



**TANGO  
Device  
Server**

# **Synrad Laser Control (type CO<sub>2</sub>) User's Guide**

## **Synrad Class**

**Revision: release\_1\_0\_1 - Author: le  
Implemented in C++**

### **Introduction:**

Class to control a Synrad Laser in two modes : - Instantaneous : the laser deliver the specified power - rampe : the power specified is reached in a fixed time.

### **Class Inheritance:**

- Tango::Device\_3Impl
  - Synrad

## Properties:

Device Properties		
Property name	Property type	Description
<b>SSAOProxy</b>	Tango::DEV_STRING	Proxy on the CPCI 6208 board
<b>AttributeSeqWProxy</b>	Tango::DEV_STRING	Proxy on the AttributeSequenceWriter Device to execute the ramp.
<b>LaserPowerType</b>	Tango::DEV_DOUBLE	The power max of the laser : - 100 for 100W or - 240 for 240 Watts

### Device Properties Default Values:

Property Name	Default Values
SSAOProxy	No default value
AttributeSeqWProxy	No default value
LaserPowerType	No default value

**There is no Class properties.**

## Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
<b>risingTime</b> : Time to reach the specified power	DEV_DOUBLE	READ_WRITE	No
<b>powerValueToReach</b> : Power value to reach	DEV_DOUBLE	READ_WRITE	No
<b>actualMode</b> : Mode of the device : - INSTANAEIOUS - RAMPE	DEV_STRING	READ	No
<b>chNum</b> : The channel number used on the 6608 board.	DEV_DOUBLE	READ_WRITE	No
<b>nbPoints</b> : Number of points to generate the rampe. The ideal number is comprised between 100 and 500 points.	DEV_DOUBLE	READ_WRITE	No

## Commands:

More Details on commands....

## Device Commands for Operator Level

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>InstantaneousMode</b>	DEV_VOID	DEV_VOID
<b>RisingMode</b>	DEV_VOID	DEV_VOID
<b>PowerOff</b>	DEV_VOID	DEV_VOID
<b>StartRamp</b>	DEV_VOID	DEV_VOID
<b>StopRamp</b>	DEV_VOID	DEV_VOID
<b>PauseRamp</b>	DEV_VOID	DEV_VOID
<b>ResumeRamp</b>	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:**

### 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 - InstantaneousMode

- **Description:** Command to switch in the instantaneous mode
- **Argin:**  
**DEV\_VOID** : no argin
- **Argout:**  
**DEV\_VOID** : no argout
- **Command allowed for:**

### 5 - RisingMode

- **Description:** Command to switch in the [rampe] mode.
- **Argin:**  
**DEV\_VOID** : no argin
- **Argout:**  
**DEV\_VOID** : no argout
- **Command allowed for:**

### 6 - PowerOff

- **Description:** Command to stop the power
- **Argin:**  
**DEV\_VOID** : no argin
- **Argout:**  
**DEV\_VOID** : no argout
- **Command allowed for:**

## 7 - StartRamp

- **Description:** Start the sequence generation.
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**

## 8 - StopRamp

- **Description:** Stop the sequence generation.
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**

## 9 - PauseRamp

- **Description:** Pause the sequence generation.
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**

## 10 - ResumeRamp

- **Description:** Restart the sequence generation previously paused (with Pause command).
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**

---

**ESRF - Software Engineering Group**



**TANGO**  
Device  
Server

# **Synrad Laser Control (type CO<sub>2</sub>) User's Guide**

## **Synrad Class**

**Revision: release\_1\_0\_1 - Author: le  
Implemented in C++**

### **Introduction:**

Class to control a Synrad Laser in two modes : - Instantaneous : the laser deliver the specified power - rampe : the power specified is reached in a fixed time.

### **Class Inheritance:**

- Tango::Device\_3Impl
  - Synrad

## Properties:

Device Properties		
Property name	Property type	Description
<b>SSAOProxy</b>	Tango::DEV_STRING	Proxy on the CPCI 6208 board
<b>AttributeSeqWProxy</b>	Tango::DEV_STRING	Proxy on the AttributeSequenceWriter Device to execute the ramp.
<b>LaserPowerType</b>	Tango::DEV_DOUBLE	The power max of the laser : - 100 for 100W or - 240 for 240 Watts

### Device Properties Default Values:

Property Name	Default Values
SSAOProxy	No default value
AttributeSeqWProxy	No default value
LaserPowerType	No default value

**There is no Class properties.**

## Attributes:

Scalar Attributes			
Attribute name	Data Type	R/W Type	Expert
<b>risingTime</b> : Time to reach the specified power	DEV_DOUBLE	READ_WRITE	No
<b>powerValueToReach</b> : Power value to reach	DEV_DOUBLE	READ_WRITE	No
<b>actualMode</b> : Mode of the device : - INSTANAEIOUS - RAMPE	DEV_STRING	READ	No
<b>chNum</b> : The channel number used on the 6608 board.	DEV_DOUBLE	READ_WRITE	No
<b>nbPoints</b> : Number of points to generate the rampe. The ideal number is comprised between 100 and 500 points.	DEV_DOUBLE	READ_WRITE	No

## Commands:

More Details on commands....



## Device Commands for Operator Level

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>InstantaneousMode</b>	DEV_VOID	DEV_VOID
<b>RisingMode</b>	DEV_VOID	DEV_VOID
<b>PowerOff</b>	DEV_VOID	DEV_VOID
<b>StartRamp</b>	DEV_VOID	DEV_VOID
<b>StopRamp</b>	DEV_VOID	DEV_VOID
<b>PauseRamp</b>	DEV_VOID	DEV_VOID
<b>ResumeRamp</b>	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:**

### 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 - InstantaneousMode

- **Description:** Command to switch in the instantaneous mode
- **Argin:**  
**DEV\_VOID** : no argin
- **Argout:**  
**DEV\_VOID** : no argout
- **Command allowed for:**

### 5 - RisingMode

- **Description:** Command to switch in the [rampe] mode.
- **Argin:**  
**DEV\_VOID** : no argin
- **Argout:**  
**DEV\_VOID** : no argout
- **Command allowed for:**

### 6 - PowerOff

- **Description:** Command to stop the power
- **Argin:**  
**DEV\_VOID** : no argin
- **Argout:**  
**DEV\_VOID** : no argout
- **Command allowed for:**

## 7 - StartRamp

- **Description:** Start the sequence generation.
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**

## 8 - StopRamp

- **Description:** Stop the sequence generation.
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**

## 9 - PauseRamp

- **Description:** Pause the sequence generation.
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**

## 10 - ResumeRamp

- **Description:** Restart the sequence generation previously paused (with Pause command).
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **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

---

# Synrad Laser Control (type CO<sub>2</sub>) Device Commands Description Synrad Class

Revision: release\_1\_0\_1 - 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 destructor 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 - InstantaneousMode

- **Description:** Command to switch in the instantaneous mode
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**

### 5 - RisingMode

- **Description:** Command to switch in the [rampe] mode.
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**  
DEV\_VOID : no argout
- **Command allowed for:**

### 6 - PowerOff

- **Description:** Command to stop the power
- **Argin:**  
DEV\_VOID : no argin
- **Argout:**

**DEV\_VOID** : no argout

- **Command allowed for:**

## 7 - StartRamp

- **Description:** Start the sequence generation.
- **Argin:**  
**DEV\_VOID** : no argin
- **Argout:**  
**DEV\_VOID** : no argout
- **Command allowed for:**

## 8 - StopRamp

- **Description:** Stop the sequence generation.
- **Argin:**  
**DEV\_VOID** : no argin
- **Argout:**  
**DEV\_VOID** : no argout
- **Command allowed for:**

## 9 - PauseRamp

- **Description:** Pause the sequence generation.
- **Argin:**  
**DEV\_VOID** : no argin
- **Argout:**  
**DEV\_VOID** : no argout
- **Command allowed for:**



## 10 - ResumeRamp

- **Description:** Restart the sequence generation previously paused (with Pause command).
  - **Argin:**  
**DEV\_VOID** : no argin
  - **Argout:**  
**DEV\_VOID** : no argout
  - **Command allowed for:**
- 

**ESRF - Software Engineering Group**