardware.
- **Argin:**
DEV_VOID :
- **Argout:**

DEV_STRING : Serial line reponse

- **Command allowed for:**

7 - Reset

- **Description:** Reset the interlock system. Must be implemented by the subclass.
- **Argin:**
DEV_VOID :
- **Argout:**
DEV_VOID :
- **Command allowed for:**

8 - GetInterlockDescription

- **Description:**
- **Argin:**
DEVVAR_LONGARRAY : List of Interlock addresses : [Word address,Bit address]*
- **Argout:**
DEVVAR_STRINGARRAY : Interlock description string array
- **Command allowed for:**

9 - SetInterlockDescription

- **Description:** Sets the descripton of the specified interlock. Note: The InterlockDefault class offers an implementation of this method and can be called directly from the subclass.
- **Argin:**
DEVVAR_STRINGARRAY : argin[0]=Word address argin[1]=Bit address argin[2]=Interlock description
- **Argout:**
DEV_VOID :
- **Command allowed for:**

10 - GetAllDescription

- **Description:**
- **Argin:**
DEV_VOID :
- **Argout:**
DEVVAR_STRINGARRAY : List of interlock description
- **Command allowed for:**

11 - GetInterlockState

- **Description:**
- **Argin:**
DEVVAR_LONGARRAY : List of Interlock addresses : [Word address,Bit address]*
- **Argout:**
DEVVAR_CHARARRAY : List of interlock state
- **Command allowed for:**

ESRF - Software Engineering Group