

'CHECKUP' - TESTS FOR YOUR CONTROL SYSTEM

JOHAN FORSBERG, AUREO FREITAS (MAX IV),
HANNO PERREY (DVEL)

Created: 2024-05-30 Thu 07:23

1. BACKGROUND

MAX IV is shifting focus to *operation*

How to check if a system is *ready* for experiments

Document human "know-how"

Replace obscure, messy scripts, etc

Complementing alarm systems and visualizations

2. TANGO CHECKUP

Checkup: a general examination of someone's or something's condition

A library for writing *tests* for a control system.

Based on **pytest** and **pytango**

Convenient fixtures, decorators and helper functions

<https://gitlab.com/MaxIV/tango-checkup>

2.1. GOALS

- Concise
- Re-usable
- Flexible
- Usable without (much) programming experience
- Give human friendly output

2.2. WHAT DO WE MEAN BY 'TEST'?

- Clear use case
- Specific condition
- Give advice
- Safe to run at any time (ideally)

"When running experiments, the pressure in chamber A needs to be below $3e-9$ mbar. Otherwise call the beamline manager."

```
@with_attributes("sys/tg_test/1/ampli")
def test_chamber_a_pressure(attribute):
    """The pressure in chamber A is good"""
    assert attribute < 3e-9, "Call the beamline manager!"
```

2.3. WHO IS IT FOR?

Beamline staff preparing for experiments

Support personnel diagnosing problems

Ideally maintained by BL staff

3. CHECKUP TEST RUNNER

A tiny wrapper around pytest

```
$ checkup bl123/test_experiment.py -k pressure --tb=line
```

```
bl123/test_experiment.py F [100%]  
  
===== FAILURES =====  
_____ test_chamber_a_pressure[sys/tg_test/1/ampli] _____  
bl123/test_experiment.py:18: in test_chamber_a_pressure  
    assert attribute < 3e-9, "Call the beamline manager!"  
E   AssertionError: Call the beamline manager!  
E   assert The value of ampli (sys/tg_test/1/ampli) is 4e-09mbar which is greater than (or equal to) the expected 3e-09!  
===== short test summary info =====  
FAILED bl123/test_experiment.py::test_chamber_a_pressure[sys/tg_test/1/ampli] -  
AssertionError: Call the beamline manager!  
===== 1 failed in 0.04s =====
```

4. ASSERTS

Standard way of checking a condition in pytest

Checkup enhances the output of asserts on tango values

```
assert attribute < 9000, "It's over 9000!"  
...  
E   AssertionError: It's over 9000!  
E   assert The value of long_scalar_w  
(sys/tg_test/1/long_scalar_w) is 10000 which is  
greater than (or equal to) the expected 9000!
```


Some specialized check functions:

```
check_tango_host("my.tango.host:10000")
```

```
check_device_state(device_proxy,  
                  DevState.ON,  
                  "Is the power cable connected?")
```

5. PARAMETRIZATION

Running the **same** test function over **many** inputs

Checkup can parametrize across *devices* and *attributes*

```
@with_devices("sys/tg_test/1", "sys/database/2")
def test_device_responds(device):
    assert device.ping() < 100, "Check network connection"
```

The `device` argument is a pytango `DeviceProxy` object

Parametrize over device class

```
@with_devices_of_class("TangoTest")
```

...or using wildcards

```
@with_devices_matching("sys/tg_test/*")
```

Parametrize attributes

```
@with_attributes("sys/tg_test/1/ampli")  
def test_ampli_is_reasonable(attribute):  
    assert attribute <= 3, "Could you try something smaller?"
```

attribute is just a (slightly enhanced) DeviceAttribute.

```
assert attribute.quality != ATTR_ALARM, "Run for the hills!"
```

```
@with_attributes(  
    dict(attribute="sys/tg_test/1/ampli",  
          threshold=3),  
    dict(attribute="sys/tg_test/1/double_scalar",  
          threshold=100)  
)  
def test_some_thresholds(attribute, threshold):  
    assert attribute > threshold, "Not big enough."
```

6. MARKS

Add "metadata" to tests

@critical, @major etc.

Also useful for running a subset of tests in a module.

```
@pytest.mark.branch_a  
def test_something():  
    ...
```

```
$ checkup test_basic.py -m branch_a
```

7. HTML REPORTS

More human friendly

Link to

Convenient to store and share links

8. HTML REPORTS

Veritas Beamline Health Report

Results from *beamline-checkup* run on Tue Mar 26 09:19:30 2024

Critical issues:

None found 👍

Major issues:

! **Check that pressure in mono is ok:** The value of Pressure ([b316a-o02/vac/vgcp-01/Pressure](#)) is 7.522725127273588e-07mbar which is less than (or equal to) the expected 9e-07! Mono pressure is too low, is O2 cleaning turned off?

Potential issues:

⚠️ **Check valves in B branch optics part:** Device [b316a-ob03/vac/vgmb-01](#) is in state CLOSE, expected OPEN. Please check valve [b316a-ob03/vac/vgmb-01](#) in B branch synoptic, try opening it.

⚠️ **Check valves in B branch optics part:** Device [b316a-ob04/vac/vgob-01](#) is in state CLOSE, expected OPEN. Please check valve [b316a-ob04/vac/vgob-01](#) in B branch synoptic, try opening it.

⚠️

9. CHECKUP @ MAX IV

Developing checkup suites with several beamlines

Goal is for BL personnel to maintain their test suites

cscheckup, a suite that checks the state of all our "standard" devices.

10. THE FUTURE?

- Better reports
- More convenience functionality
- Specialization, e.g. Sardana, IcePap...
- Turn it into a pytest plugin?
- Automated running in Gitlab CI

11. SUMMARY

- Checkup tests your control system
- A few lines of code can create tests running over 100s of devices
- <https://gitlab.com/MaxIV/tango-checkup>



12. THANKS!

- Hanno Perrey (DVEL)
- Aureo Freitas (MAX IV)

We're interested to hear what you do at other facilities!