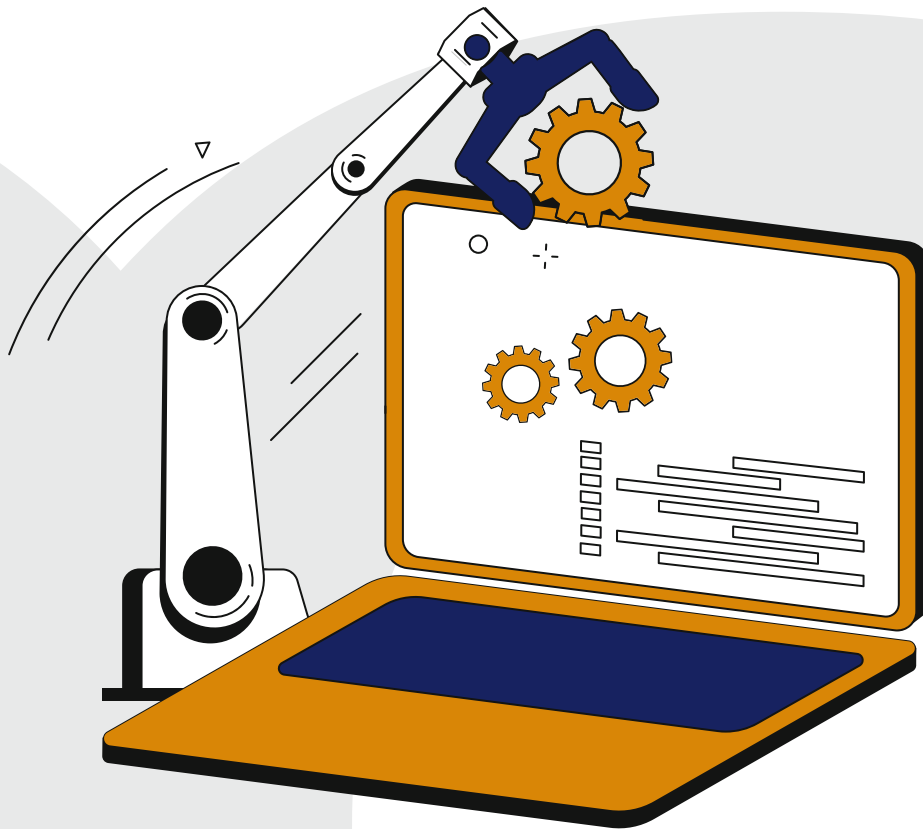


Quick Start for **BECKHOFF**



Automate Your Code

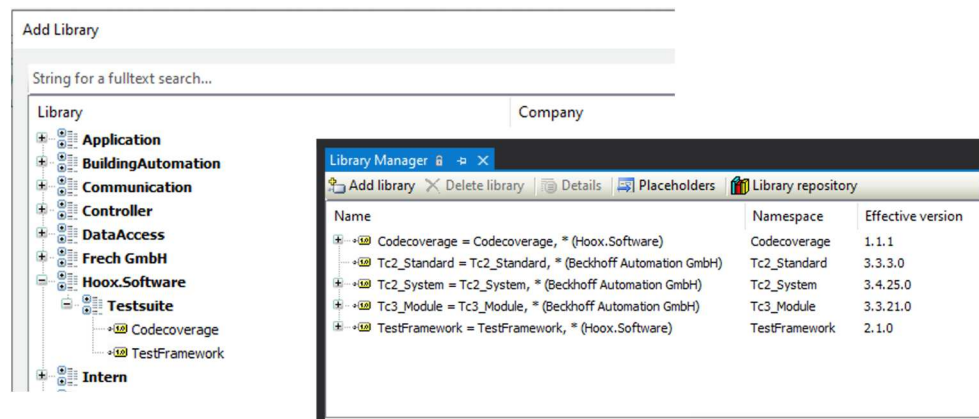
HOOX
[SOFTWARE]

HOOX Software
Patrick Dressel
Steinbühl 1
95233 Helmbrechts
M: +49 170 5260988
patrick@hoox.software

Test Library

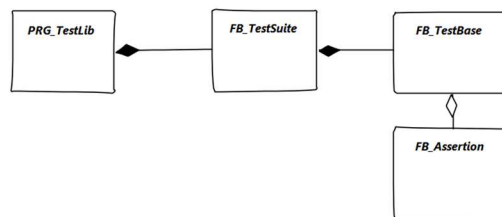
Import Library

Open the library repository management and add the new library.
Then, include the library in the application.



General

The library essentially consists of three parts: PRG_Testlib, FB_Testsuite, and FB_TestBase.



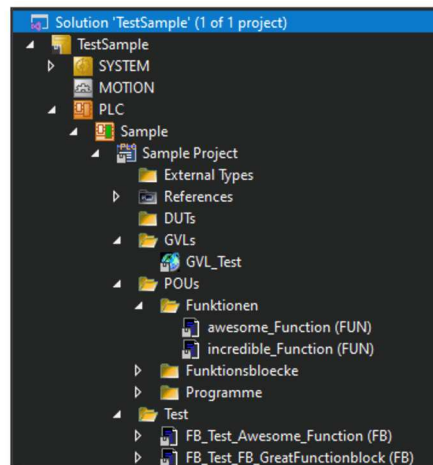
Component	Description
PRG_TestLib	Manages all configured TestSuites
FB_TestSuite	Manages the associated test cases
FB_TestBase	Base class for a test case
FB_StandardTextReport	Example for exporting the test run
FB_ReportBase	Base class for a report that is called after the test run.

The PRG_Testlib component contains the part for exporting the JUnit.xml file by default.

Custom reports can be created by inheriting from FB_ReportsBase.

Example

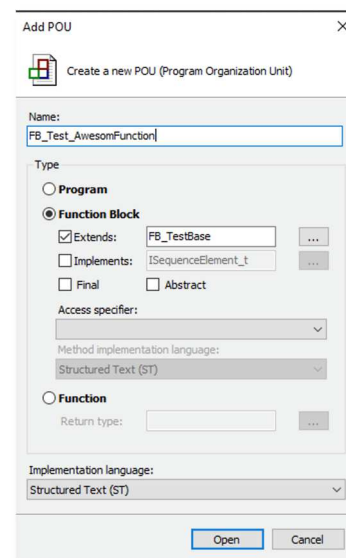
The function `awesomefunction` is to be tested.



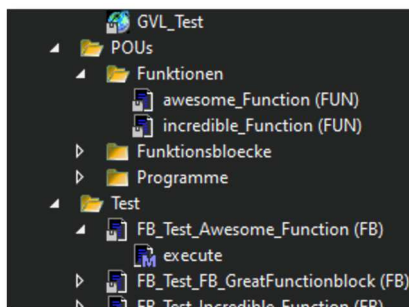
Procedure

1. Create test component

To do this, derive a new (test) class from the FB_TestBase class.



2. Override necessary methods



To use the test component, the `execute` method (Test Execution) must be overridden.

In the first step, it is sufficient to omit `prepare` and `cleanup`.

Code can be entered in these two methods that should be executed before (`prepare`) or after (`cleanup`) the test case.

3. Implement test

For the classification of the test case ("passed" or "failed"), so-called asserts must be called. These always consist of an actual and an expected value.

An assert is an assertion that the actual value corresponds : AssertTrue, or does not correspond : AssertFalse.

This value must be passed as a variable.

- Any data type can be used,
- The same data types must be used for both parameters.

Regarding the example:

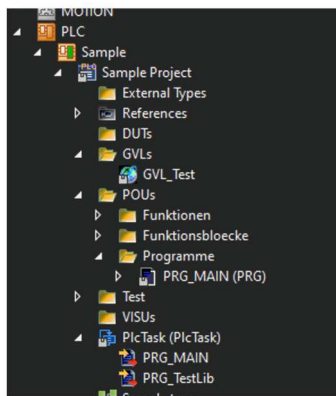
The function expects a DINT value and adds or subtracts this value from the current value of the instance.

```

19 *****
20 METHOD PUBLIC execute : ERetCode_t
21 VAR_INPUT
22 END_VAR
23
24 1
25 2
26 3
27 4
28 5
29 6
30 7
31 8
32 9
33 10
34 11
35 12
36 13
37 14
38 15
39 16
40 17

```

Additionally, it is recommended to save a hint via `setMessage(...)` before the call in case the assert fails, in order to get an indication of exactly where the error occurred.



Activate **PRG_Testlib** and assign it to a runtime task. In the example, the "PlcTask".

4. Run test

```

1 {attribute 'qualified_only'}
2 VAR_GLOBAL
3
4 bStart : BOOL;
5 bReady : BOOL;
6
7 // Register Testsuite
8 // Testsuite connects itself to PRG_TestLib
9 fb_TestSuite1 : FB_TestSuite( i_strId := 'TestSuite1',
10 i_strPath := 'Testframework\Testframework\UnitTest\');
11
12 // Register Test in Testsuite1
13 // Testcase connect itself to given Testsuite interface
14 // set Execute to false to skip the test.
15 fb_Test : FB_Test_Awesome_Function( i_strId := 'TF1.1',
16 i_strClassName := 'awesomefunction.TcPOU',
17 i_strPath := 'Source/POUs',
18 i_TestSuite := fb_TestSuite1,
19 i_Execute := TRUE);
20
21 END_VAR

```

To be able to execute the test cases, the command `i_bStart` in `PRG_Testlib` must be set. The test run is finished when `o_eState` outputs the value `eStateReady`.

Watch 1					
Expression	Appl...	Type	Value	Prepared value	Exec...
PRG_TestLib	TestS...	PRG_TESTLIB			Cyclic
i_bStart		BOOL	FALSE	TRUE	Cyclic
o_eState		ESTATE_T	eStateIdle		Cyclic
o_eResult		ERESULT_T	eStateIdle		Cyclic
o_sStatistic		SStatistic_t			Cyclic
bHwStart		BOOL	FALSE		Cyclic
bHwBusy		BOOL	FALSE		Cyclic
bHwReady		BOOL	FALSE		Cyclic
TestSuite		APPAY (0...255)			Cyclic

Instantiate Test Components

The test components must be instantiated within a global variable list. For execution, a Testsuite that manages multiple tests is needed for a better overview. For later evaluation, further information must be passed to the instance:

Testsuite

Parameter	Description
i_strId	Name/Identifier of the Testsuite
i_strPath	Path to the source code of the contained test cases

Test Case

Parameter	Beschreibung
i_strId	Unique ID of the test
i_strClassName	Class of the test object
i_strPath	Path to the source code of the contained test cases
i_iTestSuite	Associated Testsuite
i_bExecute	Execute test or not. If false, it will be shown later as SKIPPED.

```

GVL_Test  x
1  {attribute 'qualified_only'}
2  VAR_GLOBAL
3
4      bStart : BOOL;
5      bReady : BOOL;
6
7      // Register Testsuite
8      // Testsuite connects itself to PRG_TestLib
9      fb_TestSuite1 : FB_TestSuite( i_strId      := 'TestSuite1',
10                                   i_strPath     := 'Testframework\Testframework\UnitTest\');
11
12      // Register Test in Testsuite1
13      // Testcase connect itself to given Testsuite interface
14      // set Execute to false to skip the test.
15      fb_Test : FB_Test_Awesome_Function( i_strId      := 'TF1.1',
16                                           i_strClassName := 'awesomefunction.TcPOU',
17                                           i_strPath      := 'Source/POUs',
18                                           i_iTestSuite  := fb_TestSuite1,
19                                           i_bExecute    := TRUE);
20
21  END_VAR

```



www.hoox.software