

# ATEF

## Automated Test Execution Framework

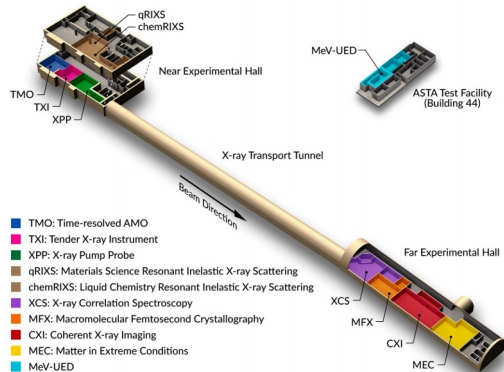
---

Robert Tang-Kong / Experimental Controls Systems  
Alex Wallace presenting

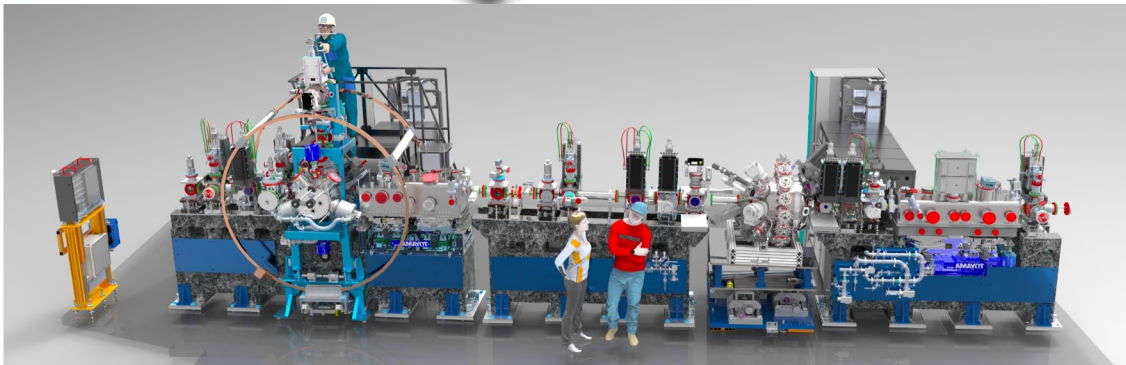
2023

# LCLS is a complicated place

## Things change frequently



- Beamlines are many, and they are complicated
  - Devices are frequently inserted and removed
- Beamlines on our linear lightsource are not fully independent



How do we manage this variability?

# “Checkouts” verify the state of the beamline

## The status quo is painfully manual and error prone

TMO (K4-Line)

TMO (K4-line starting in FEE)
<input type="checkbox"/> Review interlock overrides from previous shutdown sequence. <input type="checkbox"/> Check that vacuum systems that were vented have started to be pumped down to recover vacuum. Provide comment:
<b>Vacuum</b> <input type="checkbox"/> All pumps are on <input type="checkbox"/> All ion pumps and cold cathode report below 1e-7 Torr, if not make note <input type="checkbox"/> Gate valves and fast shutters can all be opened (open and then close again) <input type="checkbox"/> TMO starting at TV1K4-VGC-1 ---- Ending at VLS-VGC-01 (if it exists), or PCSK4-VGC-01 <input type="checkbox"/> TV1K4-VFS-1 <input type="checkbox"/> MR3K4-KBO-VFS-1
<b>Square One</b> <input checked="" type="checkbox"/> Check SQ 1 connection <input checked="" type="checkbox"/> Start SQ1 Remote Desktop session <input checked="" type="checkbox"/> Run LabView GUI <input checked="" type="checkbox"/> Ensure all readback values update (check X and Y with Andrei and no issues) <input checked="" type="checkbox"/> Record a screen shot of the current motor positions. These values will be used as a reference to return to before removing the endstation (I saved in hutch computer before figure out where I should save the image)
<b>DAQ Check</b> <input type="checkbox"/> Check DAQ is running on the correct event code (120 for beam or possibly 10) <input type="checkbox"/> All detectors needed for experiment included in roll call <input type="checkbox"/> DAQ goes from restart to running state <input type="checkbox"/> epicsArch points at current experiment <input type="checkbox"/> Run MPOD update script for HV system (mpod_desc_to_arch)
<b>Gas cabinet</b> <input type="checkbox"/> Check it is safe to set panel to service mode <input type="checkbox"/> Set panel to service mode
<b>X-ray Optics Mechantronics</b> <input type="checkbox"/> Verify MR1K4 pitch is within acceptable range (80-120 urad). <input type="checkbox"/> Move pitch small amount (5 urad)

Pages / MFX@LCLS Home / 1. Beamline Operation

Edit Save for later Watch

## 1.5 MFX Checkout

Created by Leland Bruce Gee, last modified by Frederic P Poitevin on Sep 21, 2023

Before each beamtime:

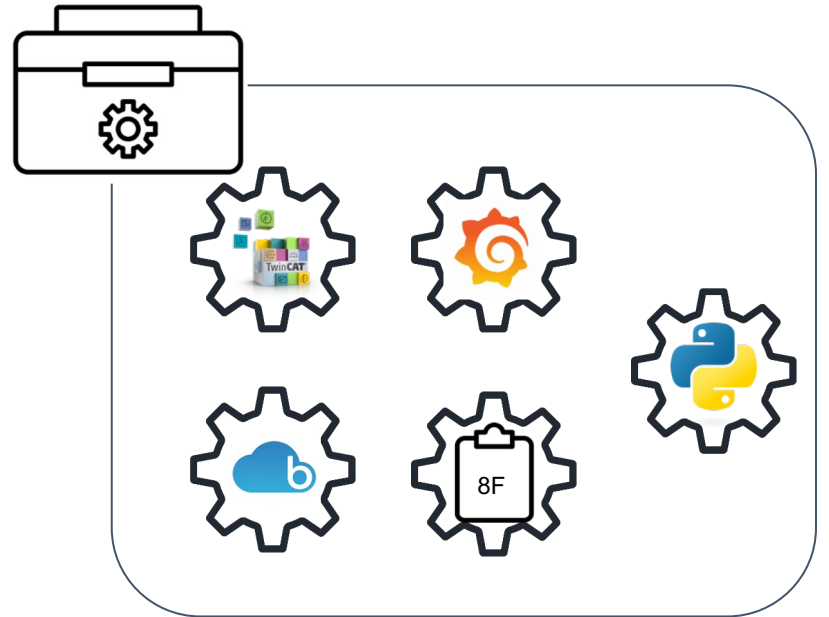
- copy the [checkout sheet](#)
- go through it and report any issue in [#mfx](#)

MFX Pre-beam Checkout			
Area	Task	Date Completed	Comment
Vacuum	<input checked="" type="checkbox"/> Gate Valve Permissions		
	<input checked="" type="checkbox"/> Passed RGA Qualification		
	<input checked="" type="checkbox"/> Beamline fully under vacuum		
Radiation Physics	<input checked="" type="checkbox"/> HPS certification valid		
	<input checked="" type="checkbox"/> BLA signed		
	<input checked="" type="checkbox"/> Test hutch search performed		
	<input checked="" type="checkbox"/> RSWCF forms all closed	09 Jul 2023	
	<input checked="" type="checkbox"/> Prefocusing and Transfocator Certification		
	<input checked="" type="checkbox"/> Administrative procedures for CRLs ready		
	<input checked="" type="checkbox"/> Modifications to focusing optics approved		

# The solution: ATEF

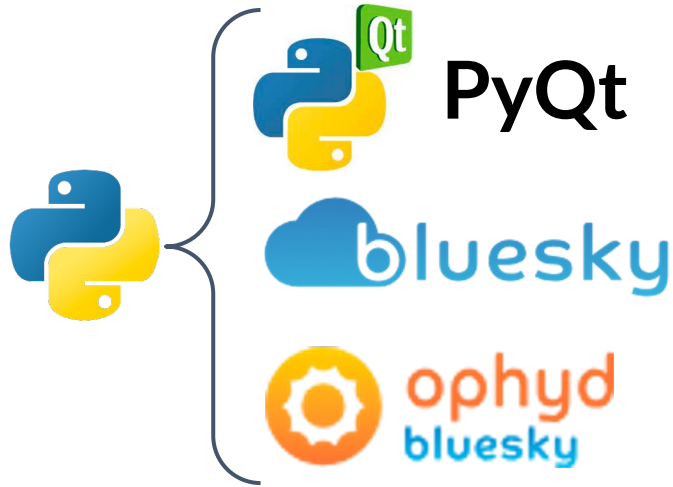
## Automated Test Execution Framework

- Represents one part of the automation toolkit
- **Goals**
  - Make “checkouts” faster and easier
  - Make “checkouts” reproducible
  - Make “checkouts” trackable

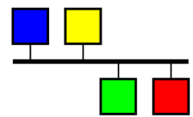


# ATEF: the stack

---

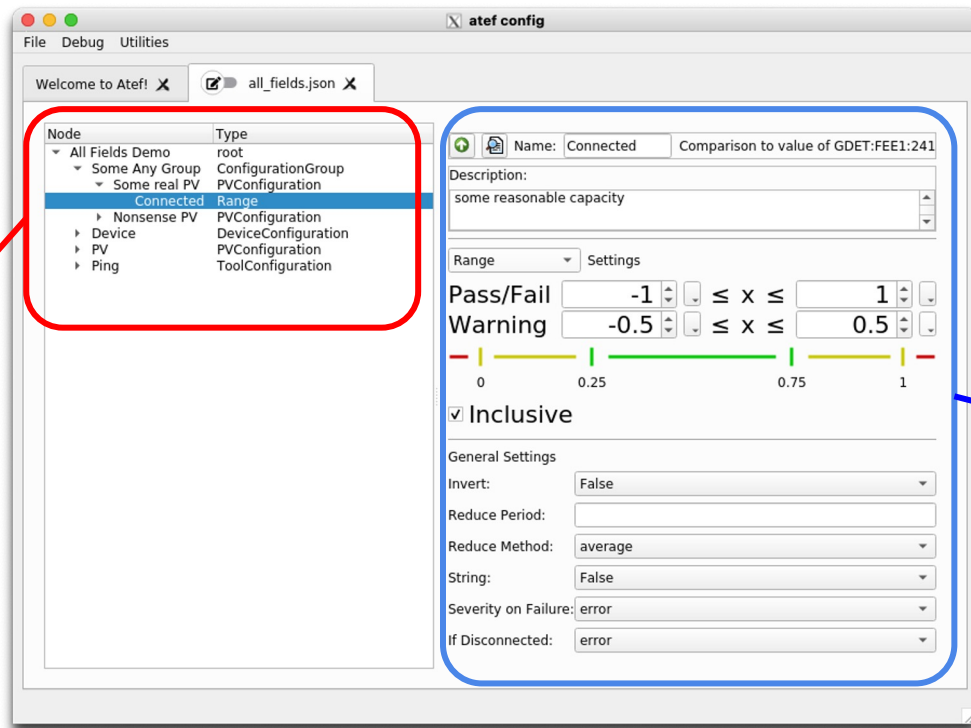


**EPICS**



- PyQt: a flexible GUI library
  - Has hooks for building/modifying everything
- Bluesky: Data Collection Framework
  - Provides experimental plan orchestration
- Ophyd: gets EPICS PVs into python
  - Allows grouping of PVs into logical devices
- EPICS: a standard, widely used controls system
  - Simple get/put interface

# Quick Tour: Basics

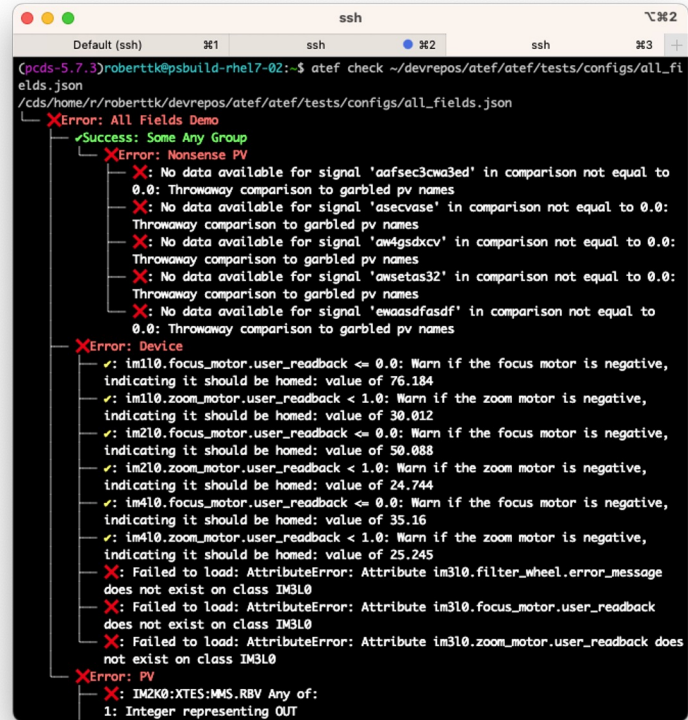
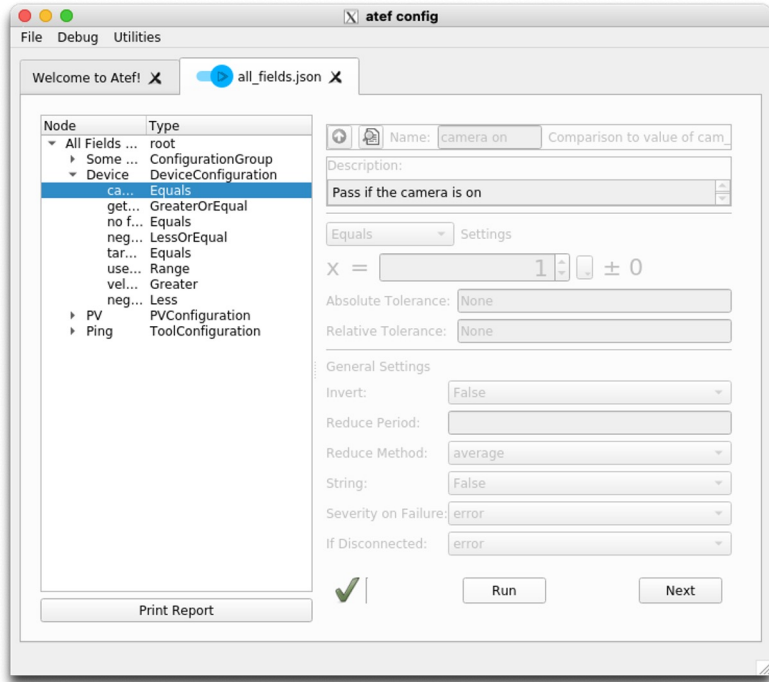


Tree-view shows  
checkout structure

Page view shows  
configurations,  
comparisons, actions,  
etc

# Quick Tour: Running Checkouts

## GUI or command line



# Quick Tour: Passive vs Active Checkouts

---

## Passive Checkouts

- Read Access only
- Steps performed asynchronously
- Steps run without user intervention

## Active Checkouts

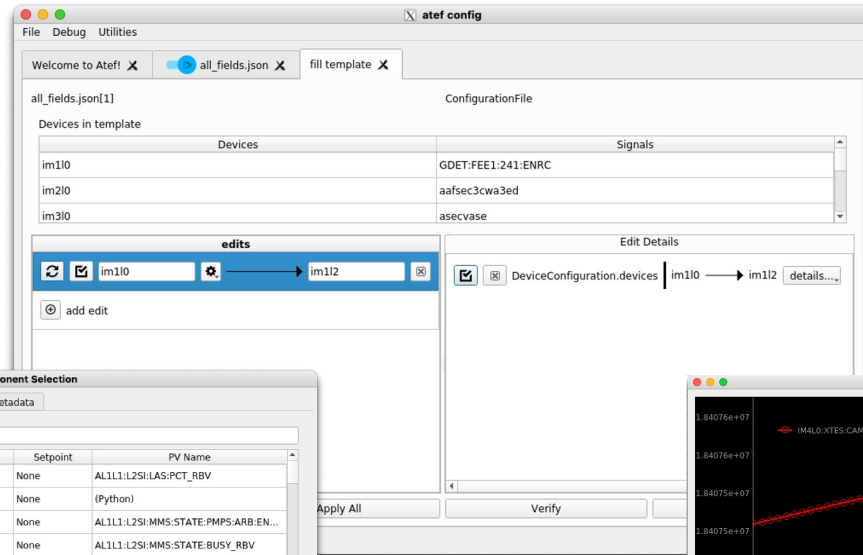
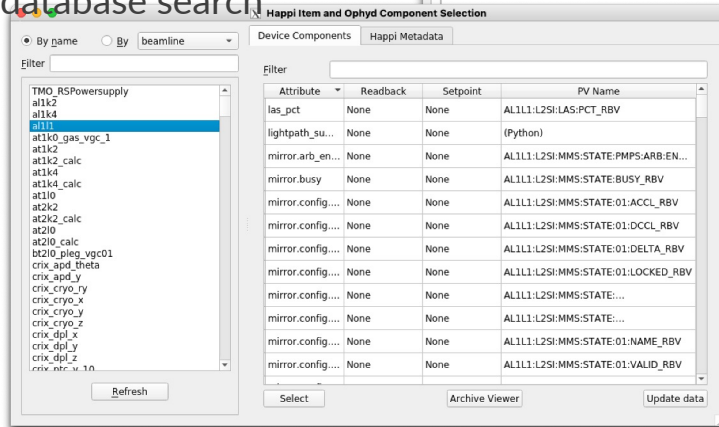
- Read and Write access
- Steps performed sequentially
- Steps allow for user verification
- Requires additional control
  - Error handling
  - Safe abort, resume

**Writing to the control system  
requires caution!**



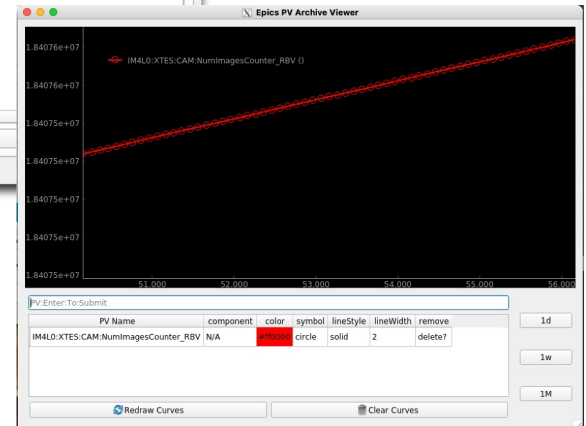
# Utility Tools

Ophyd device  
database search



Find-replace for re-  
using checkouts

EPICS Archiver viewer



# Quick Tour: Reports

- Reports auto-generated from collected data and checkout definitions
- Includes
  - Expected, recorded values
  - Checkout settings
- Minimal-to-no customization needed for reporting any checkout step

**Checkout Report**

All Fields Demo

Author: unknown  
Date: 2023-04-25  
Version: 0.0.1

camera on - im110.cam\_power  
Pass if the camera is on  
success: -

Document Approval  
Name:

**Settings**

invert	F
reduce_period	N
reduce_method	R
string	N
severity_on_failure	S
if_disconnected	S
value	1
rtol	N
atol	N

**Observed Data**

Observed Value	Timestamp
1	Thu Mar 23 14:19:2

**Device**  
Include a device configuration and use it to do all the basic comparisons  
**error: At least one configuration failed to initialize**

**Settings**

tags	[device', 'sphyd']
devices	[im110', 'im210', 'im310', 'im410']

**Results**

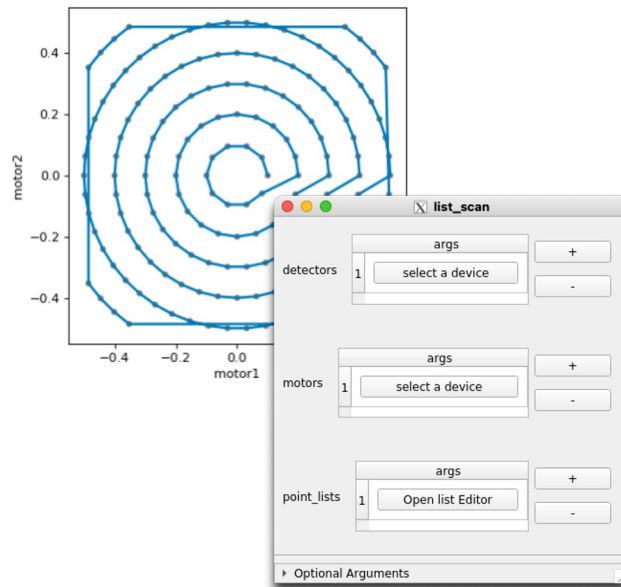
Test Name	Status
camera on - im110.cam_power	success: -
getting images - im110.detector.event_rate	success: -
no filter wheel error - im110.filter_wheel.error_message	success: -
negative focus warning - im110.focus_motor.user_readback	warning: im110.focus_motor.user_readback <= 0.0: Warn if the focus motor is negative, indicating it should be homed: value of 76.104
target out - im110.target.state	success: -
user readback realistic - im110.y_motor.user_readback	success: -
velocity nonzero - im110.y_motor.velocity	success: -
negative zoom warning - im110.zoom_motor.user_readback	warning: im110.zoom_motor.user_readback < 1.0: Warn if the zoom motor is negative, indicating it should be homed: value of 30.012
camera on - im210.cam_power	success: -
getting images - im210.detector.event_rate	success: -
no filter wheel error - im210.filter_wheel.error_message	success: -
negative focus warning - im210.focus_motor.user_readback	warning: im210.focus_motor.user_readback <= 0.0: Warn if the focus motor is negative, indicating it should be homed: value of 50.088
target out - im210.target.state	success: -
user readback realistic - im210.y_motor.user_readback	success: -
velocity nonzero - im210.y_motor.velocity	success: -
negative zoom warning - im210.zoom_motor.user_readback	warning: im210.zoom_motor.user_readback < 1.0: Warn if the zoom motor is negative, indicating it should be homed: value of 24.744
camera on - im310.cam_power	success: -
getting images - im310.detector.event_rate	success: -

Passive Checkout Report

Page: 17

# Future Feature Plans

More ways to checkout



# Lessons Learned

---

Some thoughts in retrospect

## User Feedback

Don't lose sight of the needs of the user



## Well-Defined Scope

Stakeholder requests must be contextualized



## Quality of Life

Small frustrations can snowball into a loss of stakeholder goodwill

## Flexible and Rapid Iteration

Pythonic stack facilitated quick debugging and tight iteration cycles  
Testing and continuous integration is difficult, but worthwhile

# Questions?

Thanks for your attention!

