Optimising the EPICS End User Experience (focus on CS-Studio)

Freddie Akeroyd

ISIS Computing Group
STFC ISIS Facility, GB

Science & Technology Facilities Council
ISIS

IBEX

- IBEX is our EPICS based control system
- We use Eclipse / RCP for the main IBEX GUI
  - using components from CS Studio
- Python for scripting
- Run everything on MS Windows:
  - For compatibility with existing business systems
  - Allows install alongside old SECI system
  - However IBEX can run on Linux too

Different End Users

- Visiting scientist:
  - Quickly and easily perform an experiment
  - Only see relevant information
- Instrument / beamline scientist:
  - Configure the instrument
  - Diagnose problems
- Sample environment technician:
  - Configure equipment
  - Monitor equipment across several beamlines

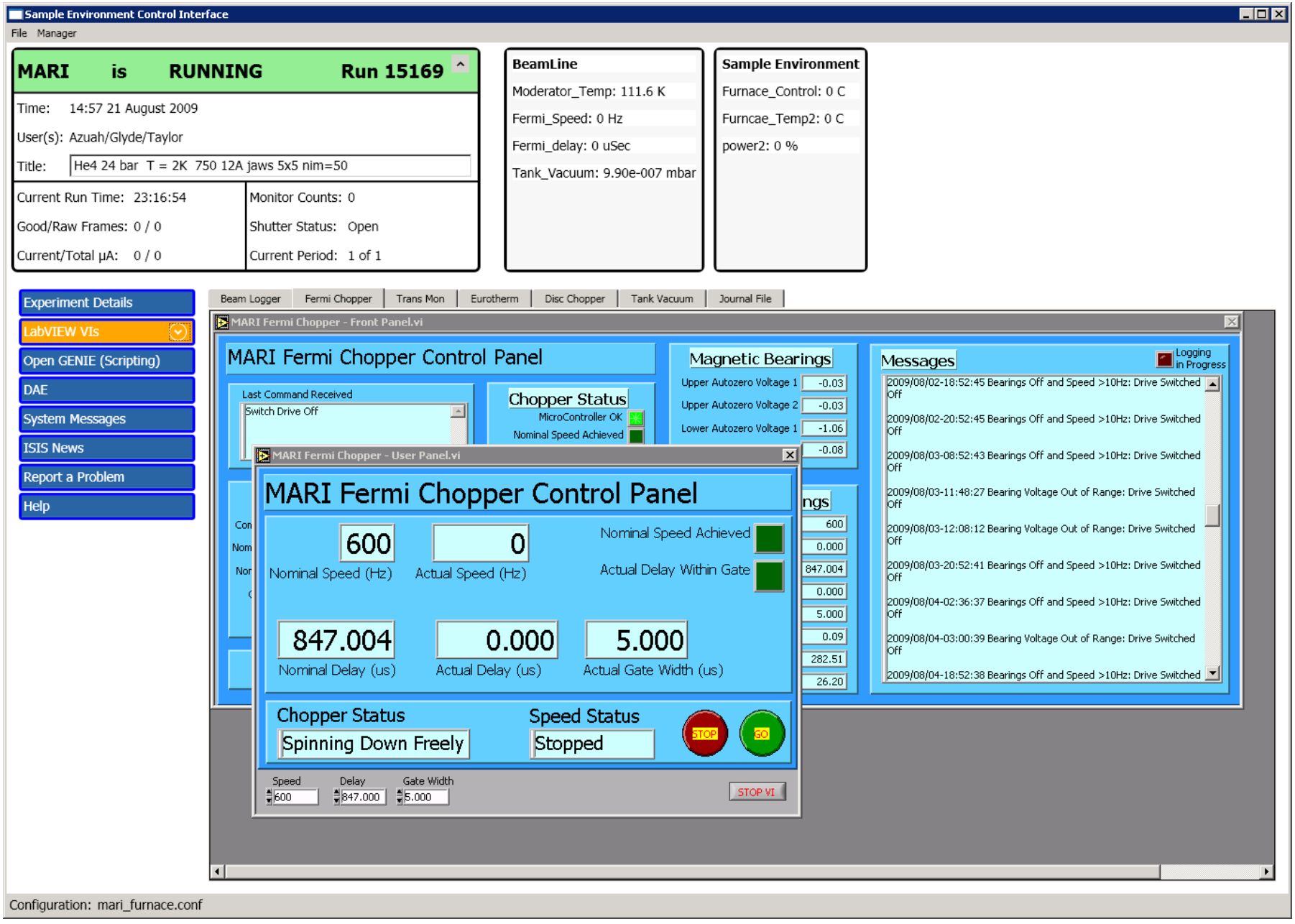
GUI Use Cases

- Web / mobile summary view (read-only)
  - Quick check, all look OK / finished
- Monitor (+ control) an experiment:
  - Read/set values, feedback on running scripts
- Reconfigure an experiment:
  - Start & stop IOCs (add equipment)
  - Change IOC macro values used during startup
  - Create an experiment script sequence
  - Create / edit a new (synoptic) view
  - Change logging parameters

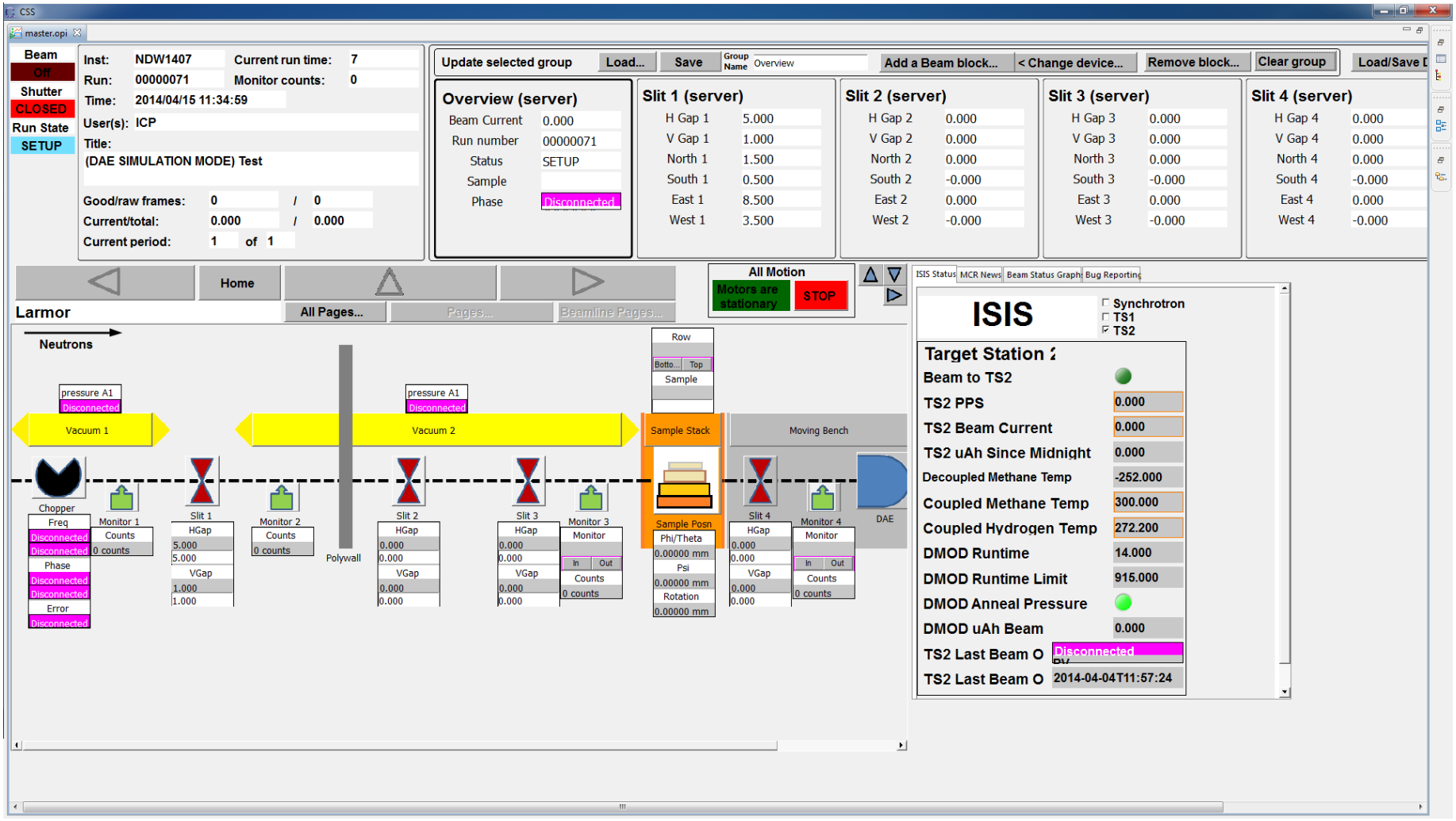
Using CS-Studio

- Originally used pre-built CS-Studio (3.x)
- Built a GUI and synoptic view using BOY
- However couldn't quite get the right "look and feel" for our users
  - See next slides

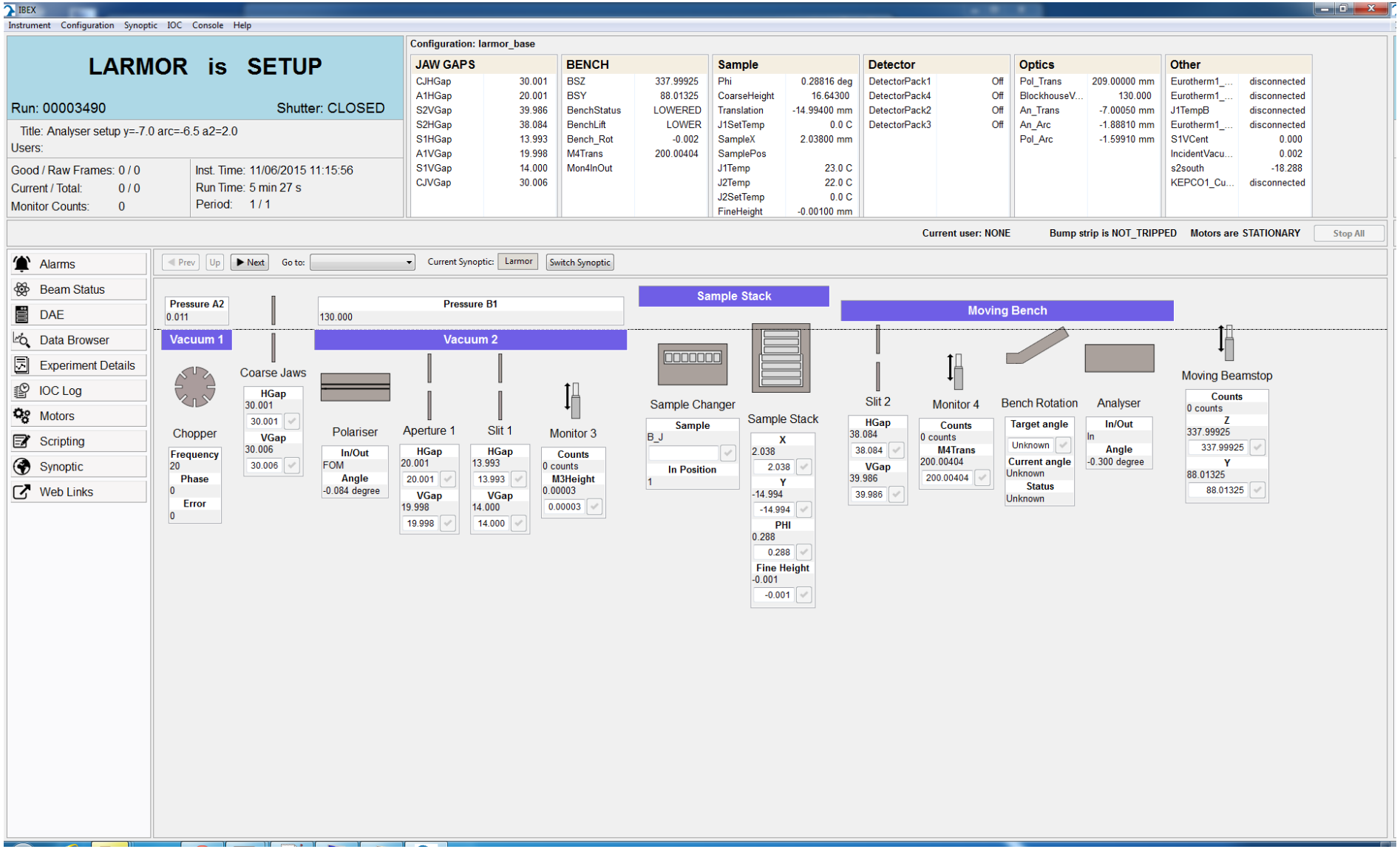
Previous SECI system



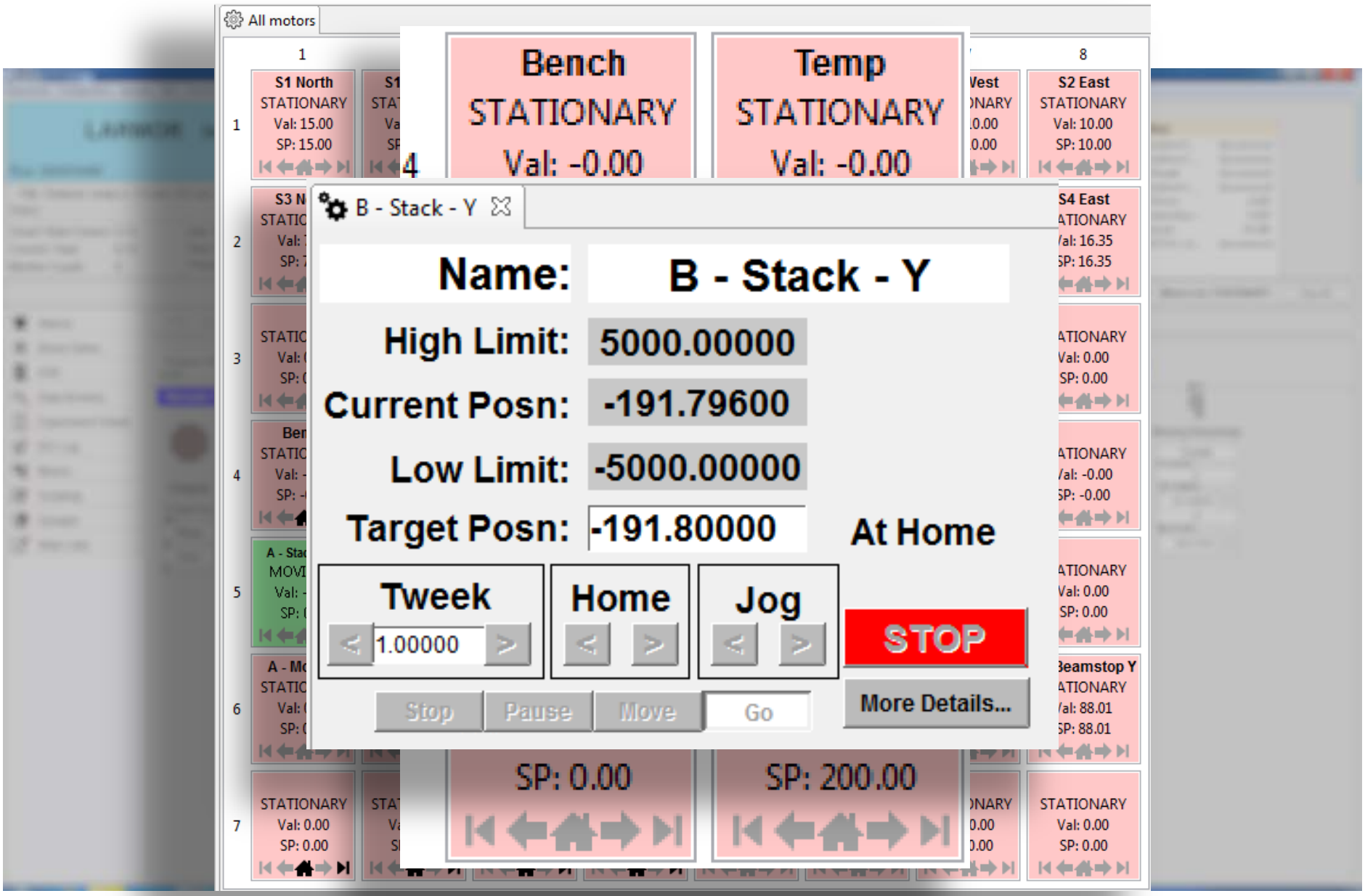
Attempt 1: BOY based GUI



Attempt 2: Eclipse/ RCP Based GUI



IBEX Client: Table of Motors View



The IBEX GUI

- Eclipse / RCP application using components from CS-Studio:
  - Such as PV Manager, data browser, …
  - But have more control over layout
- Also use other components:
  - PyDev scripting console
  - Graphing from Diamond DAWN project
- Can use CS-Studio BOY OPI files too:
  - These are used for synoptic “drill down”

The IBEX GUI (cont.)

- Using both BOY and native eclipse / RCP gives us necessary flexibility:
  - However eclipse / RCP is a steep learning curve
  - Much more work than just using BOY
  - But needed to handle complex configuration management

Other CS-Studio Applications Used

- Two CS-Studio archive engines per beamline
  - for beamline data and for experiment data
- CS-Studio based IOC Log server / ActiveMQ to manage and display IOC log messages
- AlarmServer (BEAST)

Plankton

- A python framework for creating device emulators:
  - Supports stateful devices
  - Supports “back door” to control device behaviour
- Being developed as part of ISIS / ESS in-kind
- We plan to use it for most of our devices
- See https://github.com/DMSC-Instrument-Data/plankton/

Acknowledgements

- Many people have contributed to IBEX:
  - Freddie Akeroyd, Kathryn Baker, Matt Clarke, Simon Fernandez, Lottie Greenwood, Jack Harper, Michael Hart, Gareth Howells, David Keymer, Thomas Lohnert, Chris Moreton-Smith, Dominic Oram (STFC), Martin Bell, Ian Bush, John Holt, Robert Nelson, Adrian Potter, Isabella Rey, Kris Ward, Kevin Woods (Tessella)
- Code hosted at: https://github.com/ISISComputingGroup

Tessella
Technology & Consulting

Science & Technology Facilities Council
ISIS

Part 2: Python Script server

- Something many scientists have asked for
  - Had script writers / generators before
- Hand you over to Dominic Oram

Development of a Script Server at ISIS

Dominic Oram

Science & Technology Facilities Council
ISIS

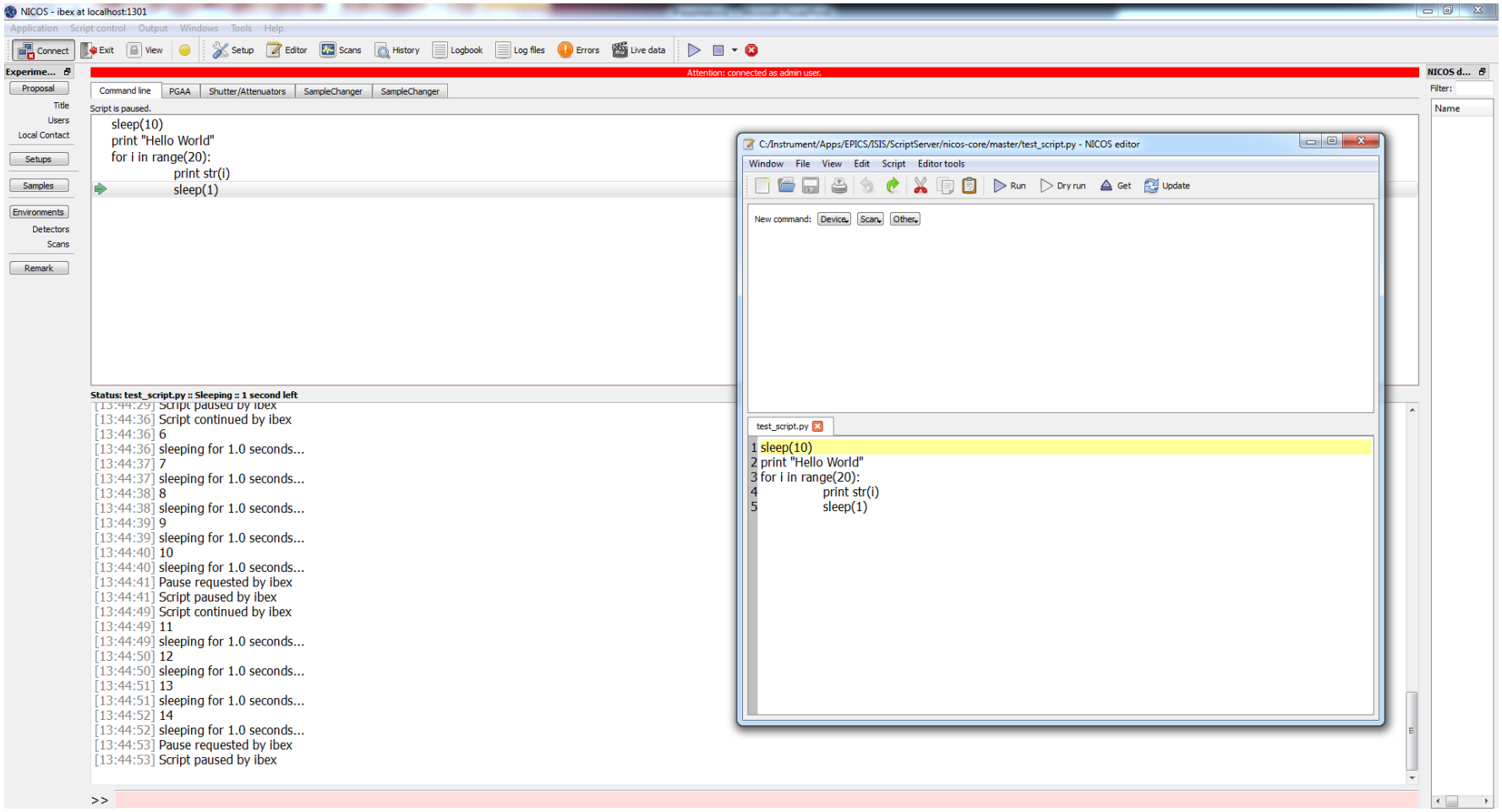
Motivation

- Users run python scripts for longer experiments
- Client side scripting is bad:
  - What happens if the network dies mid script?
  - How do we know who is running the script?
- We need to pause/kill scripts if there is an fault
- Queuing multiple scripts would be useful
- Scientists wish to change a value without having to restart their whole script
- We want to be able to do dry runs of scripts

Prospective Options

Option	Scan Server	NICOS
Used/written by	SNS	FRM II
Written in	Java	Python
GUI	CSS integrated	PyQt based GUI
Client-Server comms	XML	Pickled python objects
Available Commands	Hardcoded (originally)	Added through inheritance
Intended Scope	Used as part of a wider EPICS distribution	Used as a whole control system

NICOS


NICOS - ibex at localhost:1301
Application Script control Output Windows Tools Help
Connect Exit View Setup Editor Scans History Logbook Log files Errors Live data
Attention: connected as admin user.
Command line PGAA Shutter/Attenuators SampleChanger SampleChanger
Script is paused.
sleep(10)
print "Hello World"
for i in range(20):
print str(i)
sleep(1)
Status: test_script.py :: Sleeping :: 1 second left
[13:44:36] Script continued by ibex
[13:44:36] 6
[13:44:36] sleeping for 1.0 seconds...
[13:44:37] 7
[13:44:37] sleeping for 1.0 seconds...
[13:44:38] 8
[13:44:38] sleeping for 1.0 seconds...
[13:44:39] 9
[13:44:39] sleeping for 1.0 seconds...
[13:44:40] 10
[13:44:40] sleeping for 1.0 seconds...
[13:44:41] Pause requested by ibex
[13:44:41] Script paused by ibex
[13:44:49] Script continued by ibex
[13:44:49] 11
[13:44:49] sleeping for 1.0 seconds...
[13:44:50] 12
[13:44:50] sleeping for 1.0 seconds...
[13:44:51] 13
[13:44:51] sleeping for 1.0 seconds...
[13:44:52] 14
[13:44:52] sleeping for 1.0 seconds...
[13:44:53] Pause requested by ibex
[13:44:53] Script paused by ibex
C:/Instrument/Apps/EPICS/ISIS/ScriptServer/nicos-core/master/test_script.py - NICOS editor
Window File View Edit Script Editor tools
Run Dry run Get Update
New command: Device Scan Other
test_script.py
1 sleep(10)
2 print "Hello World"
3 for i in range(20):
4 print str(i)
5 sleep(1)
NICOS d...
Filter:
Name
Experime...
Proposal
Title
Users
Local Contact
Setups
Samples
Environments
Detectors
Scans
Remark
>>

ISIS NICOS Proxy

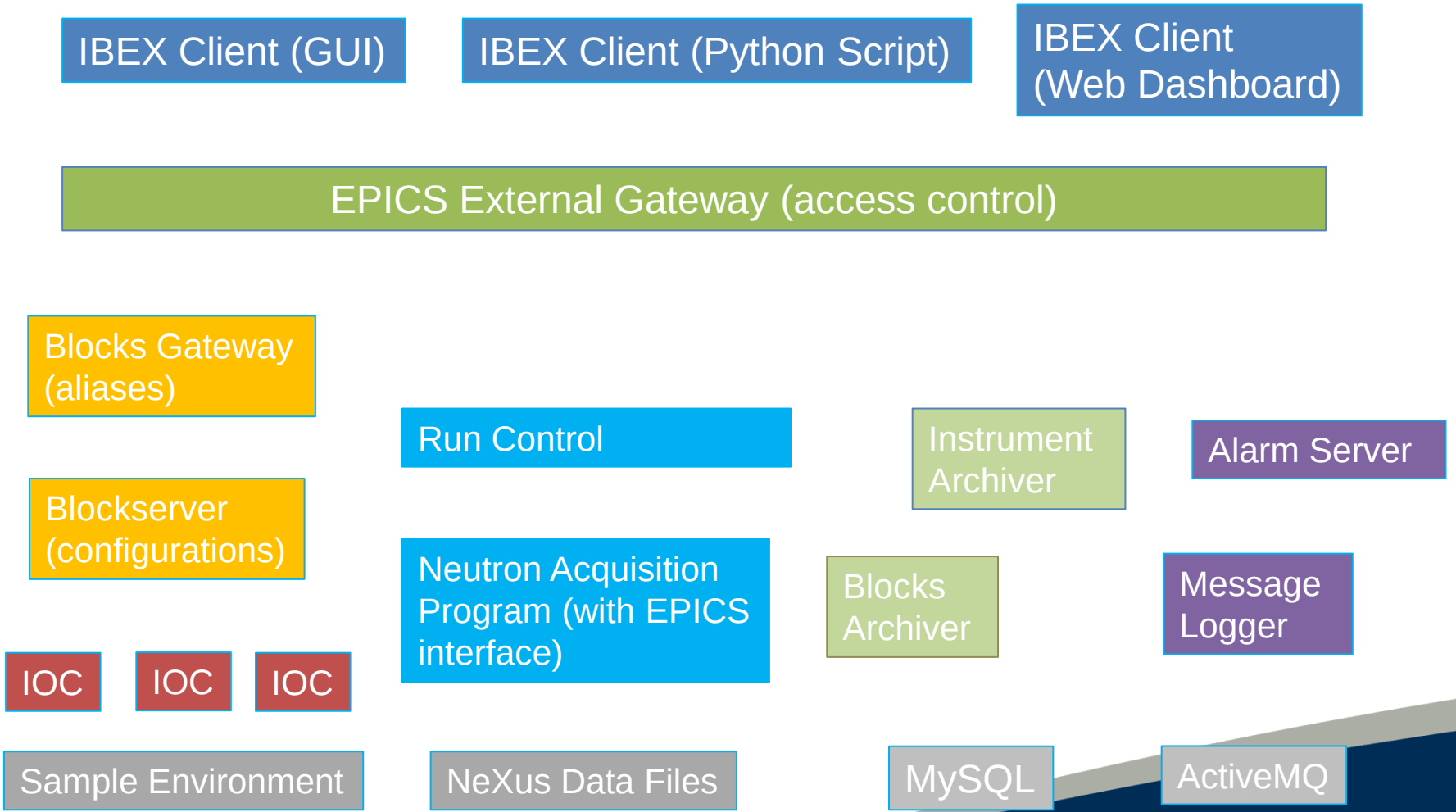
- Can’t integrate a pyQt interface into IBEX
- Wanted to separate GUI and script server implementation
- Java has no good pickled python libraries
- We have written a proxy that allows communication with NICOS via JSON and ActiveMQ
- See https://github.com/ISISComputingGroup/nicos-core

Future Work

- Integrate script server communication into IBEX
- Investigate options for using client-side genie python with the script server
- Create a number of script generator views to aid scientists in creating scripts
- Investigate how to handle changing scripts whilst they are running

Science & Technology Facilities Council
ISIS

IBEX System Components

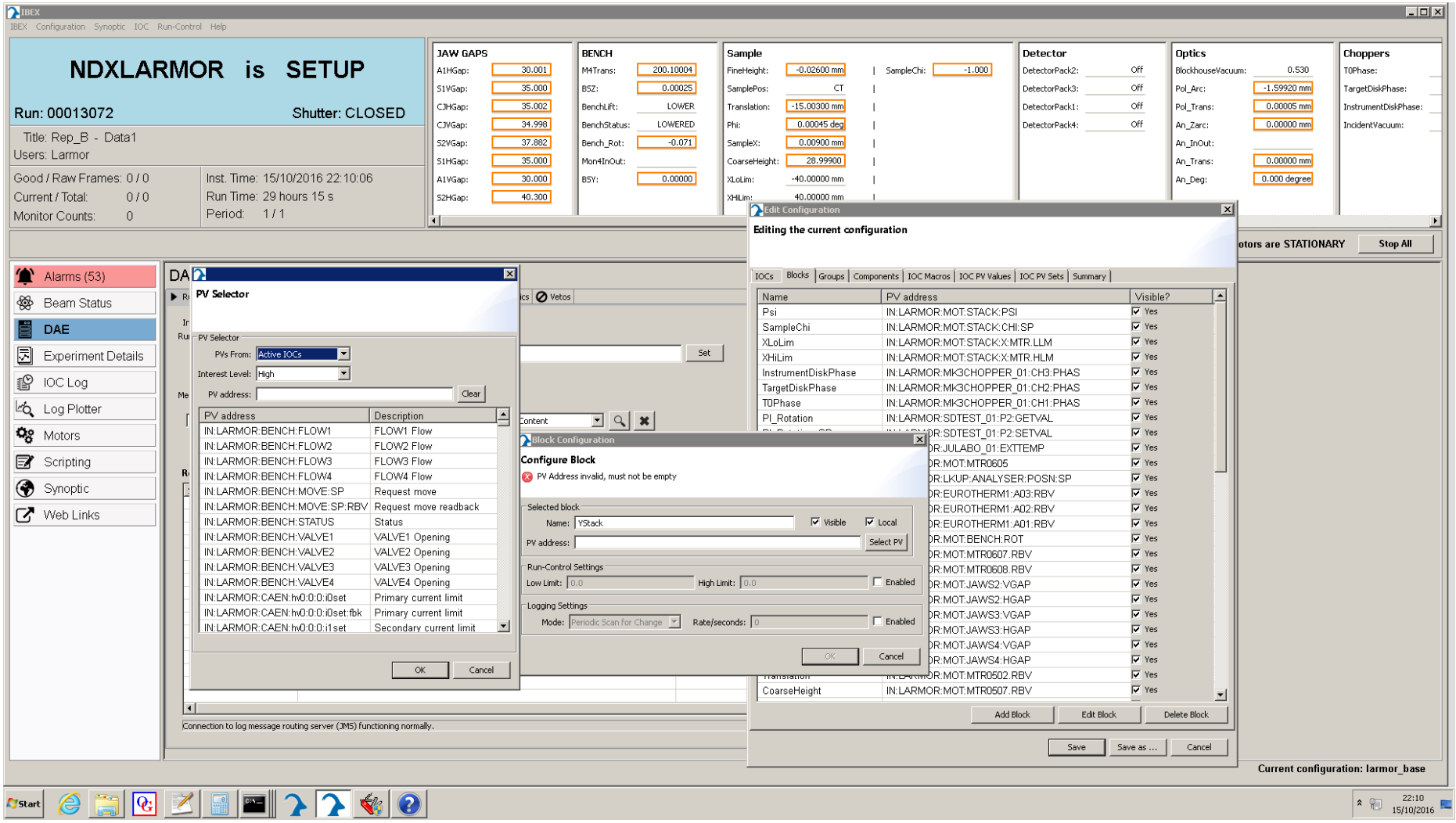


Science & Technology Facilities Council
ISIS

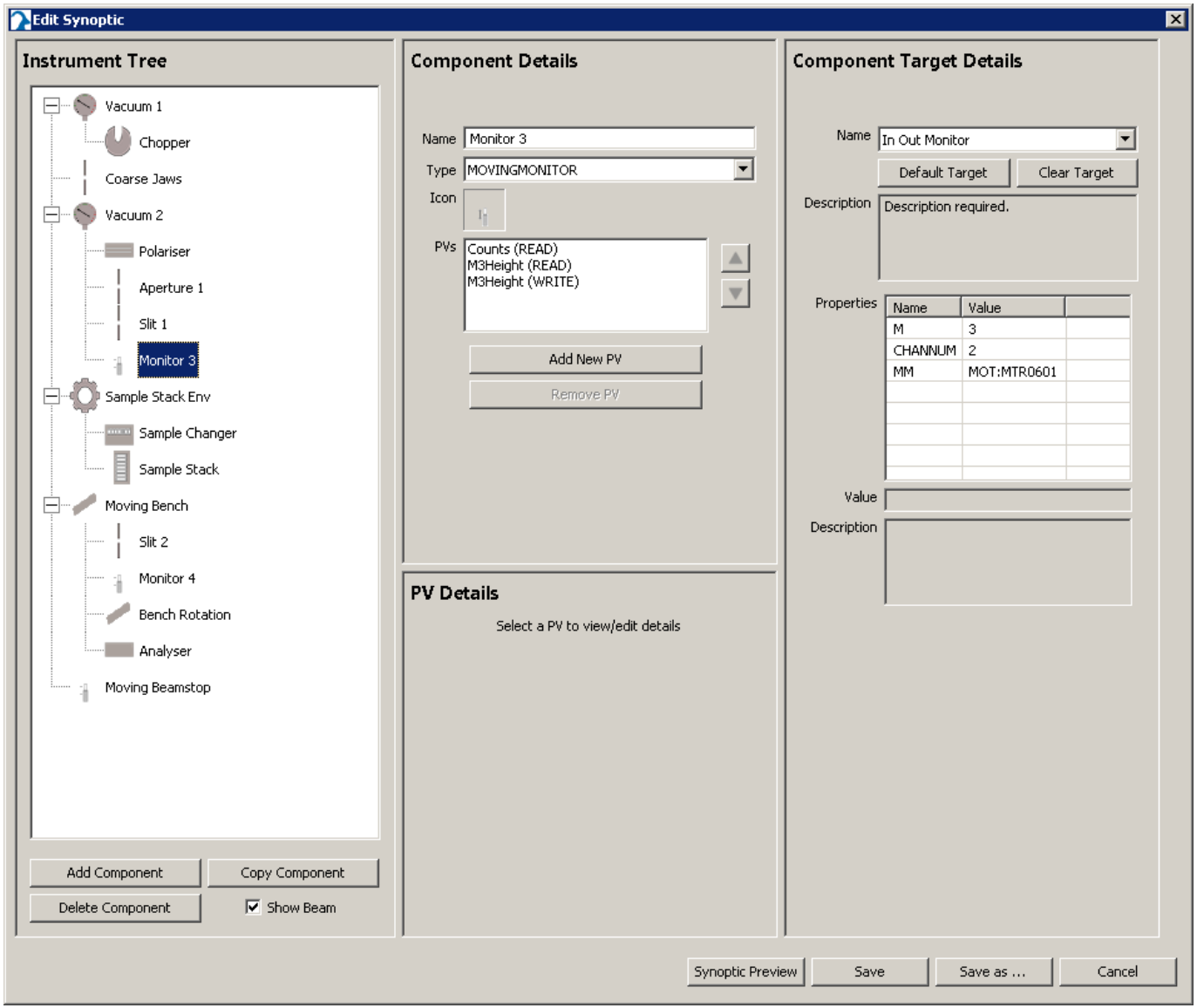
Why Choose CS-Studio?

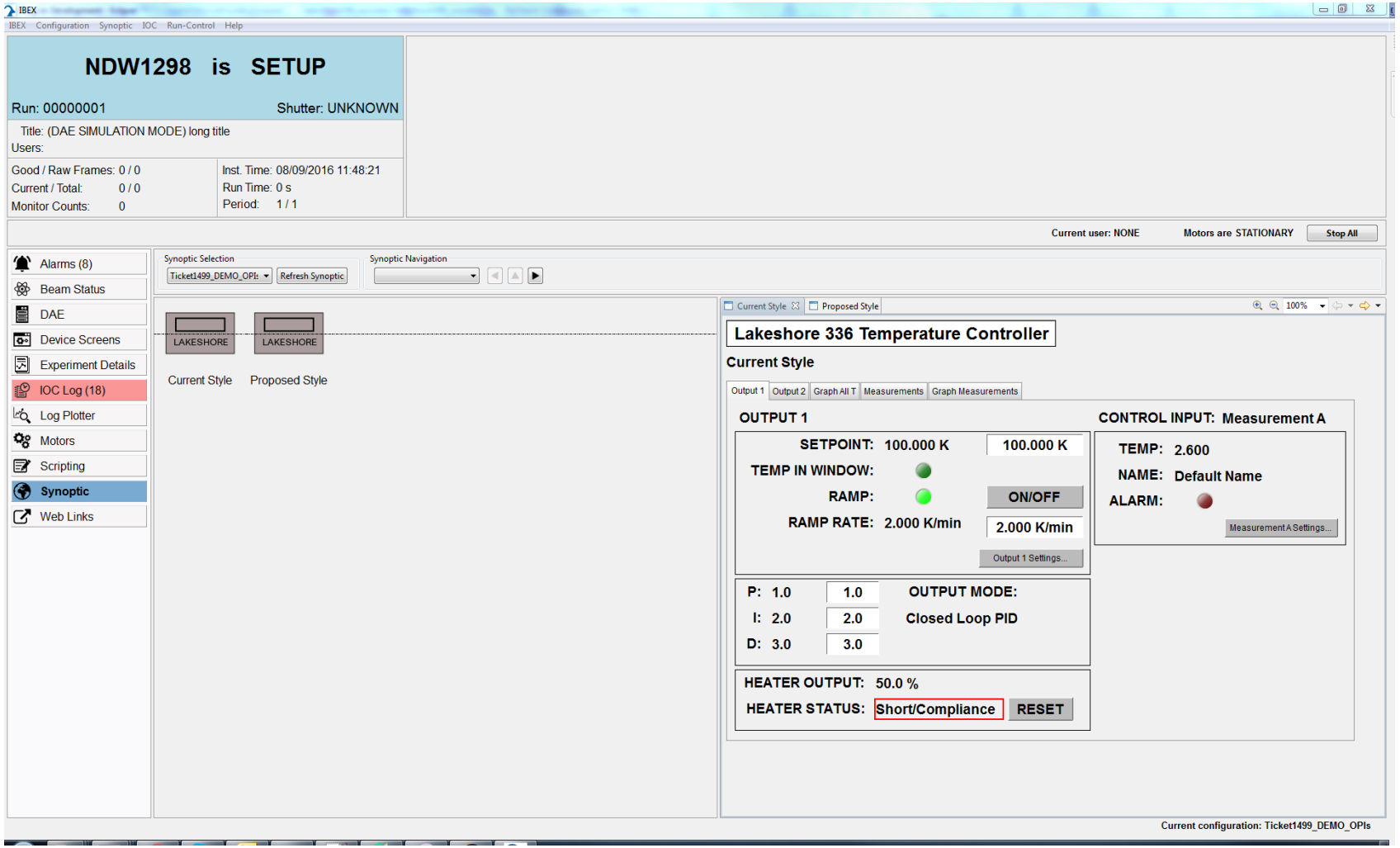
- Chose EPICS, now needed GUI framework
- Only contenders really at the time were:
  - EPICS Qt
  - CS-Studio
- Chose CS-Studio as:
  - Mature collaboration
  - Used as Diamond and SNS, ORNL
  - Provides a lot of functionality

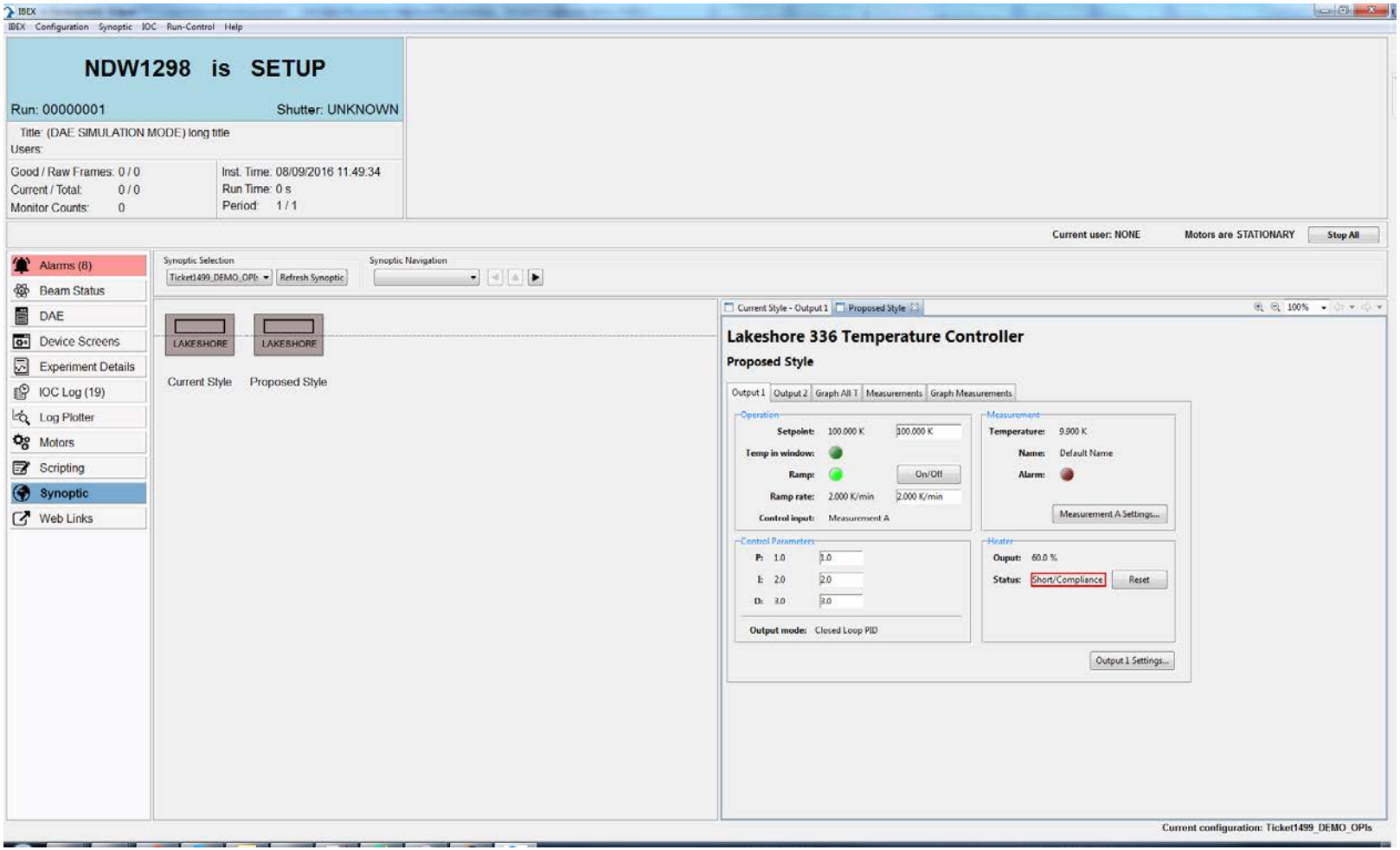
IBEX Client: Adding Blocks



IBEX Client: Synoptic Editor




IBEX
IBEX Configuration Synoptic IOC Run-Control Help
NDW1298 is SETUP
Run: 00000001
Shutter: UNKNOWN
Title: (DAE SIMULATION MODE) long title
Users:
Good / Raw Frames: 0 / 0
Current / Total: 0 / 0
Monitor Counts: 0
Inst. Time: 08/09/2016 11:48:21
Run Time: 0 s
Period: 1 / 1
Current user: NONE
Motors are STATIONARY
Stop All
Alarms (8)
Beam Status
DAE
Device Screens
Experiment Details
IOC Log (18)
Log Plotter
Motors
Scripting
Synoptic
Web Links
Synoptic Selection
Ticket1499_DEMO_OPIs
Refresh Synoptic
Synoptic Navigation
LAKESHORE
LAKESHORE
Current Style
Proposed Style
Current Style
Proposed Style
100%
Lakeshore 336 Temperature Controller
Current Style
Output 1
Output 2
Graph All T
Measurements
Graph Measurements
OUTPUT 1
SETPOINT: 100.000 K
100.000 K
TEMP IN WINDOW:
RAMP:
ON/OFF
RAMP RATE: 2.000 K/min
2.000 K/min
Output 1 Settings...
CONTROL INPUT: Measurement A
TEMP: 2.600
NAME: Default Name
ALARM:
Measurement A Settings...
P: 1.0
1.0
I: 2.0
2.0
D: 3.0
3.0
OUTPUT MODE:
Closed Loop PID
HEATER OUTPUT: 50.0 %
HEATER STATUS: Short/Compliance
RESET
Current configuration: Ticket1499_DEMO_OPIs


IBEX
IBEX Configuration Synoptic IOC Run-Control Help
NDW1298 is SETUP
Run: 00000001
Shutter: UNKNOWN
Title: (DAE SIMULATION MODE) long title
Users:
Good / Raw Frames: 0 / 0
Current / Total: 0 / 0
Monitor Counts: 0
Inst. Time: 08/09/2016 11:49:34
Run Time: 0 s
Period: 1 / 1
Current user: NONE
Motors are STATIONARY
Stop All
Alarms (8)
Beam Status
DAE
Device Screens
Experiment Details
IOC Log (19)
Log Plotter
Motors
Scripting
Synoptic
Web Links
Synoptic Selection
Ticket1499_DEMO_OPIs
Refresh Synoptic
Synoptic Navigation
LAKESHORE
LAKESHORE
Current Style
Proposed Style
Current Style - Output 1
Proposed Style
100%
Lakeshore 336 Temperature Controller
Proposed Style
Output 1
Output 2
Graph All T
Measurements
Graph Measurements
Operation
Setpoint: 100.000 K
100.000 K
Temp in window:
Ramp:
On/Off
Ramp rate: 2.000 K/min
2.000 K/min
Control input: Measurement A
Measurement
Temperature: 9.900 K
Name: Default Name
Alarm:
Measurement A Settings...
Control Parameters
P: 1.0
1.0
I: 2.0
2.0
D: 3.0
3.0
Output mode: Closed Loop PID
Heater
Ouput: 60.0 %
Status: Short/Compliance
Reset
Output 1 Settings...
Current configuration: Ticket1499_DEMO_OPIs