Test Report on the Library dpg_lib

Conventions

Each test is defined following the same format. Each test receives a unique test identifier and a reference to the ID(s) of the requirements it covers (if applicable). The goal of the test is described to clarify what is to be tested. The test steps are described in brief but clear instructions. For each test it is defined what the expected results are for the test to pass. Finally, the test result is given, this can be only pass or fail.

The test format is as follows:

Test Identifier: TEST-[I/A/D/T]-XYZ

Requirement ID(s): REQ-uvw-xyz

Verification method: I/A/D/T

Test goal: Description of what is to be tested

Expected result: What test result is expected for the test to pass

Test steps: Step by step instructions on how to perform the test

Test result: PASS/FAIL

The test ID starts with the fixed prefix 'TEST'. The prefix is followed by a single letter, which defines the test type / verification method. The last part of the ID is a 3-digits *hexadecimal* number (0..9|A..F), with the first digit identifing the module, the second digit identifing a class / function, and the last digit - the test ordering number for this object. E.g. 'TEST-T-112'. Each test type has its own counter, thus 'TEST-T-112' and 'TEST-A-112' tests are different entities, but they refer to the same object (class or function) within the same module.

The verification method for a requirement is given by a single letter according to the table below:

Term	**Definition**
Inspection (I)	Control or visual verification
Analysis (A)	Verification based upon analytical evidences
Test (T)	Verification of quantitative characteristics with quantitative measurement
Demonstration (D)	Verification of operational characteristics without quantitative measurement

Tests definition (Inspection)

Test Identifier: TEST-I-000

Requirement ID(s): REQ-INT-000

Verification method: I

Test goal: Reliable dependencies.

Expected result: If any library outside the Standard Python Library is used, it should be widely accepted, well maintained and not marked for deprecation in the nearest future (2 years).

Test steps: Check / review the source code. Findings: there are no 3rd party dependencies. The library depends on 2 other libraries by the same author, which are maintained.

Test result: PASS

Test Identifier: TEST-I-001

Requirement ID(s): REQ-UDR-000

Verification method: I

Test goal: Library's documentation.

Expected result: The requirements and design documentation, test reports and the user / programmer reference documentation is present. Reference documents provide 'use manuals' sufficient to work with the libary, and the implementation is also documented sufficiently (API reference) to ensure the maintenance and future modification of the library, providing the sufficient proficiency of the user with Python programming language.

Test steps: Check / review the documentation provided with the library. Findings: the library is thoroughly and comprehensively documented.

Test result: PASS

Tests definition (Analysis)

Test Identifier: TEST-A-000

Requirement ID(s): REQ-FUN-000, REQ-FUN-001

Verification method: A

Test goal: Check that the library implements the functionality properly and sufficiently

Expected result: The library provides functions / class methods implementing all required functionality. All required functionality as well as the additional functions / class methods operate properly: they generate the proper response to the proper input data, and raise expected exceptions in response to an improper data input.

Test steps: Check / review the source code. Run all unit-test suits defined for the individual modules; specifically UT001_base_functions, UT002_ordered_functions, UT003_data_classes, UT004_distribution_classes, UT005_special_functions and UT007_stat_tests.

Test result: PASS

Test Identifier: TEST-A-001

Requirement ID(s): REQ-SIO-000, REQ-SIO-001

Verification method: A

Test goal: Check the acceptable input data types, and which data types are returned.

Expected result: Any generic numeric sequence is accepted as the data input:

- the Python built-in data types like **list**, **tuple**
- OR any custom class instance, sub-classing **collections.abc.Sequence**, except for strings (**str**), byte-strings (**bytes**) or byte-arrays (**bytearray**)

as long as all elements are generic numbers. Generic number is:

- any real number (**int** or **float**)
- OR an instance of a class compatible with **phyqus_lib.base_classes.MeasuredValue**, i.e. having *data attributes* (fields or properties) *Value* and *SE* - representing the 'mean' ('most probable') measured value and the associated measurement uncertainty / error

The returned values are of the native Python data types - **int**, **float**, **bool** or **list** / **tuple** of elements of the three mentioned scalar types.

Test steps: Check / review the source code. Run all unit-test suits defined for the individual modules. See the respective module's test reports: TE001, TE002, TE003, TE004 and TE007.

Test result: PASS

Tests definition (Demonstration)

Test Identifier: TEST-D-000

Requirement ID(s): REQ-AWM-000

Verification method: D

Test goal: Descriptive exceptions in the case of improper data input.

Expected result: An exception is raised if the input data is not a generic numeric sequence, or the aditional arguments (as moment power, etc.) is of the improper data type or proper typed improper value. The exception provides the traceback information (if caught and processed) as well as the description of what is wrong.

Test steps: Check / review the source code. Demonstrate the fault situations in action (part of unit-tests). Findings: user-defined exceptions (see library *introspection_lib*) are used to communicate the improper input, and sufficient information is provided. See the respective module's test reports: TE001, TE002, TE003, TE004 and TE007.

Test result: PASS

Test Identifier: TEST-D-001

Requirement ID(s): REQ-USE-000

Verification method: D

Test goal: Usability of the library

Expected result: The library can be used with a basic but sufficient proficiency in Python programming language and statistics.

Test steps: Demonstrate the use of the different functions / class methods provided by the library in the separate scripts / programms following the examples provided in the reference documentation and unit-test modules. See demonstration tests DT003, DT004 and DT007.

Test result: PASS

Test Identifier: TEST-D-002

Requirement ID(s): REQ-IAR-000, REQ-IAR-002

Verification method: D

Test goal: System requirements check.

Expected result: The dependencies check script (module) is present and it detects the missing dependencies, too old versions as well as improper or too old Python interpreter version. If this script does not detect problems, the library should operate as designed.

Test steps: Install the library on different PCs using different OS. Run the check module check_dependencies.py. If missing dependencies or improper Python version is detected take the corrective actions. When the dependencies check has passed - try to work with the library, e.g. retrieve some data from the DPG. See also tested_OS. Findings: the wheel / egg distribution package installation via *pip* tool picks up the dependencies from the respective repository for the requested Python version, and reports the problem, when it is unable to find a dependecy (missing). The script *check_dependencies.py* included into the library successfully detects missing dependencies and checks the used Python interpreter version.

Test result: PASS

Test Identifier: TEST-D-003

Requirement ID(s): REQ-IAR-001

Verification method: D

Test goal: Multi-platform functionality.

Expected result: The library works as designed under MS Windows and GNU Linux OS, at least.

Test steps: Install the library and try to work with it on different PCs using different OS. Mark the tested OS + Python version in the tested_OS.md document as well as the result of the test.

Test result: PASS

Traceability

For traceability the relation between tests and requirements is summarized in the table below:

Requirement ID	Covered in test(s)	Verified [YES/NO])
REQ-FUN-000	TEST-A-000	YES
REQ-FUN-001	TEST-A-000	YES
REQ-SIO-000	TEST-A-001	YES
REQ-SIO-001	TEST-A-001	YES
REQ-INT-000	TEST-I-000	YES
REQ-AWM-000	TEST-D-000	YES
REQ-USE-000	TEST-D-001	YES
REQ-IAR-000	TEST-D-002	YES
REQ-IAR-001	TEST-D-003	YES
REQ-IAR-002	TEST-D-002	YES
REQ-UDR-000	TEST-I-001	YES

Software ready for production [YES/NO]	Rationale
YES	All tests are passed