Testing and Debugging in Tango Controls

Thomas Ives, Observatory Sciences Ltd, UK

2025-09-20

Outline

Testing

a) pytango DeviceTestContext
b) Test environment with docker compose
c) jive

Debugging

a) Network issues
b) The Tango Black Box
c) QueryEventSystem
d) OpenTelemetry

Follow Along

- Get the `demo.tar.bz2` from:

  https://indico.tango-controls.org/event/479/contributions/919/

```
mkdir demo
tar xjf demo.tar.bz2 -C demo
cd demo
python -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

Testing

a) pytango DeviceTestContext

- DeviceTestContext and MutliDeviceTestContext python contexts provided by pytango
- Useful for creating unit tests for your Tango devices
- Uses the Tango file database, so you can still provide properties

DEMO

```
pytest 01_test_context.py
pytest 02_multi_test_context.py
```

b) Test environment with docker compose

- Startup an entire Tango environment with a single command

DEMO

```
docker compose -f 03_docker_compose.yaml up -d
export TANGO_HOST=localhost:20000
pytest 03_tests.py
```

c) jive

- Tango devices are entirely discoverable -> jive can give you a GUI for free
- Available as an OCI image:

  https://developer.skao.int/projects/ska-tango-images/en/latest/catalogue/ska-tango-images-tango-jive.html

DEMO

```
docker run --security-opt label=type:container_runtime_t \
   --net=host --user $(id -u):$(id -g) \
   -e TANGO_HOST=$TANGO_HOST -e DISPLAY=$DISPLAY -e XAUTHORITY="/Xauthority" \
   -v /tmp/.X11-unix:/tmp/.X11-unix:z -v ${XAUTHORITY:-$HOME/.Xauthority}:/Xauthority:ro \
   --rm artefact.skao.int/ska-tango-images-tango-jive:7.36.5
```

Debugging

a) Network issues

- Each device server uses three TCP ports
  - 1 port used by CORBA
  - 2 ports used by the ZMQ event system
- CORBA port from the device servers IOR, available from `Database.import_devcie`
  - IORs can be parsed at https://ior.jakubchmura.pl/
- ZMQ ports are available from the `DServer.ZmqEventSubscriptionChange` command

DEMO

```
docker compose -f 03_docker_compose.yaml down
docker compose -f 04_network_issues.yaml up -d
sudo ./04_block_ports.sh enable # Linux only
```

b) The Tango Black Box

- Every Tango device records every command and attribute read/write request
  - Date
  - Request
  - Client host/PID
- Only stores the last 50 requests by default
  - Limit can be updated with the `black_box` device property

DEMO

c) QueryEventSystem

- Device servers from cppTango/pytango 10.1 onwards have a `QueryEventSystem` command.
  - Which events are being published?
  - How many events have been sent?
  - Which events are being consumer?
  - How many have been received?
- Provides information about what the event system is doing as a JSON string
- `EnableEventSystemPerfMon` allows turning on the collection of performance data from the event system.

c) QueryEventSystem

DEMO

```
docker compose -f 04_network_issues.yaml down
docker compose -f 05_query_event_system.yaml up -d
sudo ./04_block_ports.sh disable # Linux only
```

d) OpenTelemetry

- Requires cppTango 10 and it must be built with telemetry support, still consider experimental
- Requires an OpenTelemetry collector to accept the traces
- Disabled at runtime by default, enabled and configured with environment variables:
  - `TANGO_TELEMETRY_ENABLE`
  - `TANGO_TELEMETRY_TRACES_EXPORTER`
  - `TANGO_TELEMETRY_TRACES_ENDPOINT`
  - `TANGO_TELEMETRY_LOGS_EXPORTER`
  - `TANGO_TELEMETRY_LOGS_ENDPOINT`

DEMO