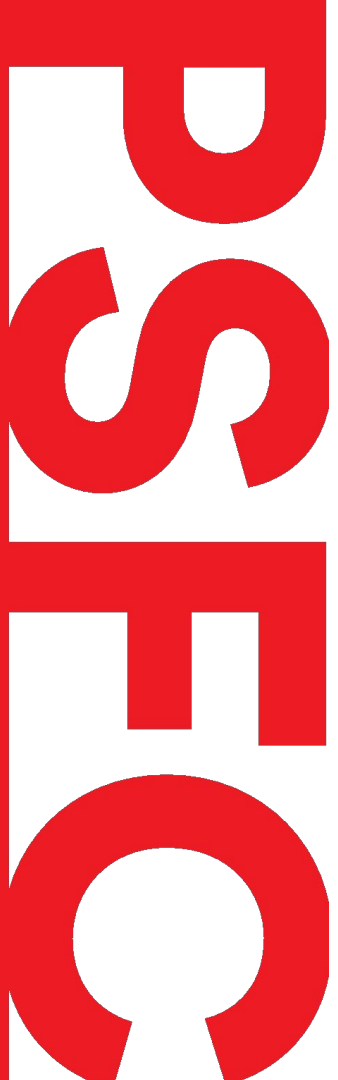# A more flexible design for MDSplus Device Drivers

Fernando Santoro
Stephen Lane-Walsh

Objective and Overview

Build MDSplus devices that are:

- Self-configured
- Reflects a given hardware or a system
- The only constraint should be the hardware, not the MDSplus device code

MDSplus Devices

- Groups of nodes with the ability to invoke methods
  - These methods are often used to communicate with hardware and collect data
- Represent a physical device or abstract system
  - Historically, it has been a *rigid* representation of a specific model of a device
  - e.g. ACQ216, ACA800, J221
- All nodes are present when adding the device
- Afterwards, the structure of the device is immutable

New Design

- When the device is added, *only* add the bare minimum set of nodes

- First, ensure that the Tree is open for edit

- Then, call a configure method that will:

  - Query the available functionality

  - Add/Remove nodes to represent this functionality

- At any point, you can reconfigure

- As a bonus, add a validation method

Example

- D-Tacq 2106 Digitizer
  - Highly customizable
  - We will represent the physical hardware and its mode of operation
    - e.g. Streaming, Transient
  - We will configure the settings for the device
    - e.g. Frequency, Trigger
  - Record data

D-Acq 2106 Digitizer - Setup

Fill in the minimum set of nodes:

- Set `ADDRESS` to the IP Address/DNS
  - e.g. "192.168.0.123"
  - Needed to communicate with the device
- Set `EPICS_NAME` to the EPICS name of the device
  - e.g. "acq2106_123"
  - Needed for some D-Tacq API calls
- Choose operational `MODE`
  - e.g "STREAM" or "TRANSIENT"
  - Needed to determine which nodes to add
- `MODULES` will be filled automatically

D-Acq 2106 Digitizer - Configuring

Method: configure(args...)

- Look at `MODE` and determine what settings are needed
- Contact the hardware and determine what features are available
- Based on those settings and features:
  - Determine which nodes need to be added, and add them
  - Determine which nodes are now extraneous, and delete/deactivate them
- By default, this method will turn nodes off instead of deleting them
  - This will preserve your settings if you want to toggle between different modes
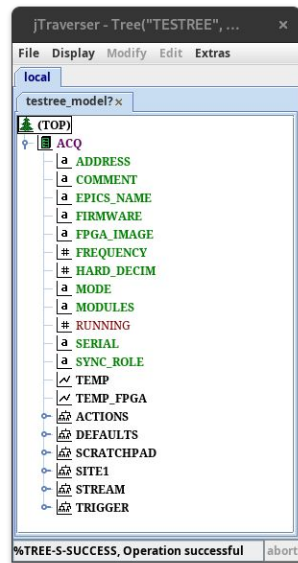 - You can pass the `delete_nodes=True` argument to actually delete them

D-Acq 2106 Digitizer - Setup/Configuring
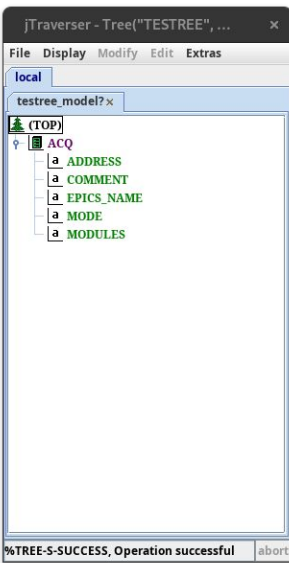
```
import MDSplus
from ACQ2106 import ACQ2106
tree = MDSplus.Tree('TESTREE', -1, 'EDIT')
ACQ2106.Add(tree, 'ACQ')

tree.ACQ.MODE.record = "STREAM"
tree.ACQ.ADDRESS.record = "192.168.0.100"
tree.ACQ.EPICS_NAME.record = "acq2106_123"

tree.ACQ.configure()
```
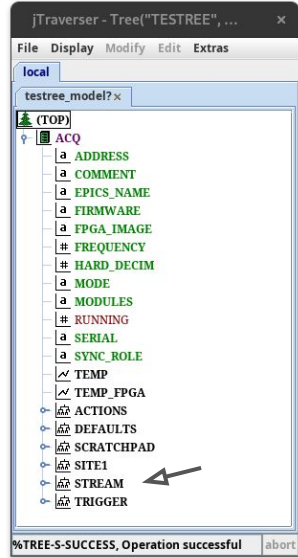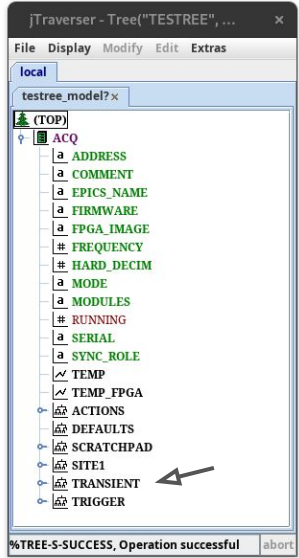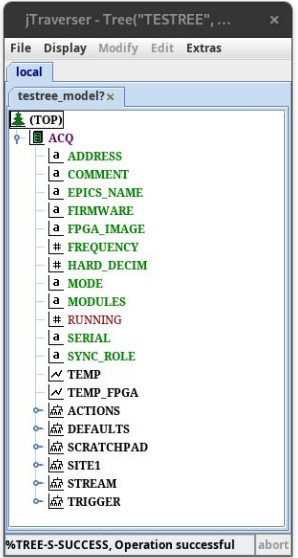
D-Acq 2106 Digitizer - Reconfiguring

```
tree.ACQ.MODE.record = "TRANSIENT"

tree.ACQ.configure(delete_nodes=True)
```

```
tree.ACQ.MODE.record = "STREAM"

tree.ACQ.configure(delete_nodes=True)
```

D-Acq 2106 Digitizer - Verifying

Method: verify()

```
tree = MDSplus.Tree('TESTREE', -1)
tree.ACQ.verify()
```

- Check communication

- Verify the data in each node:

    ○ The value is between min/max, if specified

    ○ The value is in the list of approved values, if specified

- Check if any properties have changed that require you to reconfigure

    ○ Modules

    ○ Firmware

    ○ FPGA Image

- Will throw an error if it finds any issues

Possible Applications

- A variable set of queries for a database backend

    - e.g. SQL databases, Time Series databases

- Any hardware with an API for querying features and settings

    - e.g. GeniCam compliant cameras

- Any service with an API for querying the available data

We are excited to see what the community does!

Coming Soon™

- Tooltips in jTraverser[2] to give helpful information about nodes
- C++ Devices (Stephen Lane-Walsh)
- MDSplus + CMake (Stephen Lane-Walsh)
- mdsplus_slim (Stephen Lane-Walsh)
  - Come see us on Friday for a demo!

MODULES

full path: \TESTREE::TOP:ACQ:MODULES (361)

Status: on, parent is on, compress on put, no write shot

Data: There is no data stored for this node

Comma-separated list of modules, specified by model name, or empty if no module is present. This is queried from the device.

Thank You!

This work was funded under DOE cooperative agreement DE-SC0012470.