TANGO
Device
Server

Scienta Acquisition
User’s Guide

ScientaAcquisition Class

Revision: release_1_0_7 - Author: jcpret
Implemented in C++ - CVS repository: tango-ds

Introduction:

Perform acquisition on Scienta instruments.

Class Description:

To make the device work, there are some requisites:

- The user must create an environment variable named **SES_ROOT**, which specify the path to the SES folder (e.g. C:\DeviceServers). This folder will have to contain the following things (gave with SES program):
    - The folder *dll* (contains all dll for the device)
    - The folder *data* (contains .dat files)
    - The folder *ini* (contains .ini files)
    - The folder *work*
    - The folder *factory*
    - The folder *U50*

  - The program *Ses.exe*
  - Our device server *ds_ScientaAcquisition.exe*
- Then the folder *data* must contains 2 files named **Instrument.dat** and **RunVar.dat**.
- The user can then makes some configurations using *Ses.exe*. They will be take into account in the device.
- Only one program can be opened at the same time (*Ses.exe* or *ds_ScientaAcquisition.exe*).
- The device works with version **1.2.2** of SesInstrument.dll.
- The driver creates a file named *ScientaTemp.dat* for acquisition.

Tango Device Server User's Guide

Properties:

Device Properties		
Property name	**Property type**	**Description**
SaveToFile	Tango::DEV_SHORT	1 if result of each acquisition must be saved to file, 0 otherwise.
FilePath	Tango::DEV_STRING	The path of the file saved by acquisition.

Device Properties Default Values:

Property Name	**Default Values**
SaveToFile	No default value
FilePath	No default value

There is no Class properties.

States:

States	
Names	**Descriptions**
RUNNING	An acquisition is currently running.
UNKNOWN	Unknown state of the equipement.
FAULT	The previous acquisition has failed.
STANDBY	Everything is OK.

Attributes:

Scalar Attributes

Attribute name	Data Type	R/W Type	Expert
excitationEnergy: The excitation or photon energy of the region	DEV_DOUBLE	READ_WRITE	No
mode: Whether ADC or discriminator should be used for acquisition (boolean : 0 or 1). This value is only used if the detector system has both ADC and discriminator	DEV_STRING	READ_WRITE	No
energyScale: The energy scale, Kinetic or Binding	DEV_STRING	READ_WRITE	No
highEnergy: The stopping energy for swept mode acquisitions	DEV_DOUBLE	READ_WRITE	No
lowEnergy: The starting energy for swept mode acquisitions	DEV_DOUBLE	READ_WRITE	No
fixEnergy: The energy at the centre of the active region of the detector for fixed mode acquisitions	DEV_DOUBLE	READ_WRITE	No
energyStep: The energy step for swept mode acquisitions	DEV_DOUBLE	READ_WRITE	No
stepTime: The step or dwell time in milliseconds for the region	DEV_DOUBLE	READ_WRITE	No
lensMode: The lens mode. The possible values depend on the instrument (ex: Transmission, Angular1, Angular2, Angular3)	DEV_STRING	READ_WRITE	No
passEnergy: The pass energy of the region. The possible values depend on the instrument (ex: 1,2,5,10,20,50,100,200,500)	DEV_SHORT	READ_WRITE	No
passMode: Low Pass or High Pass.	DEV_STRING	READ_WRITE	No
detectorFirstXChannel: The first active channel in the energy direction of the detector	DEV_SHORT	READ_WRITE	No
detectorLastXChannel: The last active channel in the energy direction of the detector	DEV_SHORT	READ_WRITE	No
detectorFirstYChannel: The first active channel in the angular or spatial direction of the detector	DEV_SHORT	READ_WRITE	No
detectorLastYChannel: The last active channel in the angular or spatial direction of the detector	DEV_SHORT	READ_WRITE	No
detectorSlices: The number of spectrum channels or slices in the angular or spatial direction of the detector	DEV_SHORT	READ_WRITE	No
ADCMode: Whether ADC or discriminator should be used for acquisition (boolean : 0 or 1). This value is only used if the detector system has both ADC and discriminator	DEV_SHORT	READ_WRITE	Yes
ADCMask: The grayscale bitmask of the detector. Valid bitmasks are 0xFF, 0xFE, 0xFC, 0xF8, 0xF0, 0xE0, 0xC0 and 0x80	DEV_SHORT	READ_WRITE	Yes
discriminatorLevel: The discriminator level of the detector (0 to 255)	DEV_SHORT	READ_WRITE	Yes
channels: The number of channels acquired in the energy direction of the spectrum	DEV_LONG	READ	No
slices: Channels acquired in the angular or spatial direction of the spectrum	DEV_LONG	READ	Yes
sweeps: The number of times the spectrum has been acquired	DEV_LONG	READ	Yes

Spectrum Attributes			
Attribute name	**Data Type**	**X Data Length**	**Expert**
channelScale: Channels elements containing the energy scale of the spectrum	DEV_DOUBLE	1000000	No
sliceScale: Slices elements containing the angular or spatial scale of the spectrum	DEV_DOUBLE	1000000	No
sumData: Channels elements containing the sum over all slices over the acquired data of the spectrum	DEV_DOUBLE	1000000	No

Image Attributes				
Attribute name	**Data Type**	**X Data Length**	**Y Data Length**	**Expert**
data: Matrix with Slices times Channels elements containing the acquired data. This matrix is a table of spectrum	DEV_DOUBLE	10000	10000	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
Start	DEV_VOID	DEV_VOID
Stop	DEV_VOID	DEV_VOID
ResetInstrument	DEV_VOID	DEV_VOID
ResetSupplies	DEV_VOID	DEV_VOID
TestCommunication	DEV_VOID	DEV_VOID
GetHardwareInfo	DEV_VOID	DEV_STRING
GetLensModeList	DEV_VOID	DEVVAR_STRINGARRAY
GetPassEnergyList	DEV_VOID	DEVVAR_SHORTARRAY
GetPassModeList	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::RUNNING
  - Tango::UNKNOWN
  - Tango::FAULT
  - Tango::STANDBY

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::RUNNING
  - Tango::UNKNOWN
  - Tango::FAULT
  - Tango::STANDBY

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::RUNNING
  - Tango::UNKNOWN
  - Tango::FAULT
  - Tango::STANDBY

4 - Start

- **Description:** Start the acqusition for one region with parameters configured in the attributes.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

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

5 - Stop

- **Description:** Stop the current acquisition.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

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

6 - ResetInstrument

- **Description:** Reset instrument.
- **Argin:**
  DEV_VOID :
- **Argout:**
  DEV_VOID :
- **Command allowed for:**
  - Tango::RUNNING
  - Tango::UNKNOWN
  - Tango::FAULT
  - Tango::STANDBY

7 - ResetSupplies

- **Description:** Reset supplies.
- **Argin:**
  DEV_VOID :
- **Argout:**
  DEV_VOID :
- **Command allowed for:**
  - Tango::RUNNING
  - Tango::UNKNOWN
  - Tango::FAULT
  - Tango::STANDBY

8 - TestCommunication

- **Description:** Test communication with hardware.
- **Argin:**
  DEV_VOID :
- **Argout:**
  DEV_VOID :
- **Command allowed for:**
  - Tango::RUNNING
  - Tango::UNKNOWN
  - Tango::FAULT

    - Tango::STANDBY

9 - GetHardwareInfo

- **Description:** Retrieve hardware informations.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_STRING : A decription of the hardware

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

10 - GetLensModeList

- **Description:**

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEVVAR_STRINGARRAY : The list of possible lens modes

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

11 - GetPassEnergyList

- **Description:**

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEVVAR_SHORTARRAY : The list of possible pass energies

- **Command allowed for:**

    - Tango::RUNNING
    - Tango::UNKNOWN
    - Tango::FAULT
    - Tango::STANDBY

12 - GetPassModeList

- **Description:**

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEVVAR_STRINGARRAY : The possible values for Pass Modes

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