



Blissdata integration in Sardana

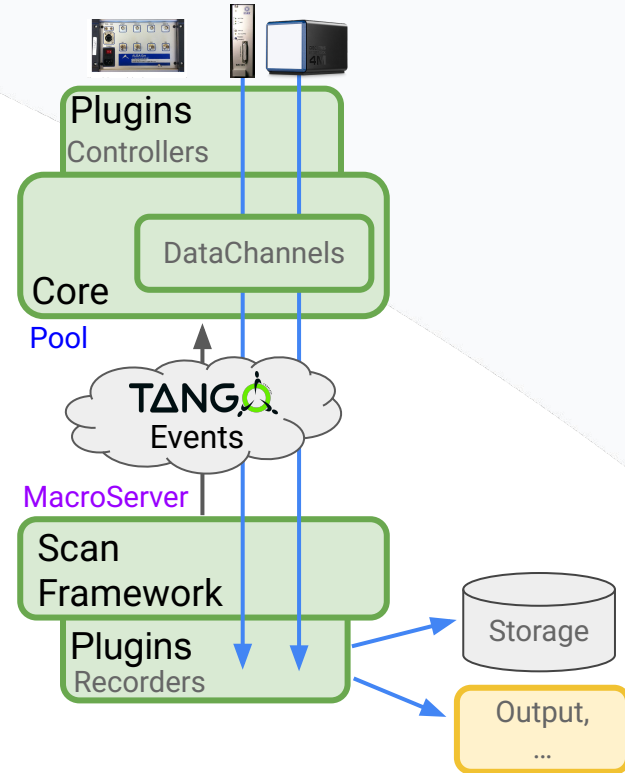
Roberto Homs
Jordi Aguilar
Oriol Vallcorba
Zbigniew Reszela

Sardana Workshop 2025
27–29 Aug 2025
MAX IV Laboratory

Continuous Scans: Improve support for multi-technique experiments

Roadmap defined in Sardana CS Workshop 2023

New tools for **data composition**



Continuous Scans: Improve support for multi-technique experiments

Roadmap defined in Sardana CS Workshop 2023

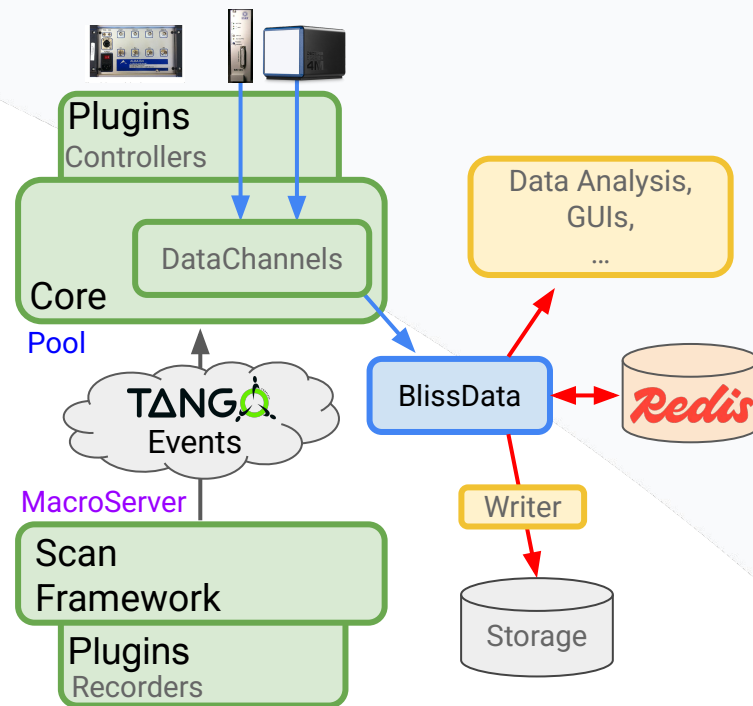
New tools for **data composition**



Integrate ESRF **Blissdata** in Sardana

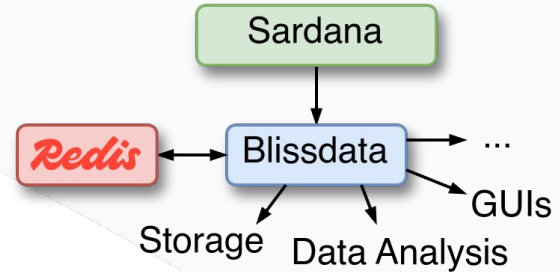
Real-time data available for **external consumers** for **data composition**, plotting, storage, analysis, or any other operation.

Minimize the stress on the MacroServer and avoid potential bottlenecks in Tango events

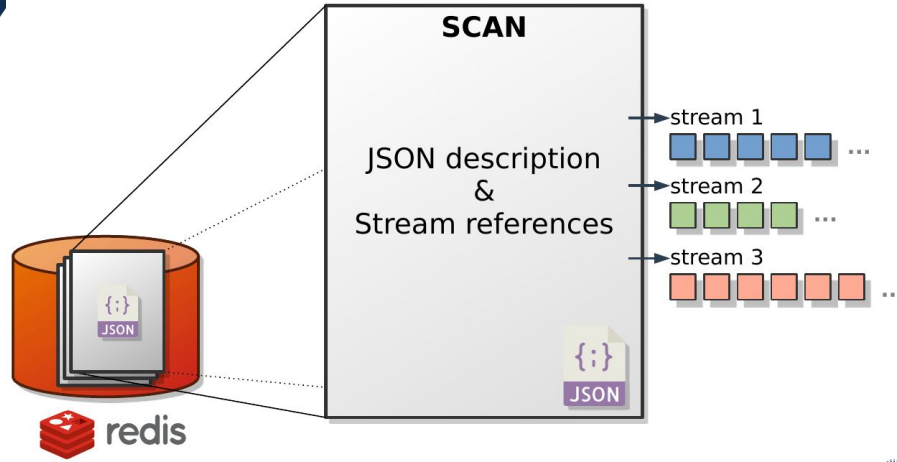


Blissdata

- Unified **API** for Online/Offline data access
- **Streaming engine** for multiple services
- It mostly relies on **Redis**, but can serve data from **other sources** through a single API
- **Independent** Python package
- ReJSON to enable use of JSON inside keys and RediSearch for optimized JSON queries
- Sinks for optimized DB access
- Available **clients**:
 - Nexus Writer
 - Flint: GUI for online plotting
 - Data analysis (EWOKS)



- Remote dictionary server
- In-memory data store (with on-disk persistence capabilities)
- Key-value pairs
- Support for complex data types
- Server extensions (e.g. reJSON)
- Streaming engine. Message broker
- ~~Open Source~~ Source Available

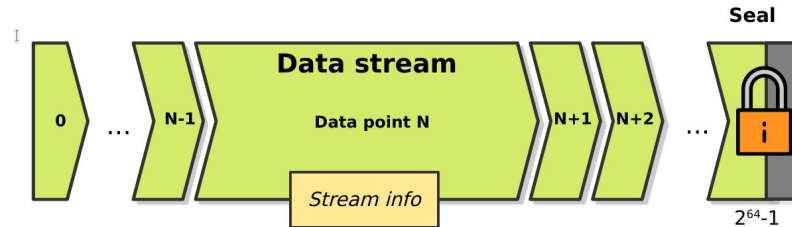


Everything built around **Scan and Stream**

They are identified in Redis by keys like:

`esrf:scan:01H25V2VFF3AA8WWJVNSWS4ECT`

`esrf:stream:01H25V2VFTPPKG79DA93134NKN`

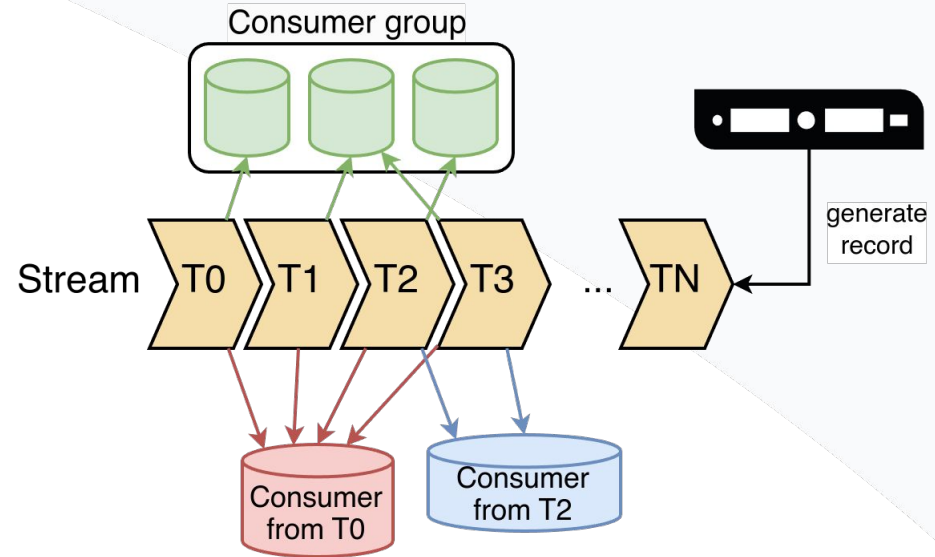


- Series accessed randomly or sequentially
- Scalars, arrays or JSON
- Encoding: Numerical, JSON
- Plugin system for custom Stream types (blissdata>=2)



Redis Streams

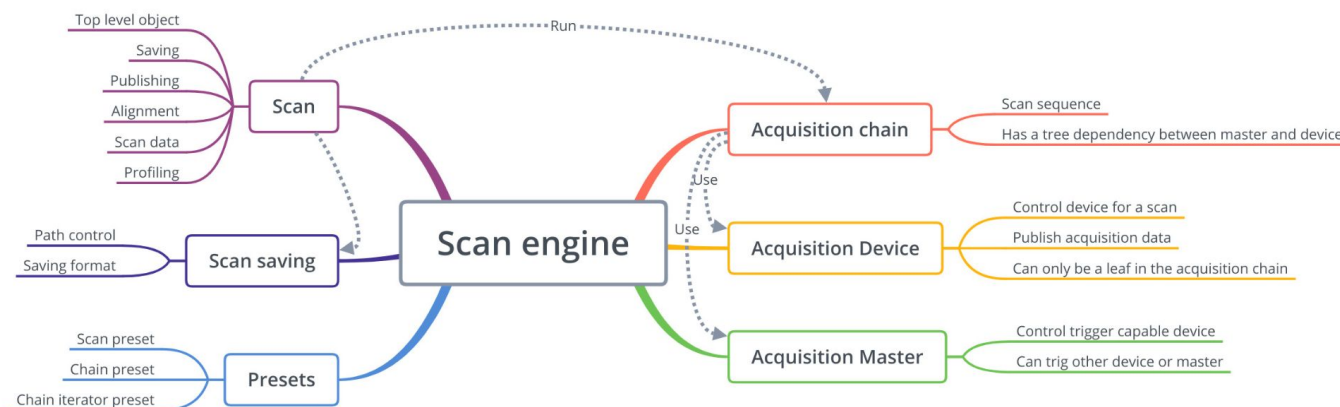
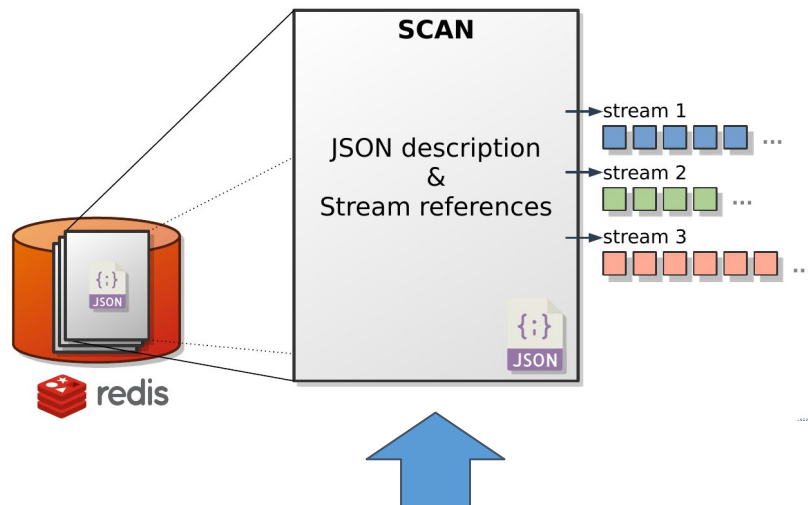
- Append-only data structure
- Random access in $O(1)$
- Unique ID for each stream entry
- Stream consumption strategies
 - Sequential consumers from specific TS
 - Consumer Groups
 - last delivered id
 - Processing ack
 - Single entry, ranges,...



Scan object

...remember... in blissdata
everything built around **Scan**
and **Stream**

Blissdata



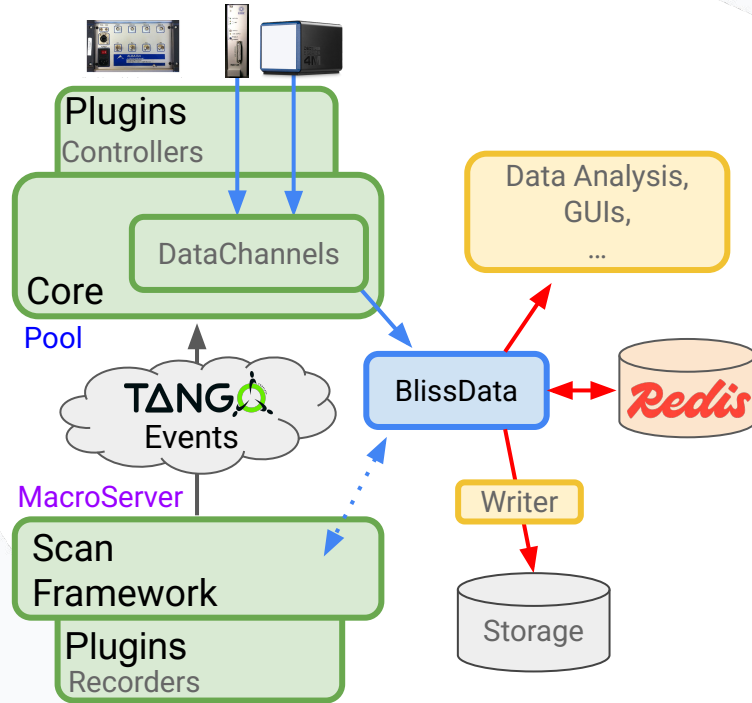
Scan object

```
scan_info = {
    #####
    # Scan metadata
    #####
    "name": scan.name,
    "scan_nb": scan.number,
    "session_name": scan.session,
    "data_policy": scan.data_policy,
    "start_time": datetime.now().astimezone().isoformat(),
    "title": f"{scan_name} {npoints} {count_time}",
    "type": f"{scan_name}",
    "npoints": npoints,
    "count_time": count_time,
    #####
    # Device information
    #####
    "acquisition_chain": {
        "timer": {
            "top_master": "timer",
            # Should not be anymore needed, but expected by the schema
            "devices": ["timer", "scalars", "vectors", "images", "external_images"],
            # Should not be anymore needed, but expected by the schema
            "scalars": [],
            # Should not be anymore needed, but expected by the schema
            "spectra": [],
            # Should not be anymore needed, but expected by the schema
            "images": [],
            # Should not be anymore needed, but expected by the schema
            "master": {},
        },
    },
    "devices": {
        "timer": {
            "name": "timer",
            "channels": ["epoch"],
            "triggered_devices": [
                "scalars",
                "vectors",
                "images",
                "external_images",
            ],
            "scalars": {"name": "scalars", "channels": ["scalars"]},
            "vectors": {"name": "vectors", "channels": ["vectors"]},
            "images": {"name": "images", "channels": ["images"]},
            "external_images": {
                "name": "external_images",
                "channels": ["external_images"],
            },
        },
        "channels": {
            "epoch": {"dim": 0},
            "scalars": {"dim": 0},
            "vectors": {"dim": 1},
            "images": {"dim": 2},
            "external_images": {"dim": 2},
        },
    },
}
```

```
#####
# Plot metadata
#####
"display_extra": {"plotselect": [], "displayed_channels": []},
"plots": [],
#####
# Nexus writer metadata
#####
"save": True,
"filename": filename,
"images_path": images_path,
"publisher": "test",
"publisher_version": "1.0",
"data_writer": "external",
"writer_options": {"chunk_options": {}, "separate_scan_files": False},
"scan_meta_categories": [
    "positioners",
    "nexuswriter",
    "instrument",
    "technique",
],
"nexuswriter": {
    "devices": {},
    "instrument_info": {"name": "esrf-id00a", "name@short_name": "id00"},
    "masterfiles": masterfiles,
    "technique": {},
},
"positioners": {
    "positioners_dial_start": {"robx": 0.0, "roby": 0.0, "robz": 0.0},
    "positioners_start": {"robx": 0.0, "roby": 0.0, "robz": 0.0},
    "positioners_units": {"robx": None, "roby": None, "robz": "mm"},
},
"instrument": {
    "@NX_class": "NXinstrument",
    "att1": {"@NX_class": "NXattenuator", "status": "in", "type": "Al"},
    "machine": {
        "@NX_class": "NXsource",
        "automatic_mode": True,
        "current": 67.0,
        "current@units": "mA",
        "filling_mode": "7/8 multibunch",
        "front_end": "",
        "message": "You are in Simulated Machine",
        "mode": "USM",
        "name": "ESRF",
        "refill_countdown": 84.0,
        "refill_countdown@units": "s",
        "type": "Synchrotron",
    },
    "primary_slit": {
        "@NX_class": "NXslit",
        "horizontal_gap": 10.0,
        "horizontal_offset": 0.0,
        "vertical_gap": 20.0,
        "vertical_offset": 0.0,
    },
},
}
```

Nexus Writer and Flint
need all this information

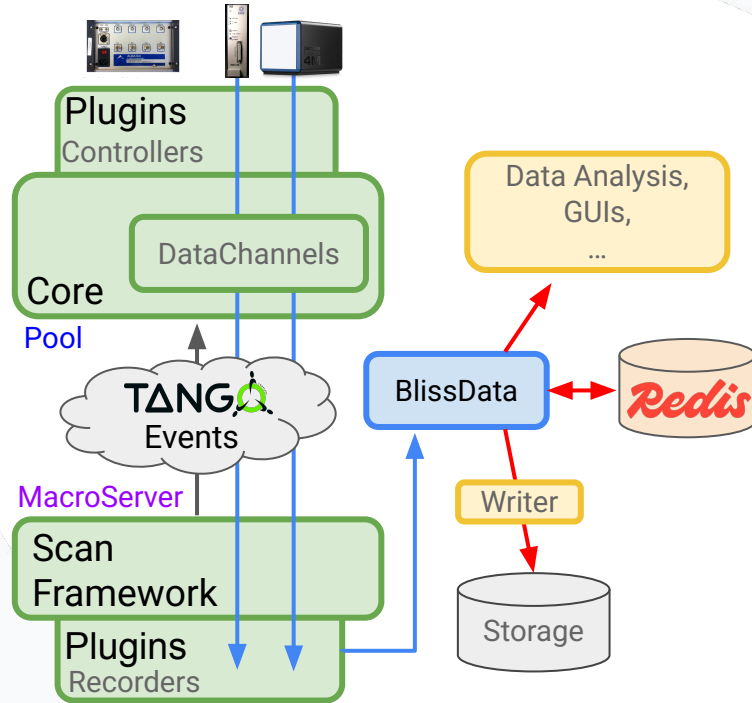
Blissdata and Sardana: Current Status



Core implementation

- Scan object created in Scan Framework (GScan -> scan_info)
- Streams populated from DataChannels
- Pool/MS usually 2 different processes in Sardana

Blissdata and Sardana: Current Status



Data composition still in Scan Framework

Recorder implementation

Constructor

- DB connection
- Create Redis **scan**

_StartRecordList

- Populate scan_info
- Create **streams** (dtype, shape, encoder,...)
- Prepare and start scan

_writeRecord

- Send data to streams

_endRecordList

- Stop and close scan (seal streams)

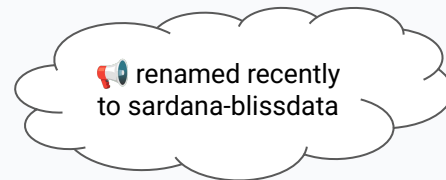


Availability

Recorder implementation

<https://gitlab.com/sardana-org/sardana-blissdata>

- **Blissdata 2.x**
- LimaStream support for sardana-limaccds 2D exp channels
- Macros for Nexus writer configuration and Experiment metadata management
- Documented and easy to test via Pixi
- Basic SPEC writer client





Availability

Core implementation

https://gitlab.com/sardana-org/sardana/-/merge_requests/2065

- Blissdata 2.x.
- LimaStream support for sardana-limaccds 2D exp channels (WiP, conflicts with NXscanH5_FileRecorder in continuous scans and managing NaNs)
- Pending to disable tango events Pool-MS for DataChannels.
- Pending management of multiple sync descriptions and OUTPUT/JSON recorders
- To be reviewed and tested (e.g. not tested with HW synchronization, probably will need to be adapted to manage multiple values received).

Blissdata API for scans and streams

Listeners based on Scan States

Sequential (cursors) and random access

```
Door macroserver 1 [ 39]: ascan dmot01 1 10 10 .1
Connected to redis://localhost:6379
blov/spec writer/test is OFF
blov/bliss nxwriter/test session is OFF
Nexus writer saving is disabled, check NexusWriterOpts
Scan #9 started at Sun May 5 11:32:48 2024. It will take at least
0:00:04.631810
```

#Pt	No	dmot01	ct01	zerod01	oned01	ttdouble	dt
0	1	0.1	98.0511	(1024,,)	26.7599	2.48938	
1	1.9	0.1	100.73	(1024,,)	22.3125	2.89027	
2	2.8	0.1	100.58	(1024,,)	22.3125	3.30872	
3	3.7	0.1	99.9522	(1024,,)	22.3125	3.73288	
4	4.6	0.1	101.245	(1024,,)	22.3125	4.15194	
5	5.5	0.1	99.615	(1024,,)	22.3125	4.55655	
6	6.4	0.1	99.0825	(1024,,)	17.8583	5.01625	
7	7.3	0.1	101.788	(1024,,)	17.8583	5.44547	
8	8.2	0.1	99.6318	(1024,,)	17.8583	5.88142	
9	9.1	0.1	100.754	(1024,,)	17.8583	6.27416	
10	10	0.1	100.434	(1024,,)	17.8583	6.67517	

```
Scan #9 ended at Sun May 5 11:32:55 2024, taking 0:00:06.794788.
Dead time 60.0% (setup time 23.8%, motion dead time 54.2%)
```

```
In [1]: from sardana_redis.utils.sardana_redis_utils import
get_data_store

In [2]: db = get_data_store( "redis://localhost:6379" )

In [3]: timestamp, key = db.get_last_scan()

In [4]: db.get_last_scan()
Out[4]: ('1714901568550-0', 'esrf:scan:01HX41JK15HVK1ANB18Q2CQKJ1')

In [5]: scan = db.load_scan(key)

In [6]: scan.streams
Out[6]:
{'#Pt No': <blissdata.redis_engine.stream.Stream at
0x7f663fd595b0>,
'dmot01': <blissdata.redis_engine.stream.Stream at 0x7f663ff3d2e0>,
'ct01': <blissdata.redis_engine.stream.Stream at 0x7f663ff55a60>,
'zerod01': <blissdata.redis_engine.stream.Stream at
0x7f663fe24e20>,
'oned01': <blissdata.redis_engine.stream.Stream at 0x7f663fe11b80>,
'ttdouble': <blissdata.redis_engine.stream.Stream at
0x7f663fe21760>,
'dt': <blissdata.redis_engine.stream.Stream at 0x7f663fe21910>}

In [7]: scan.streams['ttdouble'][3]
Out[7]: 22.312528079133646
```



Example LimaStream with sardana-limaccds for NexusWriter

```

props = sar_limaccd_2d_ctrl.get_property([
    "LimaCCDDeviceName",
    "FirstImageNumber",
    "H5DatasetPath",
    "HardwareSavingExtraPrefix"])

...

LimaStream.make_definition(
    expchan_label,
    dtype=dtype,
    shape=shape,
    server_url=props["LimaCCDDeviceName"][0],
    frames_per_acquisition=self.macro.nb_points,
    acquisition_offset=sar_limaccd_2d_ctrl.SavingNextNumber,
    saving={
        "file_offset": sar_limaccd_2d_ctrl.SavingNextNumber,
        "frames_per_file": sar_limaccd_2d_ctrl.SavingFramePerFile,
        "file_format": "hdf5",
        "file_path": value_ref_pattern,
        "data_path": props["H5DatasetPath"],
    },
)

...

i = self.first_image_number + record.recordno
ch_stream.send({"last_index": i, "last_index_saved": i})

```

Device properties [controller/limaccdtwodcontroller/ctrl_lima2D]

Property name	Value
DataArrayVersion	2
firstimagenumber	0
H5DatasetPath	entry_0000/measurement/data
klass	LimaCCDTwoDController
latencytime	0
library	LimaCCDCtrl.py
limaccddevicename	lima/limaccds/1
type	TwoDExpChannel

from controller

Experiment Configuration

View mode Edit mode You started editing experiment configuration. External changes will show a pop-up dialog.

Measurement Group Snapshot Group Storage

Active Measurement Group test_meas

	Channel	enabled	output	Data Type	Plot Type	Plot Axes	Timer	Monitor	Synchronizer	Synchronization	Ref Enabled	Ref Pattern	Cond
1	ct01	true	true	float64	No	<mov>	ct01	ct01	software	Trigger			
2	zero01	true	true	float64	No	<mov>	oned01	oned01	software	Trigger			
3	oned01	true	true	float64	No	<mov>	twod01	twod01	software	Trigger	false		
4	twod01	false	true	float64	Image	<mov>	fsm_sim1	fsm_sim1	software	Trigger	true	file://home/...	
5	fsm_sim1	true	true	float64	No								

Reload Keep editing Apply

from measurement group configuration



Example LimaStream with sardana-limaccds for NexusWriter

```
Door_macroserver_1 [ 7]: get_lima_conf fsm_sim1
fsm_sim1
{'create_folders': True,
 'directory': '{ScanDir}/fsm_sim1',
 'index': '04d',
 'prefix': 'fsm_sim1_',
 'scan_sub_dir': 'scan_{ScanID:04d}',
 'suffix': '.h5'}
```

```
Door_macroserver_1 [ 8]: lima hook
```

```
Configured fsm_sim1 channel according to the lima configuration: file:///tmp/scans/fsm_sim1/scan_0141/fsm_sim1_{index:04d}.h5
```

```
Door_macroserver_1 [ 9]: ascan dcm kev 12 15 5 .2
```

```
Connected to Redis at redis://localhost:6379
```

```
No RedisWritersTango environment variable found
```

```
Scan Redis KEY esrf:scan:01JX7GE0D4JRXQQ24RNC64FKPH
```

```
Scan #142 started at Sun Jun 8 12:12:28 2025. It will take at least 0:00:02.785343
```

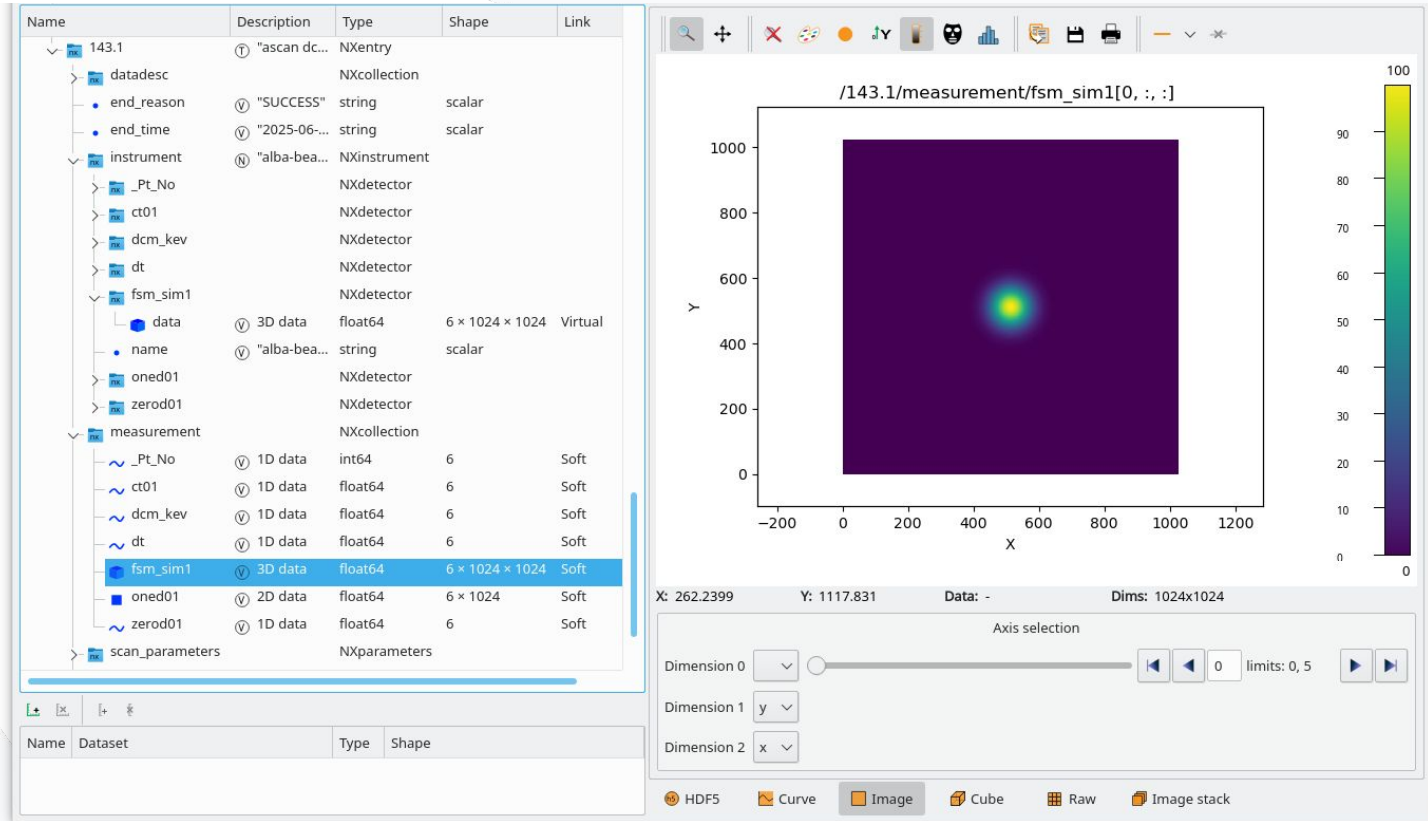
```
Scan URL /tmp/scans/nxwriter.h5:/143.1
```

#Pt	No	dcm_kev	ct01	zerod01	oned01	fsm_sim1	dt
0	12	0.2	99.5941	(1024,)	h5file:///tmp/scans/fsm_sim1/scan_0141/fsm_sim1_0000.h5:entry_0000/measurement/data	0.55001	
1	12.6	0.2	100.406	(1024,)	h5file:///tmp/scans/fsm_sim1/scan_0141/fsm_sim1_0000.h5:entry_0000/measurement/data	1.04009	
2	13.2	0.2	99.2838	(1024,)	h5file:///tmp/scans/fsm_sim1/scan_0141/fsm_sim1_0000.h5:entry_0000/measurement/data	1.52264	
3	13.8	0.2	100.197	(1024,)	h5file:///tmp/scans/fsm_sim1/scan_0141/fsm_sim1_0000.h5:entry_0000/measurement/data	2.01013	
4	14.4	0.2	99.6477	(1024,)	h5file:///tmp/scans/fsm_sim1/scan_0141/fsm_sim1_0000.h5:entry_0000/measurement/data	2.49502	
5	15	0.2	99.9543	(1024,)	h5file:///tmp/scans/fsm_sim1/scan_0141/fsm_sim1_0000.h5:entry_0000/measurement/data	3.01098	

```
Scan #142 ended at Sun Jun 8 12:12:32 2025, taking 0:00:03.492297. Dead time 65.0% (setup time 0.6%, motion dead time 47.9%)
```



Example LimaStream with sardana-limaccds for NexusWriter





Demo time

- ✓ Configure recorder (including Lima exp channel)
- ✓ Environment variables and macros
- ✓ ScanRecorder vs DataRecorder
- ✓ View data in Redis
- ✓ NexusWriter
- ✓ Core implementation
- ✓ Flint?



Issues detected and pending tasks

- Installation conflict: Blissdata 2.x requires pydantic >2 and Sardana (config) requires pydantic<2. We need to update Sardana-config to use pydantic 2 (or remove from sardana project and keep it as a separate tool, as it is planned).
- ESRF NexusWriter is working with base Streams and LimaStreams. To be seen how to adapt for custom streams. E.g. other 2d reference systems.
- Problems with Flint plotting data
- Sardana Pool/MS as separate processes publishing to the same stream. Current implementation is not efficient.
- Improve core implementation (store streams in pool, disable tango events, multiple sync descriptions)
- Correlation of values in Redis with no timestamp when mixing software/hardware.



Thank you
time for questions

Blissdata integration in Sardana

Roberto Homs
Jordi Aguilar
Oriol Vallcorba
Zbigniew Reszela

Sardana Workshop 2025
27–29 Aug 2025
MAX IV Laboratory