OAK RIDGE National Laboratory

# Archive

July 2026

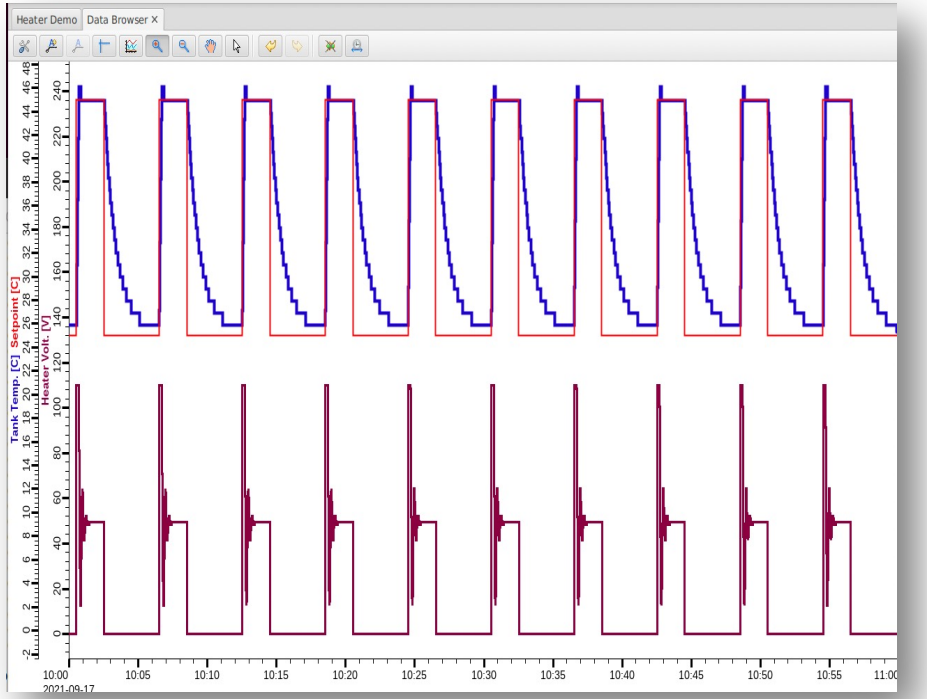
Kay Kasemir

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

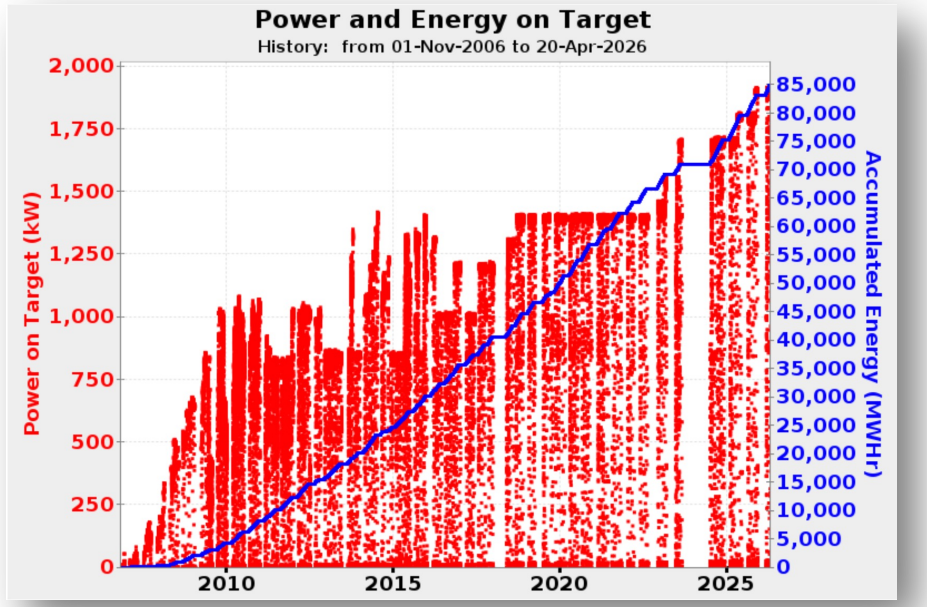
U.S. DEPARTMENT OF ENERGY

# End-User View of Archive

- Databrowser can show history
  - Easy, "Scroll back" on time axis



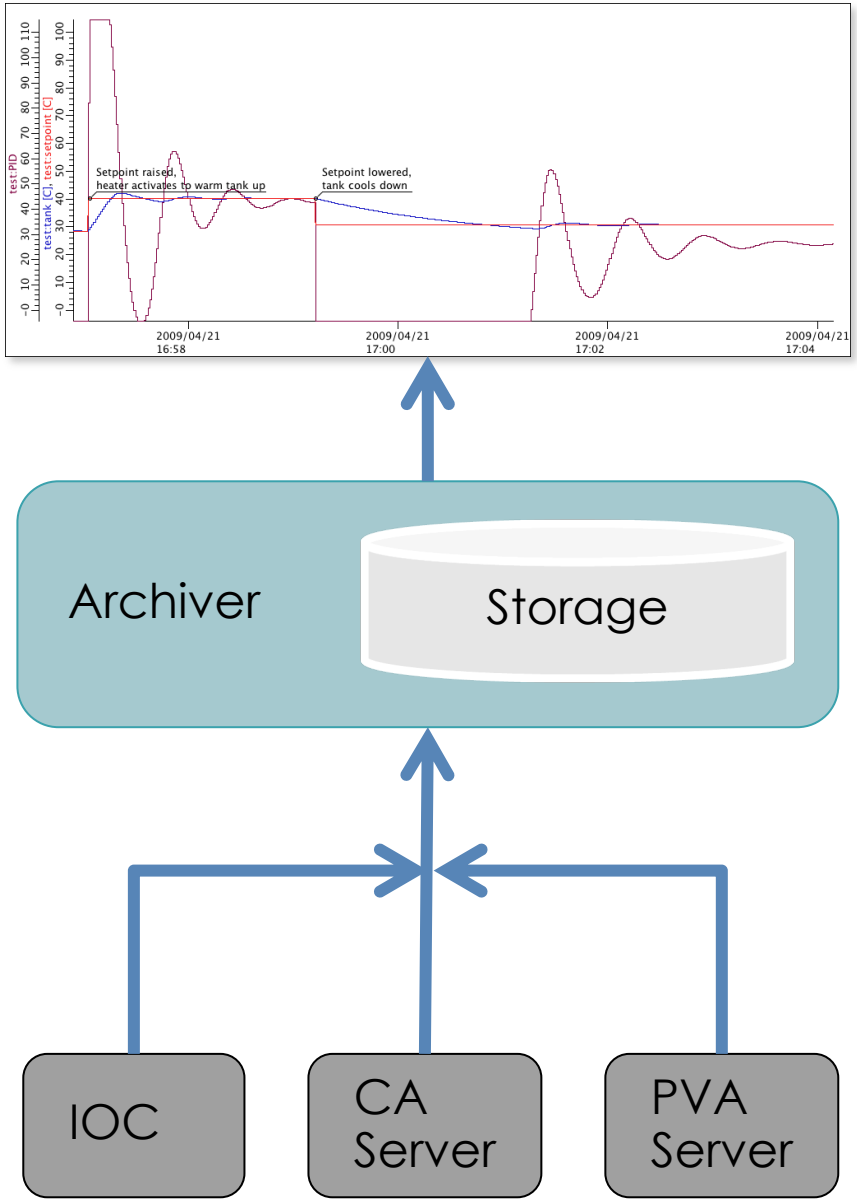

- May access history from Python, Java, … to perform your own analysis
  - Write a program to do that



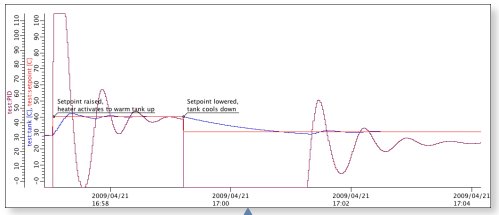
# Basic Technical View of Archive

## Archiver

- Reads PVs via Channel- or PVAccess
- Writes samples to Storage
- Provides history from Storage



# More Detail




Configuration:

Which channels?
How?

Archive Engine (Service)

Read PVs,
write samples to storage

CA, PVA

IOC

Storage

Theory:
*Tons of fast storage*

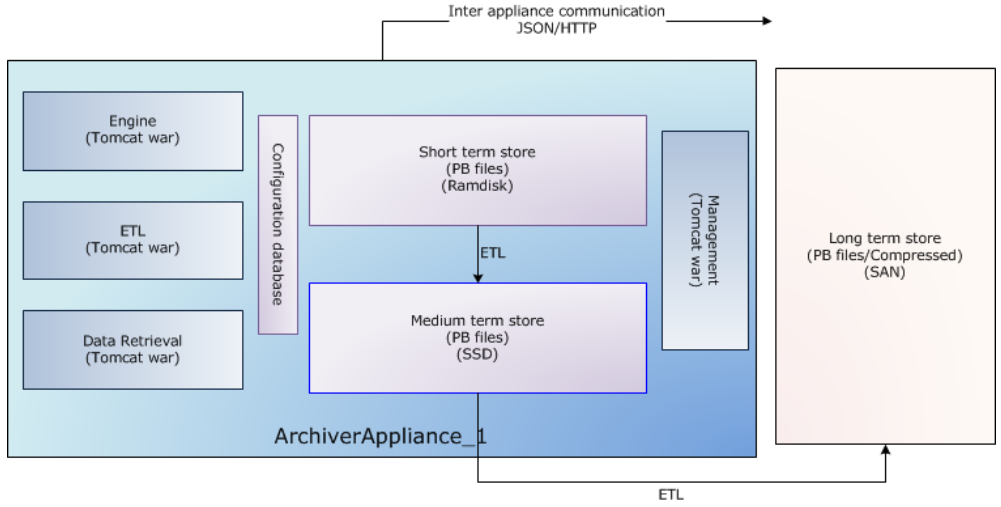
Practice:
Slow,
*limited space*

# Choices…

| | Channel Archiver | Relational Database | TimescaleDB | EPICS Archiver Applicance |
|---|---|---|---|---|
| Storage | Custom "data" and "index" files | MySQL, Postgres, Oracle | Postgres optimized for time series | Custom |
| Storage Speed | Fast | Slow | Better than Slow | Fast |
| Storage Maintainability | Eventually impossible | Trivial (for RDB admin) | Trivial (for RDB admin) | A few files per channel |
| Configuration | XML file | RDB with XML file import/export | RDB with XML file import/export | Web interface, XML file import/export |
| Readout | DataBrowser, custom C++ lib | DataBrowser, any RDB client | DataBrowser, any RDB client | DataBrowser, web interface, web service |

# Archiver Appliance Detail

- Storage: Files with "Protobuf"-encoded samples
  - One file per Channel and Stage
- Example Stages:
  - RAM disk for "today"
  - Solid-state disk for "this month"
  - NFS-mounted folder for "older"
- May create cluster of applicances




- Web interface to add channels, monitor performance, fetch data
- See https://epicsarchiver.readthedocs.io/en/stable for "Quickstart" and more

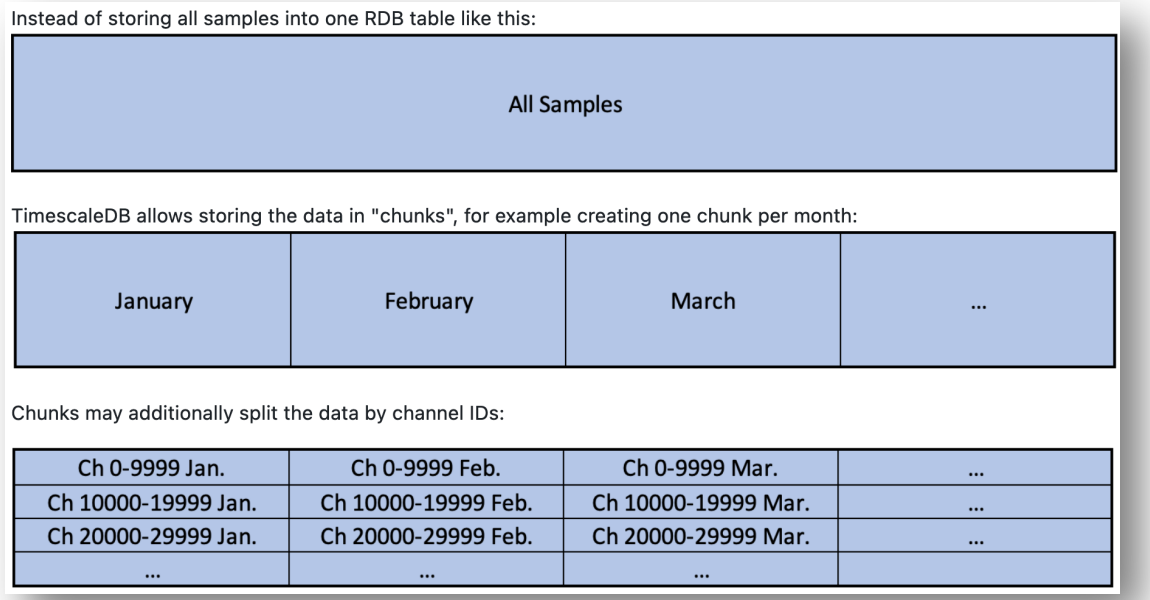
# Relational Database Detail

- Storage: MySQL, Postgres, Oracle
  - Ideal if you can leverage existing RDB cluster and admin support
  - RDBs have been reliable for decades
- Archive Engine, run as Linux service, writes samples to RDB
  - May run one for “Vacuum”, one for “Cryogenics”, one for “Beamlines” etc.
- Data accessible by Data Browser and pretty much any programming language

- See https://github.com/ControlSystemStudio/phoebus/blob/master/services/archive-engine/doc/index.rst

# TimescaleDB Detail

- RDB Admins may develop site-specific ways to "partition" the data and provide stored procedures for optimized data readout
- TimescaleDB = Postgres with extensions to automatically partition time series data




- https://github.com/ControlSystemStudio/phoebus/blob/master/app/data browser-timescale/README.md

# Decisions…

- Archiver appliance
  - Safe option, used by many in EPICS community
  - You'll need somebody to maintain the data
- RDB
  - Very dependable, ideal if you already have RDB admin support and want to access the data in various ways
- TimescaleDB
  - New, no operational experience
  - Promising option for new RDB setup, headstart over site-specific partinioning solutions

# IOC Configuration

Archive engines subscribe to “archive” aka “log” events (DBE_ARCHIVE)

```
camonitor -m l the_pv_name
```

For analog records, configure the ADEL field*

- Unfortunately, not perfect for data with logarithmic behavior like vacuum pressures.

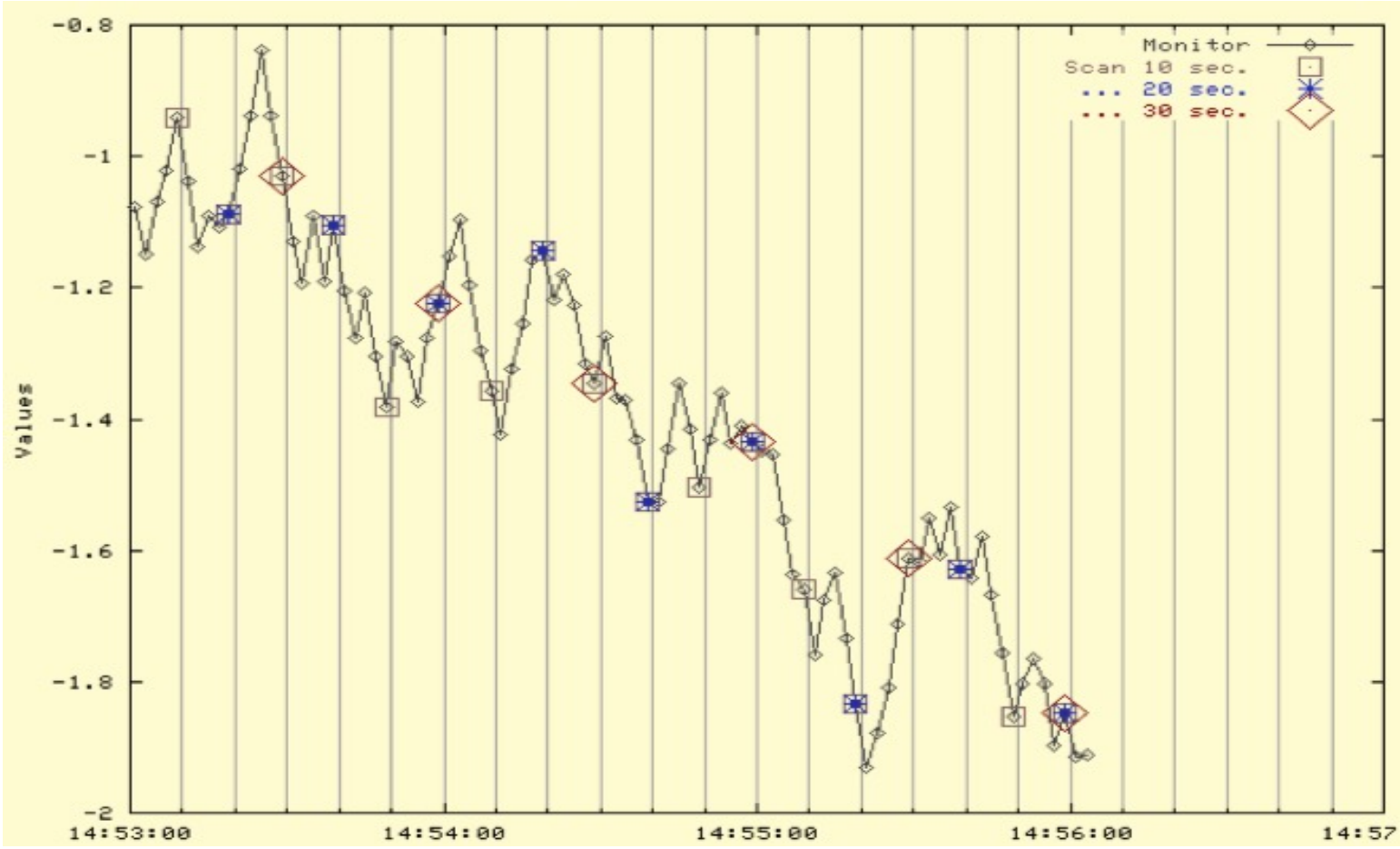
Assert that your IOC knows the time

- Check time stamps reported by `camonitor`

* Also: EGU, PREC, ZNAM, ONAM, …
If it’s worth archiving, it should be properly configured.

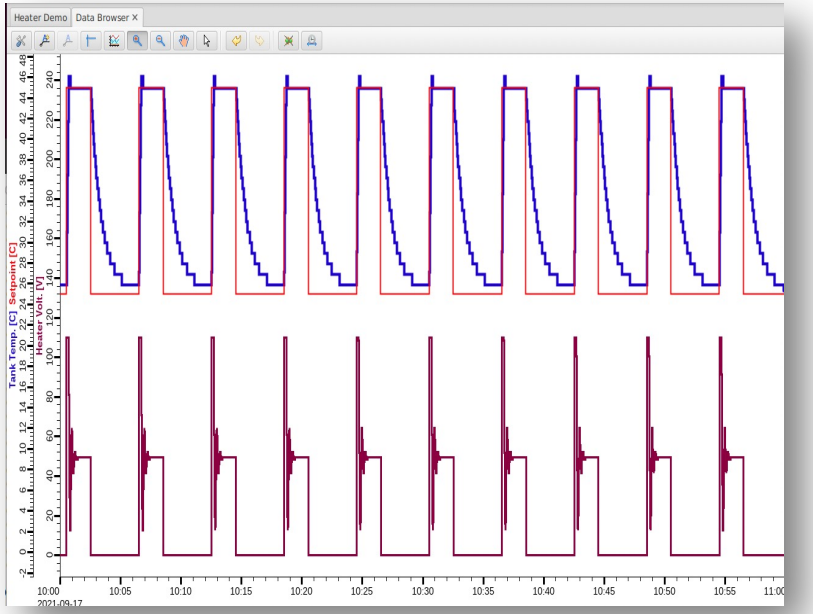
# Archive Engine Options

- Monitor
  - Tries to save every received update
  - Based on ADEL
  - Might need an expected-update-period to allocate buffers, skipping samples if there are too many
- Scan
  - Writes the most recent value every N seconds
  - Stores original time stamps



# Viewing Archived Data

## Open Data Browser, add PV, zoom/pan/set time range




Relies on Data Browser Preferences:

```
org.csstudio.trends.databrowser3/urls=jdbc:mysql://localhost/archive|RDB
org.csstudio.trends.databrowser3/archives=jdbc:mysql://localhost/archive|RDB
org.csstudio.trends.databrowser3/use_default_archives=true
org.phoebus.archive.reader.rdb/user=report
org.phoebus.archive.reader.rdb/password=$report
```

# Archive

- Fundamentally simple: Store values of PVs
- There is no perfect implementation
  - Can't store everything at high rate forever
- On IOC side, configure ADEL (and EGU, PREC, ZNAM, ONAM)
- Looking at data in Data Browser is easy

# RDB example in training setup

# Initial RDB Archive Installation

See https://github.com/ControlSystemStudio/phoebus/blob/master/services/archive-engine/doc/index.rst

1. Install MySQL or MariaDB, PostgreSQL, Oracle
2. Setup archive tables

Check that this happened:

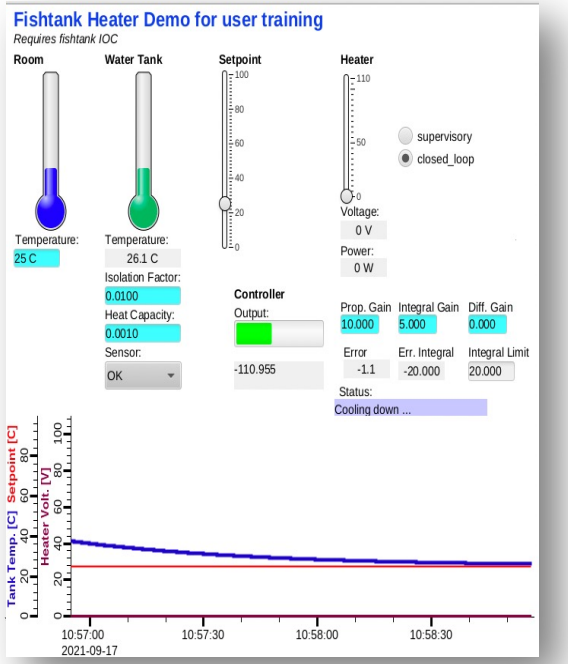
```
sudo systemctl start  mariadb.service
sudo systemctl status mariadb.service
archive-engine -list
```

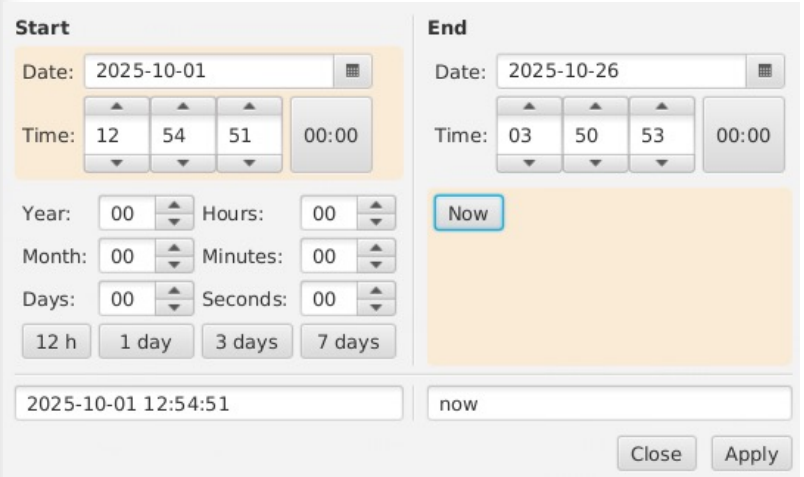
# Fishtank Example

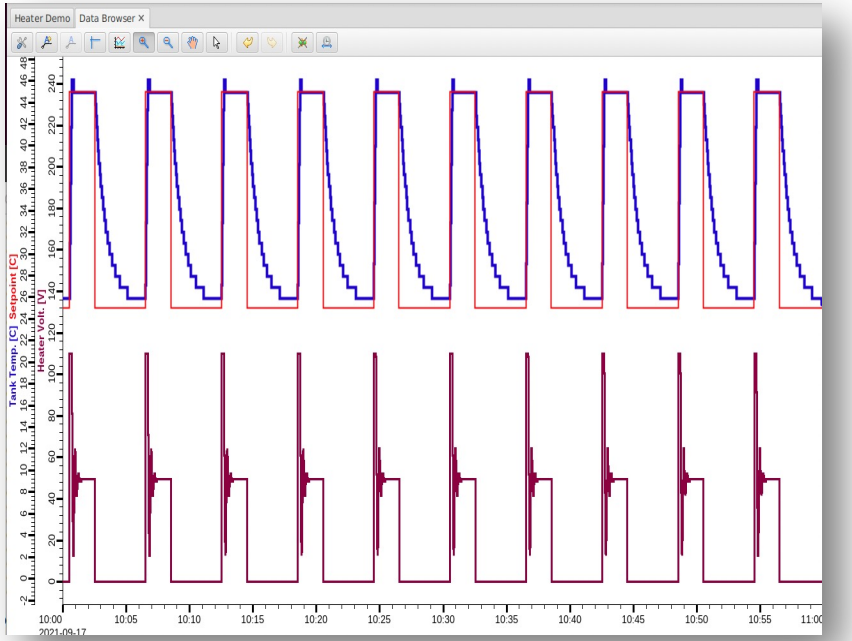
- Run IOC

```
cd /ics/examples/fishtank
./st.cmd
```

- In CS-Studio, open `/ics/examples/fishtank/heater.bob` ➔ Current data
- Right-click on plot, “Open Data Browser”
  - Right-click to Show Toolbar
  - Change time range ➔ Archived data!



# Example archive for ‘fishtank’

```
cd /ics/examples/14_archive/

archive-engine -help

archive-engine -list

cat fishtank.xml
```

Import configuration and start sample engine:

```
archive-engine -engine fishtank -import `pwd`/fishtank.xml

archive-engine -engine fishtank
```

Check http://localhost:4812
in web browser

```
lynx -dump http://localhost:4812/main
lynx -dump http://localhost:4812/stop
```

## Archive Engine

| Summary | |
|---|---|
| **Version** | 4.0.0 |
| **Description** | Archive Engine |
| **State** | RUNNING |
| **Start Time** | 2025-10-15 09:44:25.418102610 |
| **Uptime** | 00:01:04 |
| **Groups** | 1 |
| **Channels** | 3 |
| **Batch Size** | 500 samples |
| **Write Period** | 30 sec |
| **Write State** | OK |
| **Last Written** | 2025-10-15 09:45:25.430333264 |
| **Write Count** | 4 samples |
| **Write Duration** | 0.0 sec |
| **Idle Time** | 100.0 % |
| **Memory** | 144.0 MB of 2244.0 MB used (6.4 %) |

-Main- -Groups- -Disconnected- -Version-

## Archive Engine Group Demo

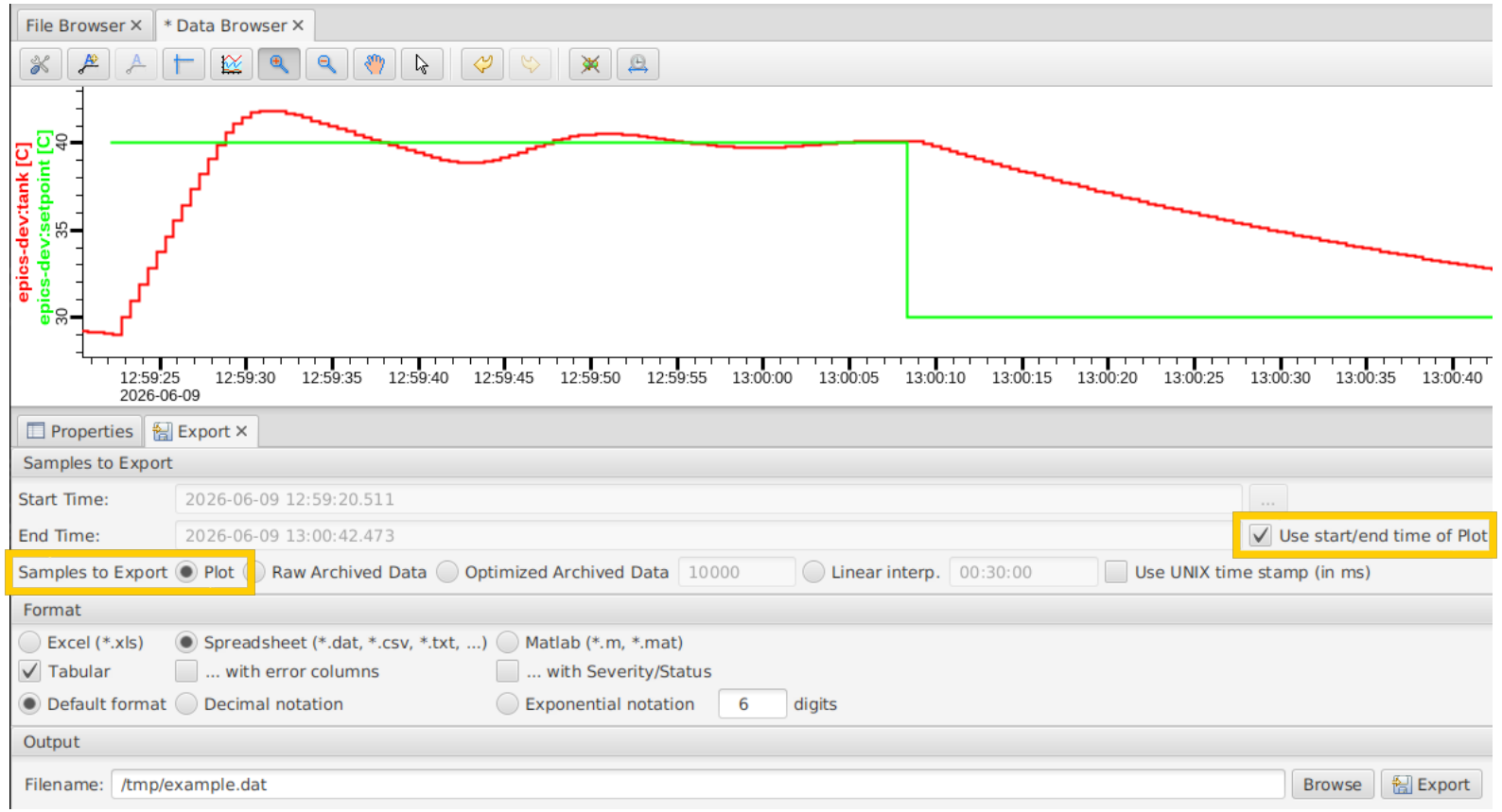
**Status**

**State** Enabled

### Channels

| Channel | Connected | Mechanism | Current Value | Last Archived Value | Received Values | Queue Len. | Queue Avg. | Queue Max. | Capacity | Overruns |
|---|---|---|---|---|---|---|---|---|---|---|
| epics-dev:heat_V | Connected | on delta [00:00:00.100, 1.0] | 2025-10-15 09:44:40.446930576 0.0 [NONE/NONE] | 2025-10-15 09:44:40.446930576 0.0 [NONE/NONE] | 1 | 0 | 0.7 | 1 | 600 | 0 |
| epics-dev:setpoint | Connected | on change [00:00:00.100] | 2025-10-15 09:44:09.719618145 25.0 [NONE/NONE] | 2025-10-15 09:44:25.434099735 25.0 [NONE/NONE] | 1 | 0 | 0.7 | 1 | 600 | 0 |
| epics-dev:tank | Connected | on delta [00:00:00.100, 1.0] | 2025-10-15 09:46:43.447074926 25.914184745877858 [NONE/NONE] | 2025-10-15 09:46:16.447000830 26.573017166288064 [NONE/NONE] | 247 | 0 | 2.8 | 4 | 600 | 0 |

# If there is no archive setup: Data Browser




- Run data browser as usual
- Export "Plot" data
- File looks just like data exported from archive

.. Workaround to capture a few hours of data, maybe over night

# If there is no archive setup: camonitor

- Capture output of camonitor

```
$ camonitor epics-dev:tank | tee example.log
epics-dev:tank                 2026-06-09 13:04:33.782281 39.1624
epics-dev:tank                 2026-06-09 13:04:34.282304 39.0474
…
```

- Show with for example Gnuplot. (`sudo dnf install gnuplot`)