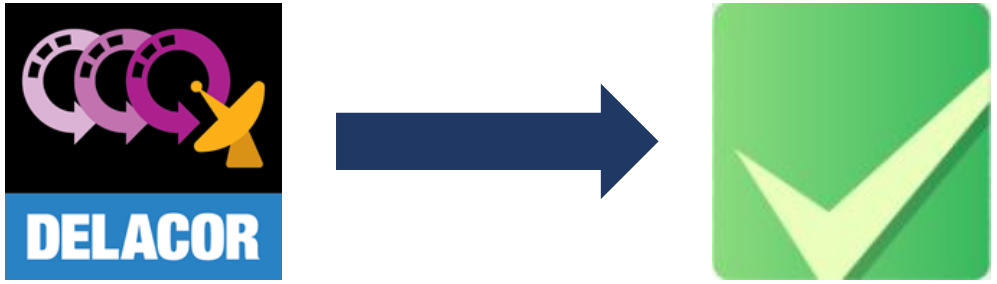
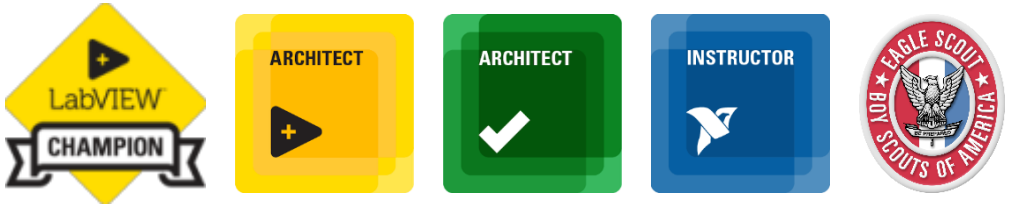


DQMH In TestStand

Sam Roundy

~jiggawax~



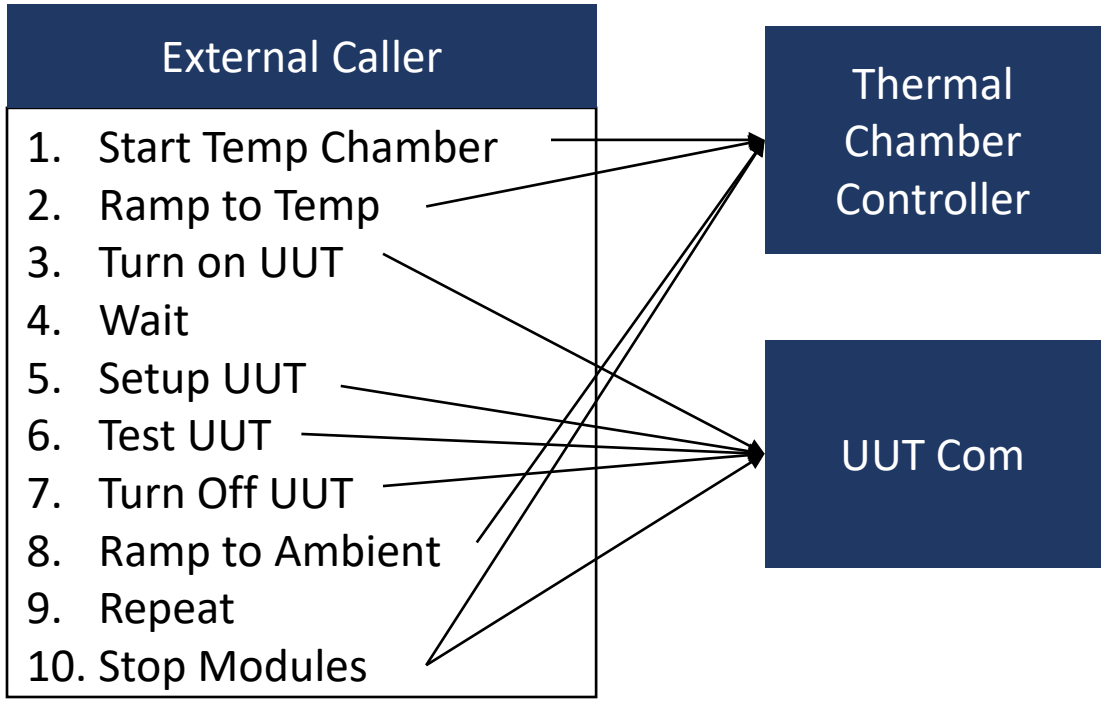
Founder

Agenda

- Brief Overview Of DQMH
  - Module Types
  - Events
- The role of TestStand
- Handling Events
  - Request
  - Broadcast
  - Request and Wait for Reply
- Debugging
- Demo
- Deployment

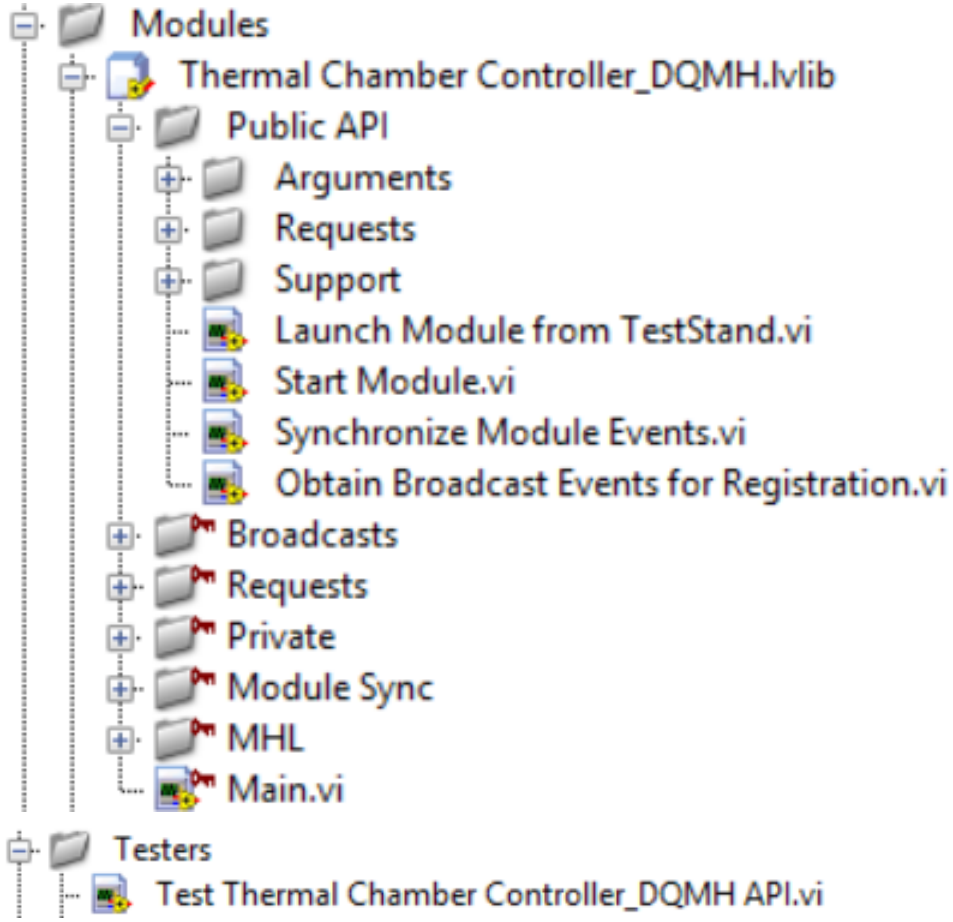
DQMH Modules

- A group of VIs that are related
- Own memory space
  - Can store data
- Easily reusable
- Uses Queued (silent) Messages to communicate
- External Callers can call modules at any point



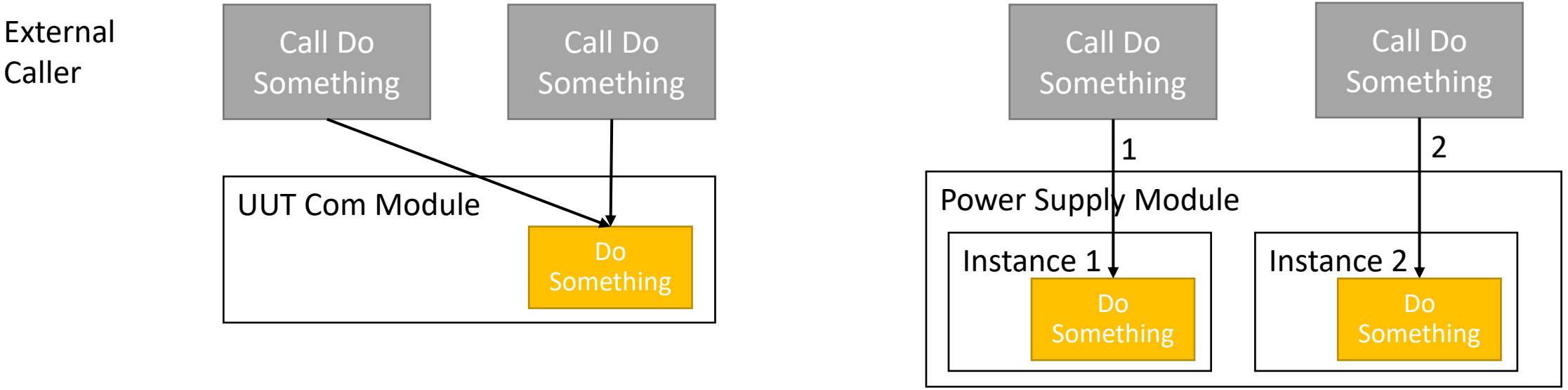
A look inside of a module

- Main.vi
  - Basically the brains for the module
- Public API
  - VIs that tell the module to do something
  - Start Module, Obtain Broadcast Events, Do Something, Show Panel, Hide Panel, Etc…
- Private VIs
  - Anything that helps the module that is not exposed to callers
- Testers
  - Technically not part of the module
  - VERY IMPORTANT!
  - Mimics external caller



2+ Types of Modules

Singleton	Cloneable
• Can only have 1 instance in memory at a time • Any call to the Public API will be to the single instance	• Can have multiple instances in memory at a time • Call Public API methods by using a unique id



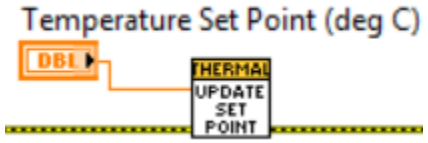
Events

- Ways to get information to and from a DQMH module
- The “API” for DQMH module
- Types of Events:
  - Request
  - Broadcast
  - Request and Wait for Reply
  - Roundtrip

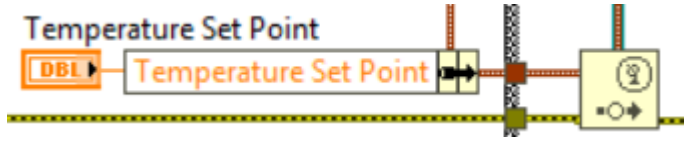
Request Event

Request – External Caller tells DQMH Module to do something

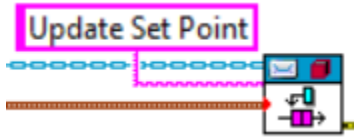
1- Caller Sends Request



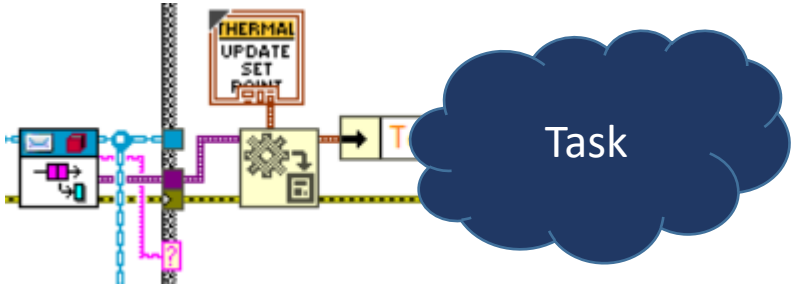
2- Public API Triggers Event



3- Main.vi Event Handler Enqueues



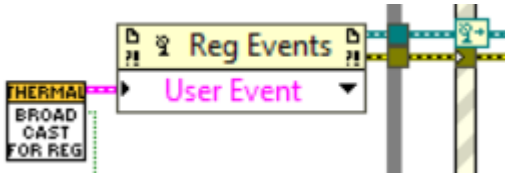
4- Main.vi Message Handler Loop Dequeues and performs task



Broadcast Event

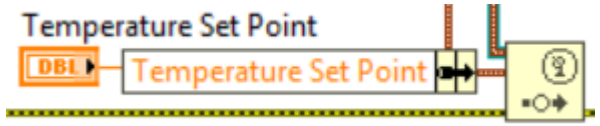
Broadcast – DQMH Module gets data to the External Caller

1- Caller Registers for Events

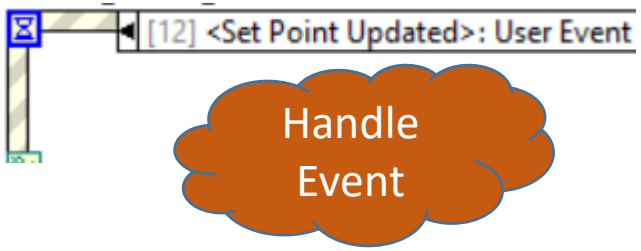


2- Main.vi calls broadcast vi at some point

3- Broadcast vi triggers event with message



4- Caller handles event

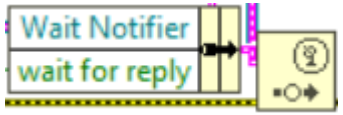


Request and Wait For Reply Event

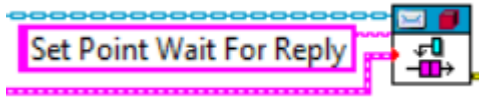
Request and Wait For Reply – External Caller requests information from DQMH Module

1- Caller calls Public API

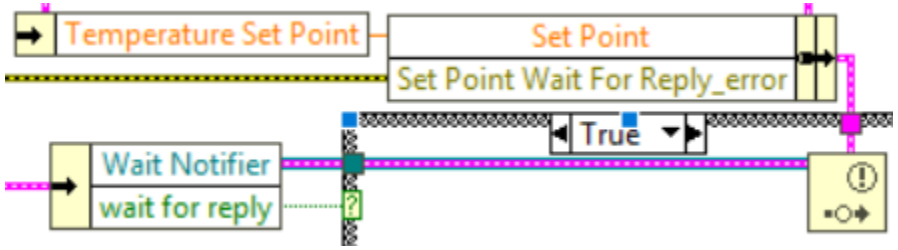
2- Public API triggers event



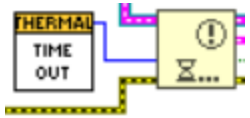
3- Main.vi Event Handler Enqueues



4- Main.vi Message Handler does task and triggers notifier



5- Public API waits on Notifier and returns payload



Roundtrip

Request and Wait For Reply – External Caller requests information from DQMH Module

Broadcast – DQMH Module gets data to the External Caller

The role of TestStand

External Caller Layer

BHAG
Big Hairy
Audacious
GUI

DQMH Power
Supply Tester.vi

DQMH Module Layer

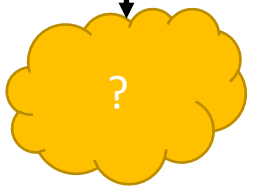
Thermal
Chamber
Controller

UUT Com

Power Supply

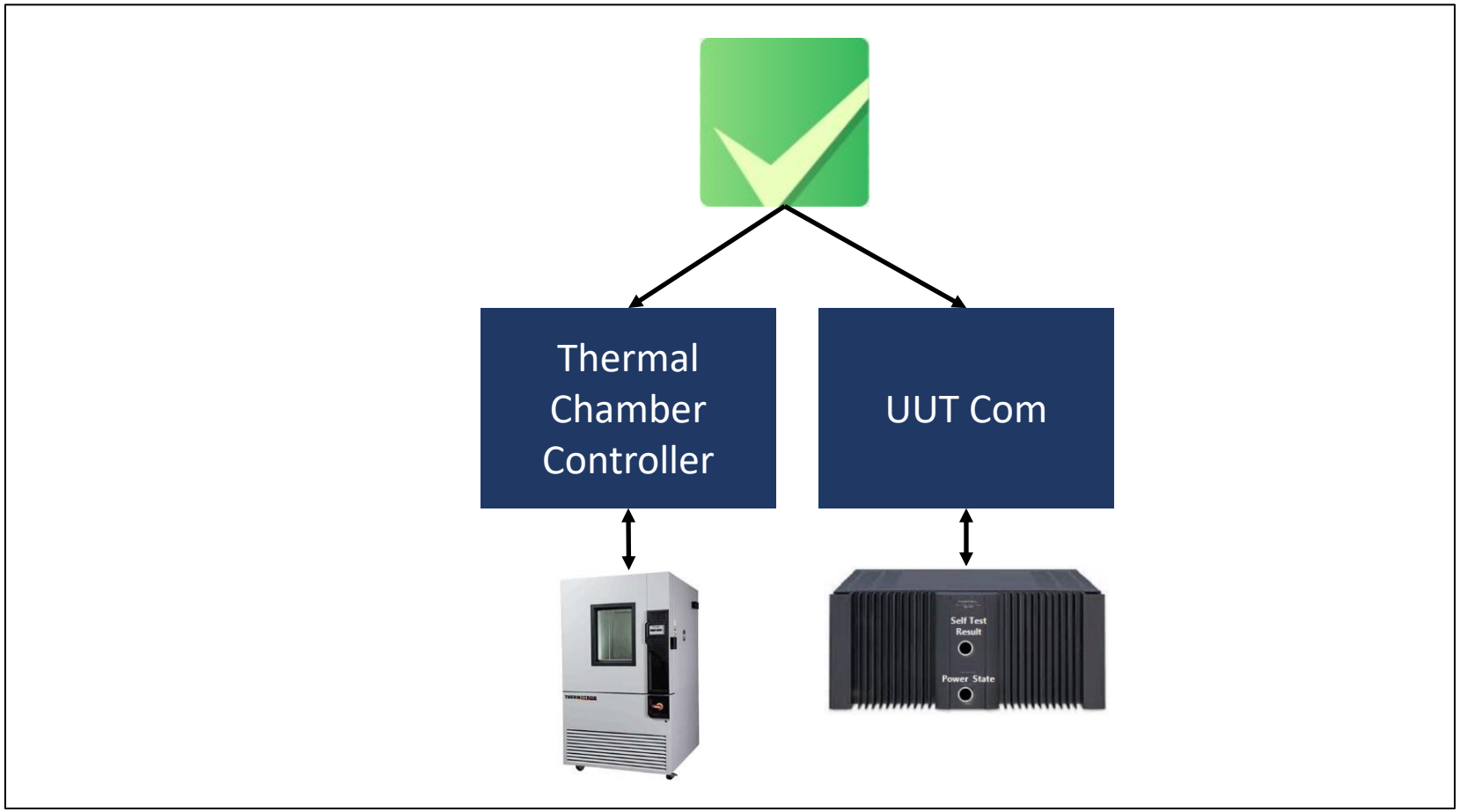
Whatever

Low Level Layer
Hardware Layer



Calling Modules From TestStand

- Requests
- Request and Wait for Reply
- Broadcast

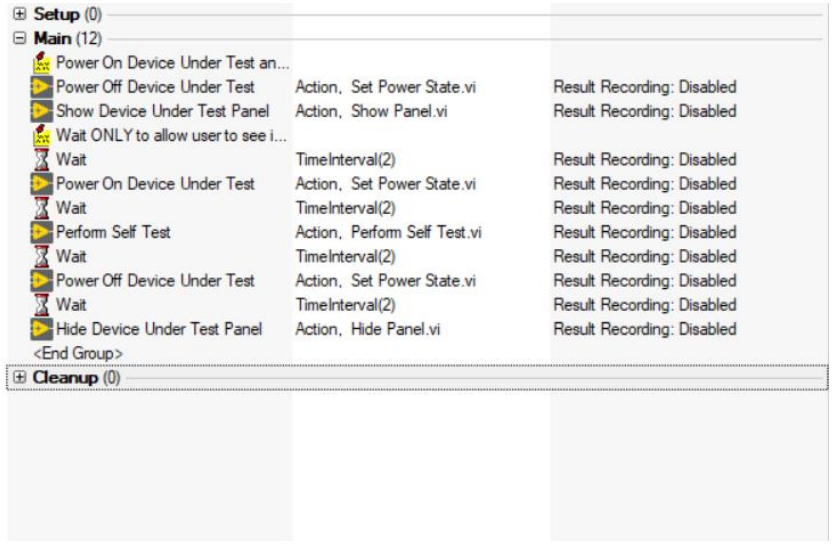


Call Request

- Requests
- Request and Wait for Reply
- Broadcast

- Call as a normal step
- Does not care about acknowledgement

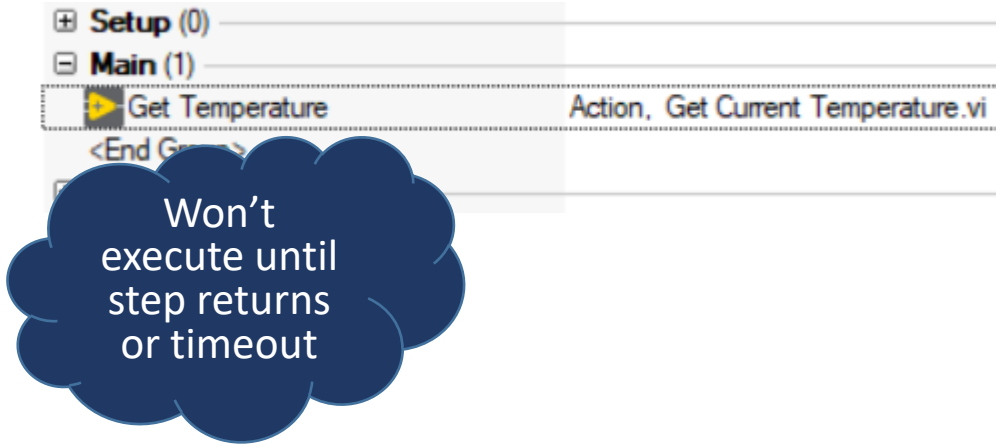
Test UUT



Call Request and Wait for Reply

- Requests
- Request and Wait for Reply
- Broadcast

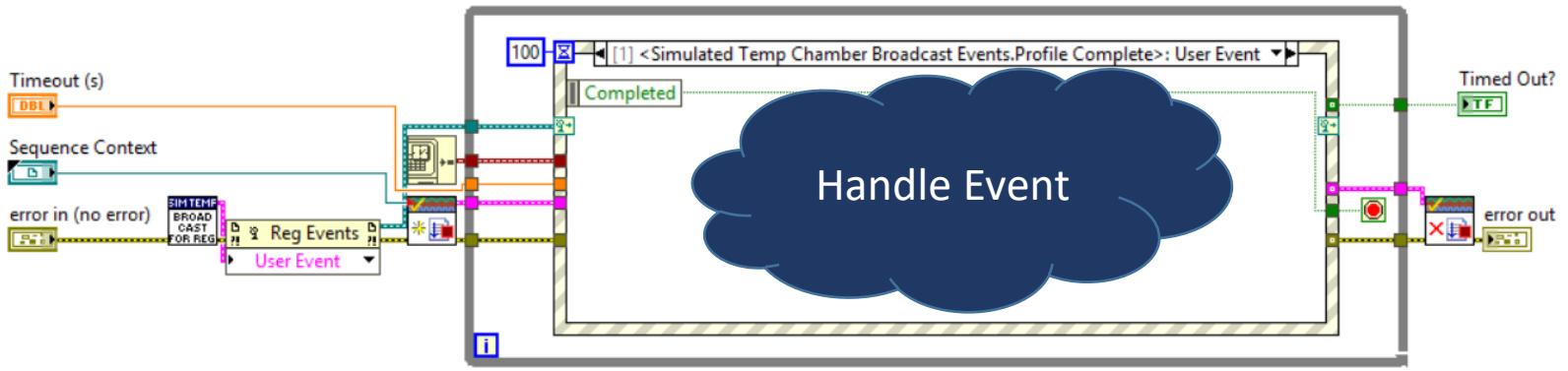
- Call as a normal step
- Will thread lock until complete



Call Broadcast

- Requests
- Request and Wait for Reply
- Broadcast

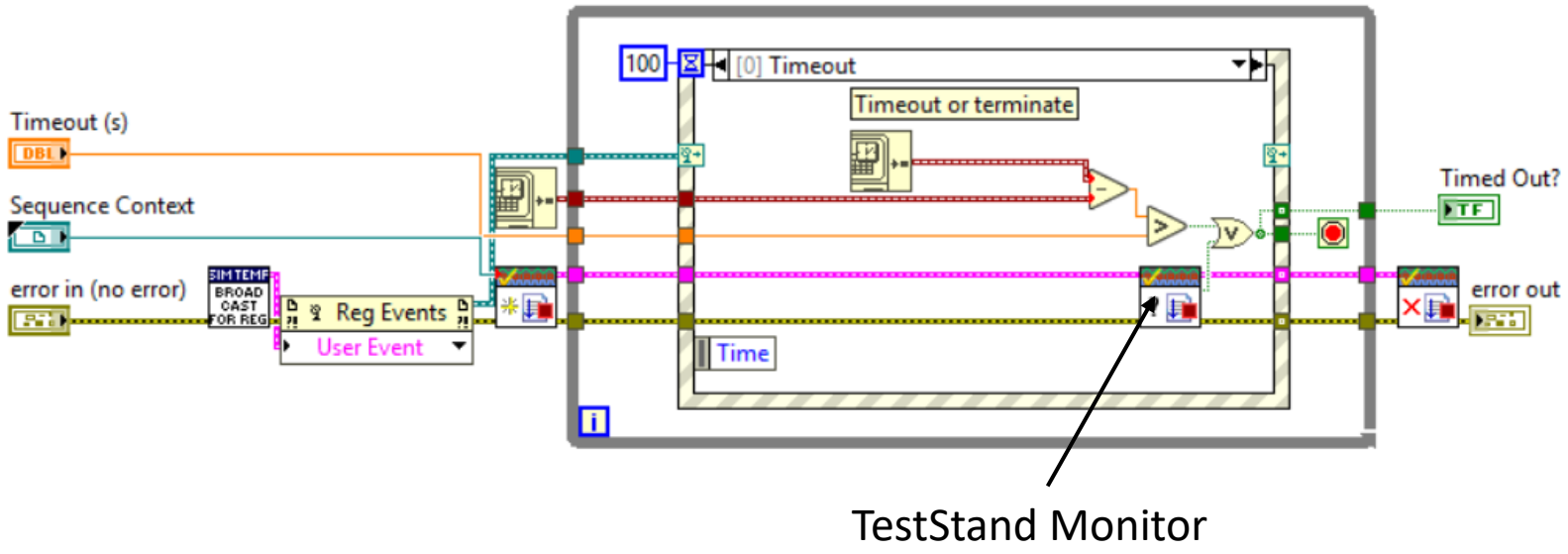
- Create a VI to handle event



Call Broadcast – Exit Strategy

- Requests
- Request and Wait for Reply
- Broadcast

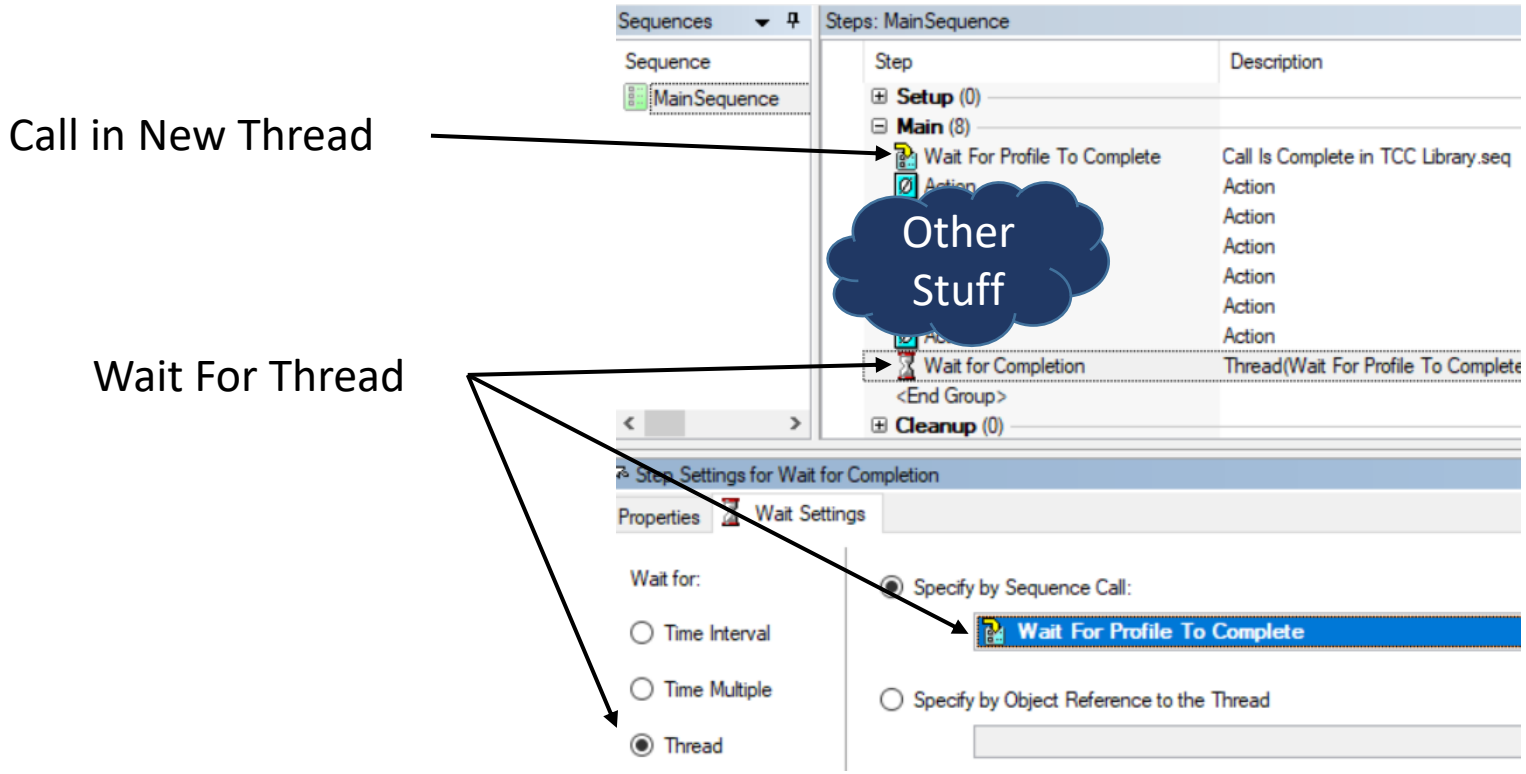
- Need an Exit Strategy – could throw an error
- Use the TestStand Monitor for termination



Call Broadcast – Asynchronous

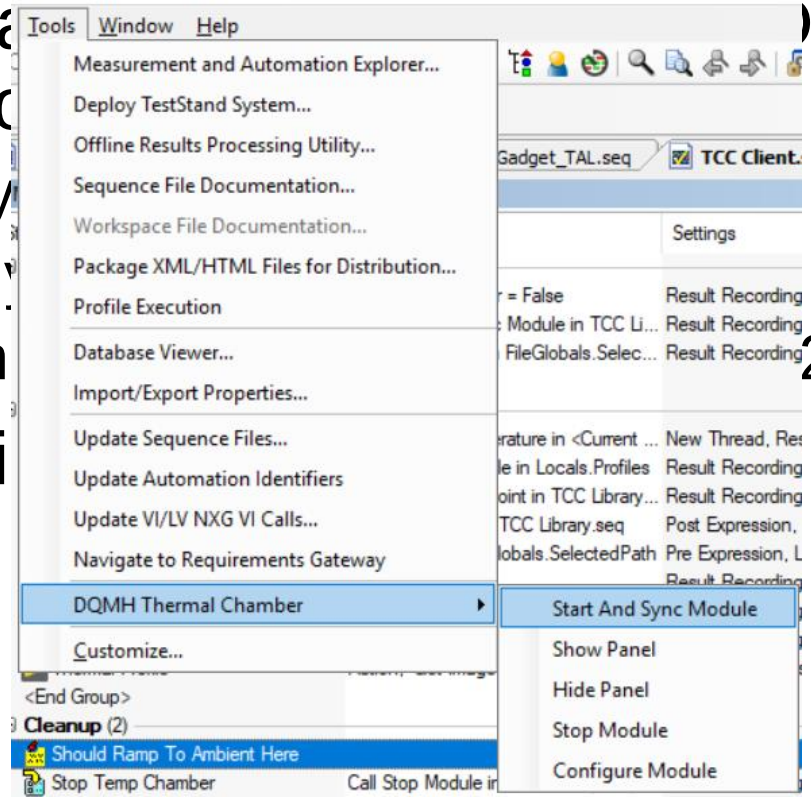
- Requests
- Request and Wait for Reply
- Broadcast

- Need to be asynchronous? Call in new thread



Other Considerations

- Place all DQMH calls into a library (e.g. Temp Chamber Library.seq)
- Do not use any TestSta[illegible]QMH module (unless that is its purpo[illegible]
  - It should be completely[illegible]
  - Can be used by LV onl[illegible]
  - Create a Layer if you n[illegible]2 (e.g. Broadcast)
- Debug Capabilities usi[illegible]

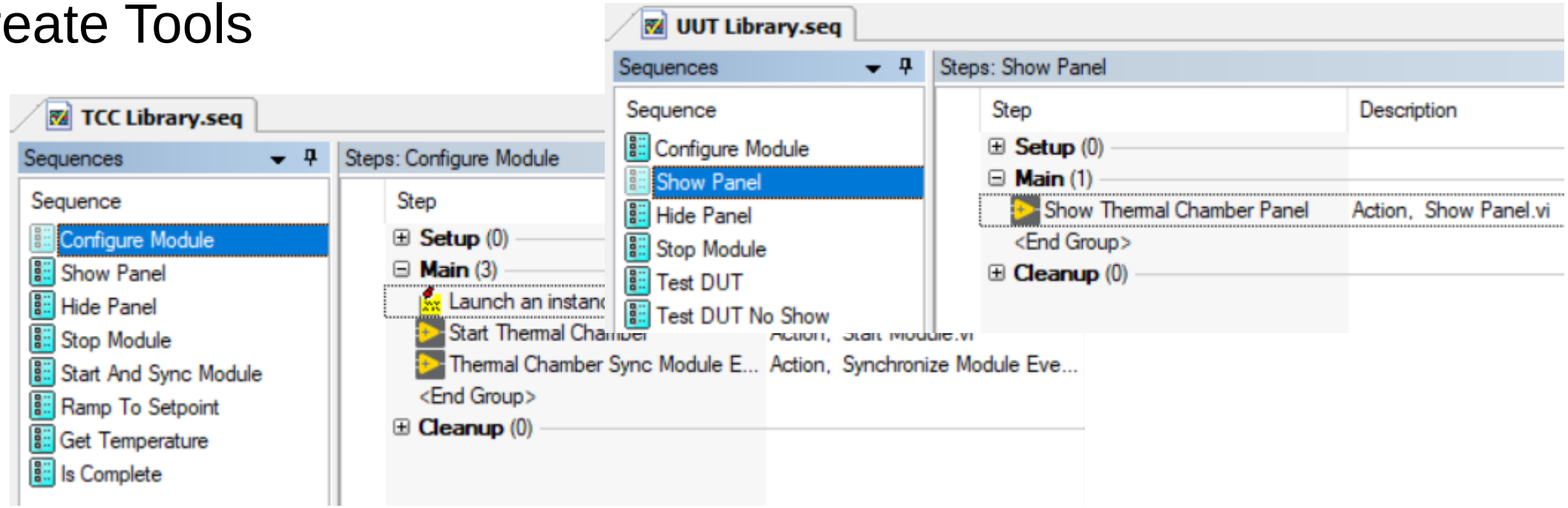


Demo

- Add UUT Tools

DQMH Sequence File Libraries

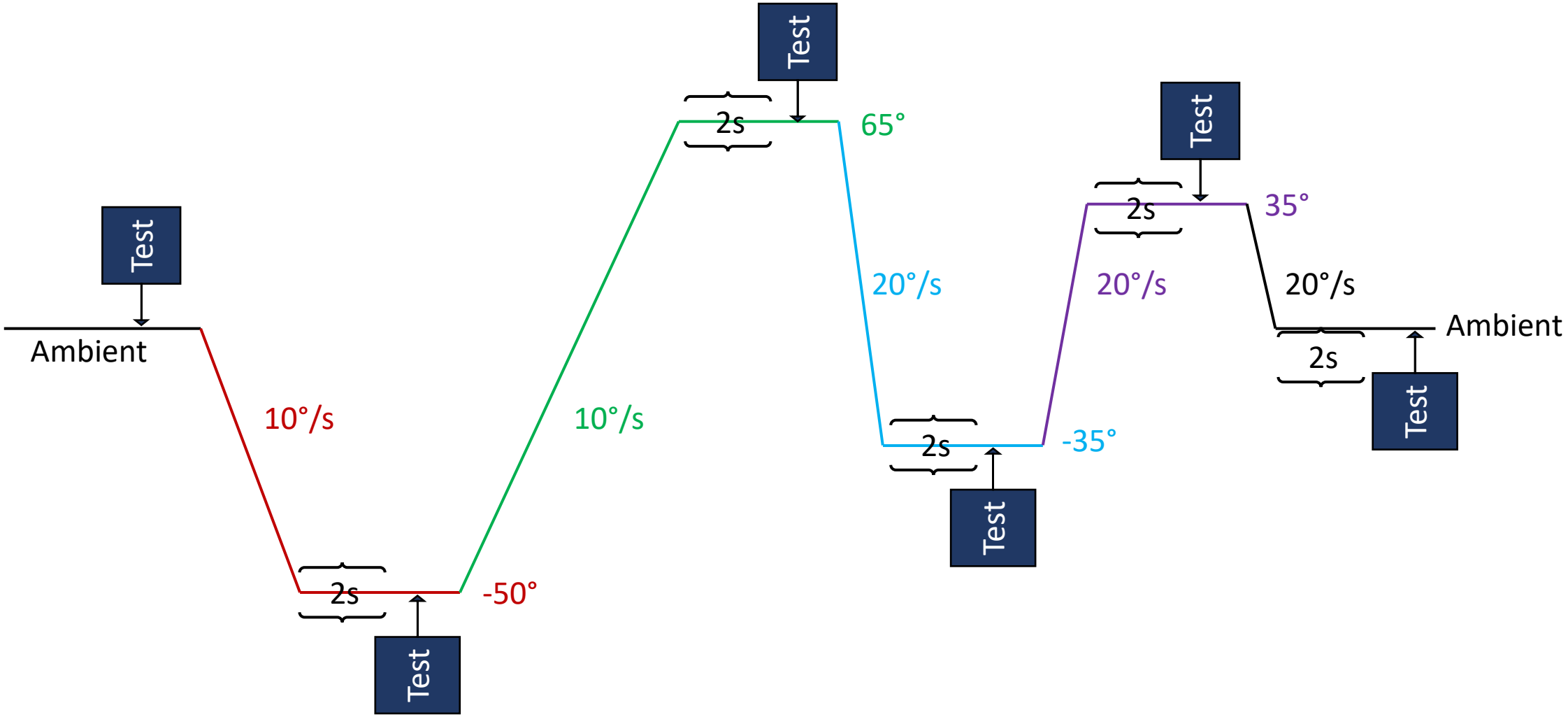
- Only call DQMH from here
- Share between all TestStand code
- Create Tools



Scenario – User Story

- Thermal testing a UUT
- Test at ambient before starting. If fail then don’t test.
- Cycle cold to -50@10deg/sec, hold for 2 seconds then test UUT
- Cycle hot to +60@10deg/sec, hold for 2 seconds then test UUT
- Cycle cold to -35@20deg/sec, hold for 2 seconds then test UUT
- Cycle hot to +35@20deg/sec, hold for 2 seconds then test UUT
- Cycle to ambient@20deg/sec, hold for 2 seconds then test UUT

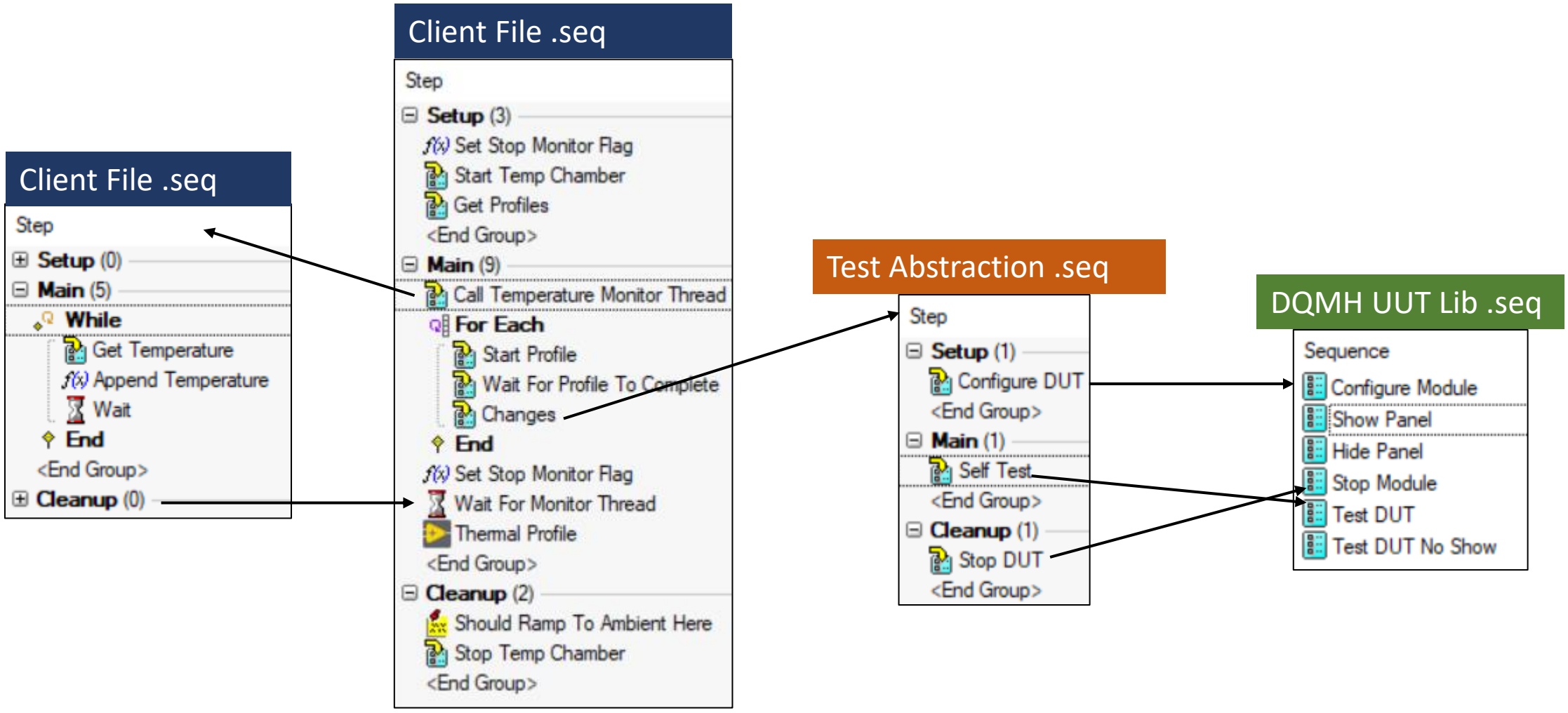
Scenario - Architecture



Scenario - Table

Cycle	Setpoint (°C)	Ramp Rate (°C/s)	Hold Time (s)
Ambient	22	0	0
1 Cold -50	-50	10	2
1 Hot +65	+65	10	2
2 Cold -35	-35	20	2
2 Hot 35	35	20	2
3 Ambient	22	20	2

Thermal Example Architecture



Demo

- Thermal Chamber Demo
- Show DQMH TestStand libraries
- Explain the thermal chamber demo
- Execute
- Use Tools to show and hide panel

Deployment

- Create Packed Project Libraries from your DQMH modules
- Deploy DQMH Library and Module separate from other code