scanLib Documentation

Release 0.1

Argonne National Laboratory

Jun 10, 2022

CONTENTS

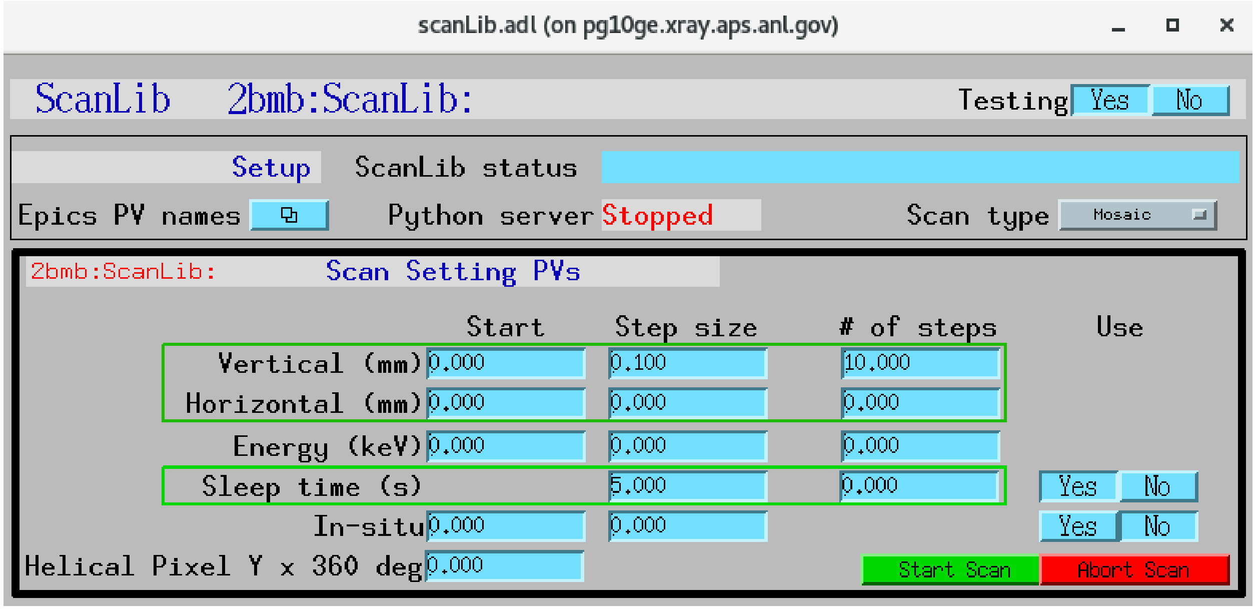
CHAPTER

ONE

CONTENT

1.1 About

This is a fully functional epics IOC that provide an example on how to create&serve epics PVs, how to connect to PV served by other epics IOCs (both referring to them by PV prefix or by full PV name, both customizable at run time) and how to support a callback via python.



scanLib.adl (on pg10ge.xray.aps.anl.gov)

ScanLib 2bmb:ScanLib:

Testing: Yes / No

Setup

ScanLib status

Epics PV names

Python server Stopped

Scan type Mosaic

2bmb:ScanLib: Scan Setting PVs

	Start	Step size	# of steps	Use
Vertical (mm)	0.000	0.100	10.000	
Horizontal (mm)	0.000	0.000	0.000	
Energy (keV)	0.000	0.000	0.000	
Sleep time (s)		5.000	0.000	Yes / No
In-situ	0.000	0.000		Yes / No
Helical Pixel Y x 360 deg	0.000			

1.2 Install directions

1.2.1 Build EPICS base

Warning: Make sure the disk partition hosting ~/epics is not larger than 2 TB. See tech talk and Diamond Data Storage document.

```
$ mkdir ~/epics
$ cd epics
```

- Download EPICS base latest release, i.e. 7.0.3.1., from https://github.com/epics-base/epics-base:

  ```
  $ git clone https://github.com/epics-base/epics-base.git
  $ cd epics-base
  $ make -sj
  ```

1.2.2 Build a minimal synApps

To build a minimal synApp:

```
$ cd ~/epics
```

- Download in ~/epics assemble_synApps.sh
- **Edit the assemble_synApps.sh script as follows:**
  1. Set FULL_CLONE=True
  2. Set EPICS_BASE to point to the location of EPICS base. This could be on APSshare (the default), or a local version you built.

  For scanlib you need

  1. ASYN=R4-37
  2. AUTOSAVE=R5-10
  3. BUSY=R1-7-2
  4. XXX=R6-1

  You can comment out all of the other modules (ALLENBRADLEY, ALIVE, etc.)
- Run:

  ```
  $ assemble_synApps.sh
  ```

- This will create a synApps/support directory:

  ```
  $ cd synApps/support/
  ```

- Edit asyn-RX-YY/configure/RELEASE to comment out the lines starting with:

  ```
  IPAC=$(SUPPORT)/
  SNCSEQ=$(SUPPORT)/
  ```

> **Warning:** If building for RedHat8 uncomment **TIRPC=YES** in asyn-RX-YY/configure/CONFIG_SITE

- Clone the scanlib module into synApps/support:

  ```
  $ git clone https://github.com/tomography/scanlib.git
  ```

- Edit configure/RELEASE add this line to the end:

  ```
  SCANLIB=$(SUPPORT)/scanlib
  ```

- Edit Makefile add this line to the end of the MODULE_LIST:

```
MODULE_LIST += SCANLIB
```

- Run the following commands:

```
$ make release
$ make -sj
```

1.2.3 Build the python server

To build the **scanLib** python server you need to have Conda installed.

Next, create a dedicated conda environment for scanLib by running:

```
(base) $ conda create --name scanlib python=3.9
```

then:

```
(base) $ conda activate scanlib
```

and install the required python packages:

```
(scanlib) $ pip install pvapy
(scanlib) $ pip install pyepics
```

Finally you can build **scanLib** with:

```
(scanlib) $ cd ~/epics/synApps/support/scanLib/
(scanlib) $ python setup.py install
```

To run the python server:

```
(scanlib) $ python -i start_scanlib.py
```

1.3 Usage

scanLib is a simple EPICS IOC to use as a template for creating EPICS PVs. **scanLib** also provides callback support examples using python.

1.3.1 Start EPICS IOC

Edit ~/epics/synApps/support/scanlib/configure to set EPICS_BASE to point to the location of EPICS base, i.e.:

```
EPICS_BASE=/APSshare/epics/base-3.15.6
```

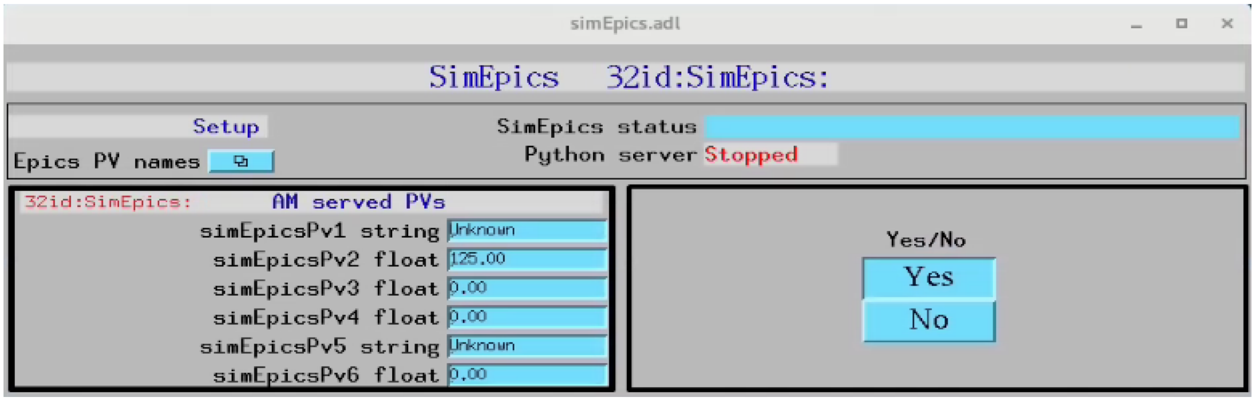
then start the **scanLib** epics ioc with:

```
$ cd ~/epics/synApps/support/scanlib/iocBoot/iocScanLib
$ start_IOC
```

1.3.2 Start MEDM screen

```
$ cd ~/epics/synApps/support/scanlib/iocBoot/iocScanLib
$ start_medm
```

scanLib control screen with no python server running:



simEpics.adl

SimEpics 32id:SimEpics:

SimEpics status

Python server Stopped

32id:SimEpics: AM served PVs

simEpicsPv1 string Unknown
simEpicsPv2 float 125.00
simEpicsPv3 float 0.00
simEpicsPv4 float 0.00
simEpicsPv5 string Unknown
simEpicsPv6 float 0.00

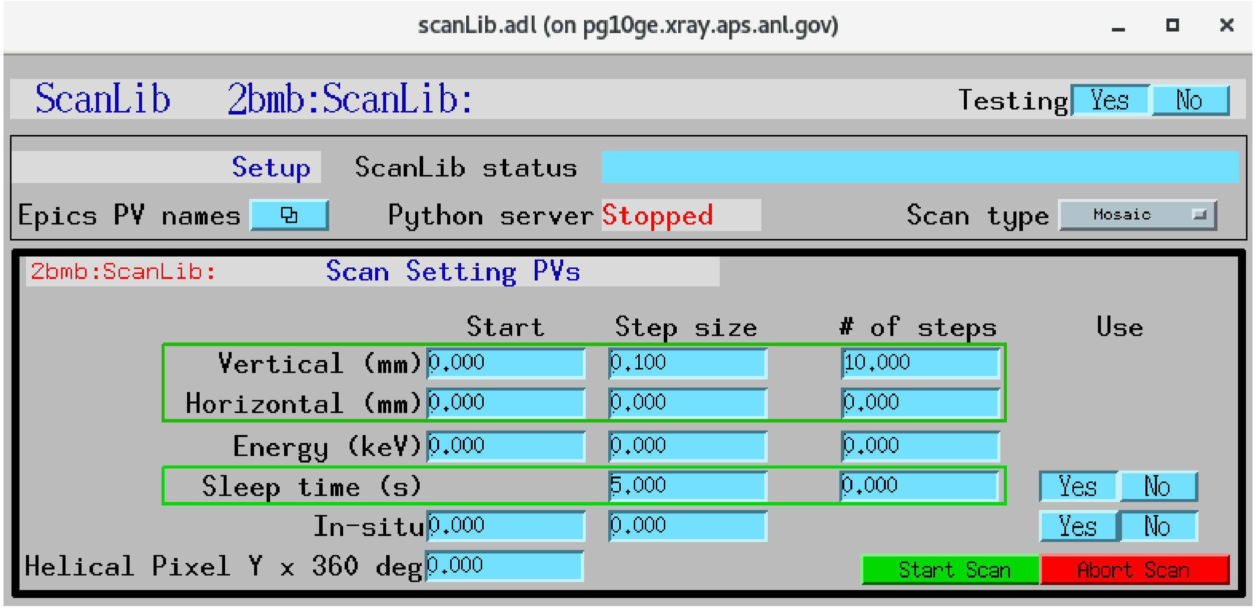
1.3.3 Start python server

```
$ bash
(base) $ conda activate scanlib
(scanlib) $ cd ~/epics/synApps/support/scanlib/iocBoot/iocScanLib
(scanlib) $ python -i start_scanlib.py
    configPVS:
    TomoscanPVPrefix : 2bmb:TomoScan:
    InsituPVName : 32id:m1
    SleepSelect : Yes
    scanLibPv1 : Unknown
    scanLibPv2 : 500.00
    scanLibPv3 : 0
    scanLibPv4 : 0
    scanLibPv5 : Unknown
    scanLibPv6 : 0

    controlPVS:
    Example : None
    ScanLibStatus : divide by 2
    Watchdog : -2265

    pv_prefixes:
    Camera : 2bmbSP2:
    >>>
```

scanLib control screen with python server running:



Python server Stopped

Scan type Mosaic

2bmb:ScanLib: Scan Setting PVs

	Start	Step size	# of steps	Use
Vertical (mm)	0.000	0.100	10.000	
Horizontal (mm)	0.000	0.000	0.000	
Energy (keV)	0.000	0.000	0.000	
Sleep time (s)		5.000	0.000	Yes No
In-situ	0.000	0.000		Yes No
Helical Pixel Y x 360 deg	0.000			

1.3.4 Testing

...

1.4 scanLibApp EPICS application

scanLib is an EPICS application, providing scan support to tomoScan. scanLib consisting of:

- A database file and corresponding autosave request file that contain the PVs required by the scanlib.py base class.
- OPI screens for medm
- An example IOC application that can be used to run the above databases. The databases are loaded in the IOC with the example substitutions file, scanLib.substitutions.

1.4.1 Base class files

The following tables list all of the records in the scanLib.template file. These records are used by the scanlib base class and so are required.

scanLib.template

This is the database file that contains only the PVs required by the scanlib.py base class scanLib.template.

PV Prefixes

Record name	Record type	Description
$(P)$(R)TomoScanPVPrefix	stringout	Contains the prefix for the tomoScan, e.g. 2bmb:TomoScan:

PV names

Record name	Record type	Description
$(P)$(R)InsituPVName	stringout	Contains the PV name controlling the in-situ enviroment parameter, e.g. 32id:m1

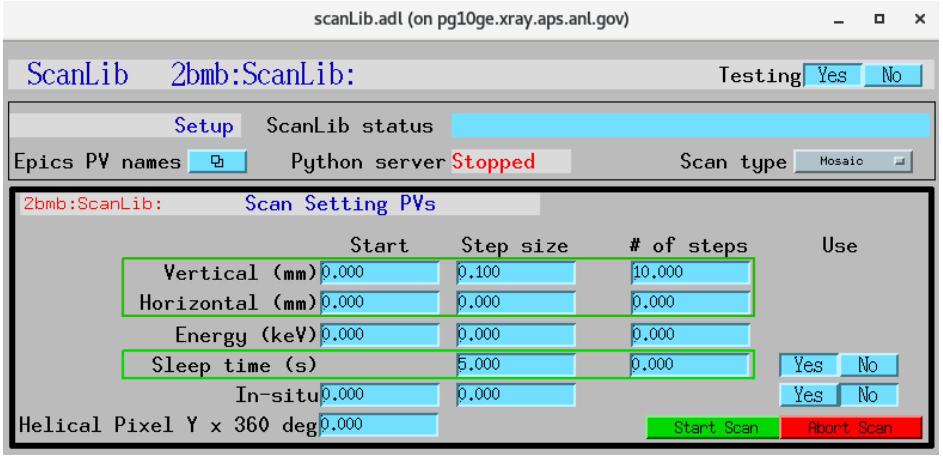
ScanLib served PVs

Record name	Record type	Description
$(P)$(R)VerticalStart	stringout	Contains a string PV.
$(P)$(R)VerticalStepSize	ao	Contains a float PV.
$(P)$(R)VerticalSteps	ao	Contains a float PV.
$(P)$(R)HorizontalStart	stringout	Contains a string PV.
$(P)$(R)HorizontalStepSize	ao	Contains a float PV.
$(P)$(R)HorizontalSteps	ao	Contains a float PV.
$(P)$(R)InsituStart	stringout	Contains a string PV.
$(P)$(R)InsituStepSize	ao	Contains a float PV.
$(P)$(R)InsituSteps	ao	Contains a float PV.
$(P)$(R)EnergyStart	stringout	Contains a string PV.
$(P)$(R)EnergyStepSize	ao	Contains a float PV.
$(P)$(R)EnergySteps	ao	Contains a float PV.
$(P)$(R)PixelsYPer360Deg	ao	Contains a float PV.
$(P)$(R)SleepTime	ao	Contains a float PV.
$(P)$(R)YesNoSelect	mbbo	Contains a float PV.

medm files

scanLib.adl

The following is the MEDM screen `scanLib.adl` during a scan. The status information is updating.



scanLib.adl (on pg10ge.xray.aps.anl.gov)

ScanLib 2bmb:ScanLib: Testing Yes No

Setup ScanLib status

Epics PV names Python server Stopped Scan type Mosaic

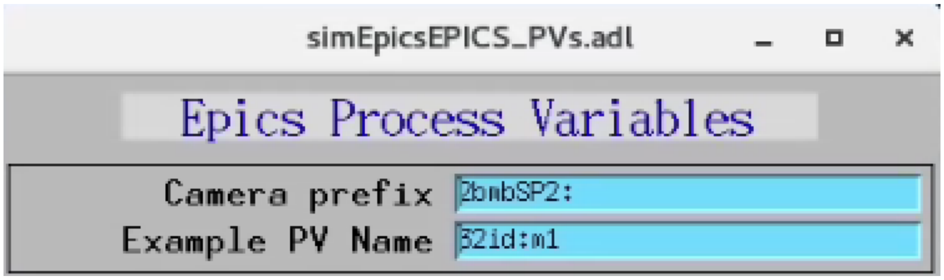
2bmb:ScanLib: Scan Setting PVs

	Start	Step size	# of steps	Use
Vertical (mm)	0.000	0.100	10.000	
Horizontal (mm)	0.000	0.000	0.000	
Energy (keV)	0.000	0.000	0.000	
Sleep time (s)		5.000	0.000	Yes No
In-situ	0.000	0.000		Yes No
Helical Pixel Y x 360 deg	0.000			

scanLibEPICS_PVs.adl

The following is the MEDM screen `scanLibEPICS_PVs.adl`.

If these PVs are changed scanLib must be restarted.



simEpicsEPICS_PVs.adl

Epics Process Variables

Camera prefix 2bmbSP2:

Example PV Name 32id:m1