

# LBRUlibs updates.

Krzysztof Furman – 28.09.2021

---

# Unit Tests

- To create and run a unit test:
  - Create a .cxx unit test file in the unittest folder of your plugin
  - Add it to CMakeLists.txt as a unit test
  - When building the program add “--unittest” to have it run unit tests for locally downloaded plugins in sourcecode folder
  - It is not required to perform a “--clean” build to run unit tests (saves a lot of time when debugging them)

```
=====  
RUNNING UNIT TESTS IN /users/furman/DAQ/2.8/build/./lbrulibs/unittest  
=====  
Start of unit test suite "/users/furman/DAQ/2.8/build/./lbrulibs/unittest/ZMQPubSub_test"  
Running 1 test case...  
Entering test module "ZMQPubSub_test"  
/users/furman/DAQ/2.8/sourcecode/lbrulibs/unittest/ZMQPubSub_test.cxx(20): Entering test suite "ZMQPubSub_test"  
/users/furman/DAQ/2.8/sourcecode/lbrulibs/unittest/ZMQPubSub_test.cxx(22): Entering test case "SendReceiveTest"  
/users/furman/DAQ/2.8/sourcecode/lbrulibs/unittest/ZMQPubSub_test.cxx(39): info: check m_publisher_connected has passed  
/users/furman/DAQ/2.8/sourcecode/lbrulibs/unittest/ZMQPubSub_test.cxx(40): info: check m_subscriber_connected has passed  
/users/furman/DAQ/2.8/sourcecode/lbrulibs/unittest/ZMQPubSub_test.cxx(55): info: check rcvcd != 0 has passed  
/users/furman/DAQ/2.8/sourcecode/lbrulibs/unittest/ZMQPubSub_test.cxx(56): info: check msg.size() == 4 has passed  
/users/furman/DAQ/2.8/sourcecode/lbrulibs/unittest/ZMQPubSub_test.cxx(58): info: check received == "TEST" has passed  
/users/furman/DAQ/2.8/sourcecode/lbrulibs/unittest/ZMQPubSub_test.cxx(22): Leaving test case "SendReceiveTest"; testing time: 20004648us  
/users/furman/DAQ/2.8/sourcecode/lbrulibs/unittest/ZMQPubSub_test.cxx(20): Leaving test suite "ZMQPubSub_test"; testing time: 20004687us  
Leaving test module "ZMQPubSub_test"; testing time: 20004721us  
  
*** No errors detected  
End of unit test suite "/users/furman/DAQ/2.8/build/./lbrulibs/unittest/ZMQPubSub_test"  
  
Testing complete for package "./lbrulibs". Ran 1 unit test suites.  
  
Test results are saved in /users/furman/DAQ/2.8/log/unit_tests_Wed_1_Sep_12_59_27_BST_2021.log  
(dbt-pyenv) furman@heppc500:~/DAQ/2.8/sourcecode/lbrulibs/unittest$ █
```

Basic ZMQ Pub-Sub unit test passed.

# Unit Tests – completed so far

- ZMQ Pub-Sub:
  - Fully working with a poller
  - Exact same setup as in our plugin
- ZMQ Stream raw TCP:
  - Fully working with a poller mimicking the setup for Pub-Sub
  - C++ version of library needs you to copy a received message on the sender to get the routing ID towards the receiver (not an issue when you do not have to code the sender yourself)
- PACMAN Frame:
  - Fully working, testing retrieving information from the frame
  - Needed to use a test message to cast into the frame, since get() functions need a message as an argument
  - This design might not be easy to maintain and is not in line with other data formats – worth discussing possibility of changes

# Unit Tests – to be done

- The remaining unit test is for the ZMQLink from our plugin itself:
  - Has to be general (just plugin methods, agnostic to readout socket type)
  - Requires arguments parsed from a json config file
  - Unit tests cannot take file I/O streams so need to replace those arguments somehow
  - Need to investigate how to do it

# Next steps

- Developing unit tests specifically calling ZMQLink:
  - Hard coded arguments/parameters
  - Using what was learnt in general ZMQ unit tests to send test messages to it
- Taking part in the readout repo reorganization
- Looking into the new Grafana dashboards from “Pocket” repo to get some plots

**Thank you**



**Queen Mary**  
**University of London**