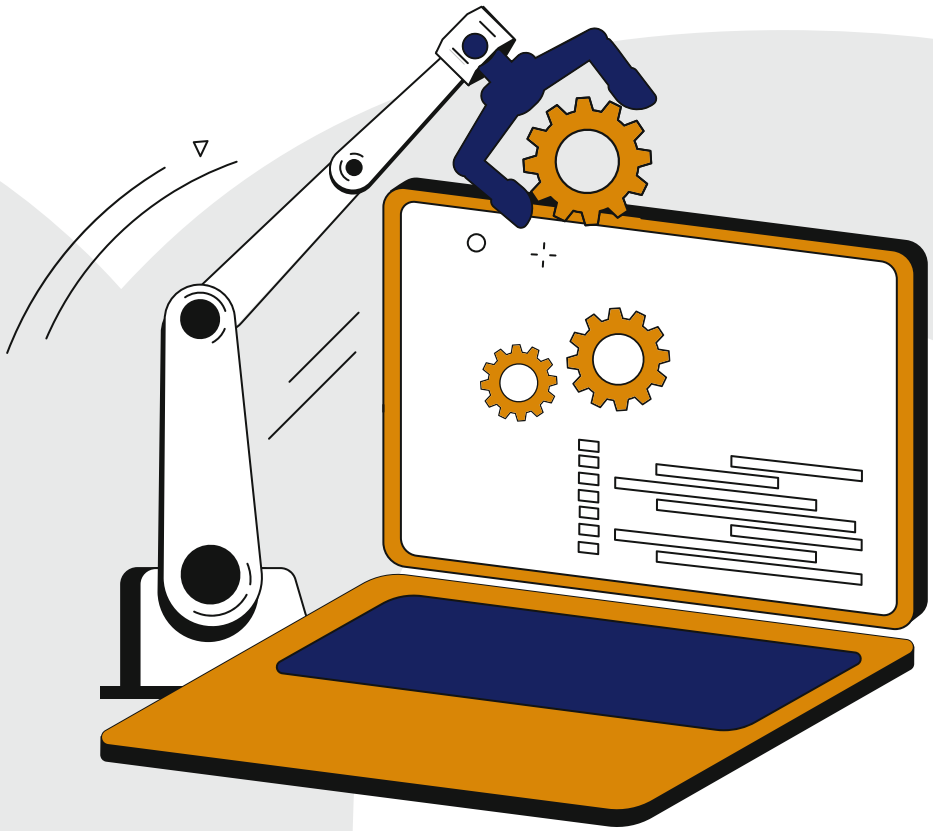
Quick Start for SIGMATEK



Automate Your Code

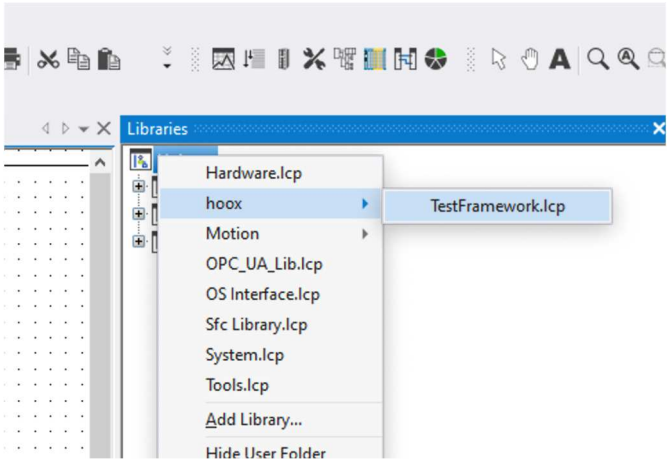
HOOX
[SOFTWARE]

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

Testbibliothek

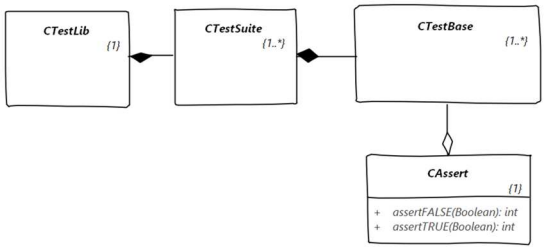
Bibliothek importieren

Die Verwaltung des Bibliotheksrepository öffnen und die neue Bibliothek hinzufügen. Danach in der Applikation die Bibliothek einbinden.



Allgemein

Die Bibliothek besteht im wesentlichen aus drei Teilen: CTestlib, CTestsuite und CTestBase.



Baustein	Beschreibung
`CTestLib`	Verwaltet alle konfigurierten TestSuites
`CTestSuite`	Verwaltet die zugeordneten Testfälle
`CTestBase`	Basisklasse für einen Testfall
`CStandardTextReport`	Beispiel für einen Export des Testlaufs
`CReportBase`	Basisklasse für einen Report der nach dem Testlauf aufgerufen wird.

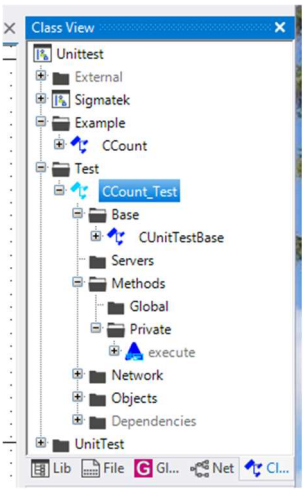
Die Klasse CTestlib enthält standardmäßig die Reports für den Export der JUnit und Coverage Informationen.

Eigene Reports können durch Veerbung von CReportsBase erstellt werden.

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

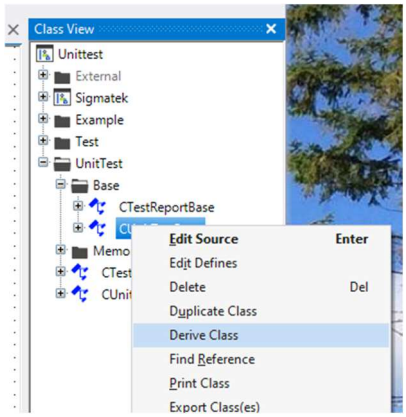
Beispiel

Es soll die Klasse *CCount* getestet werden.

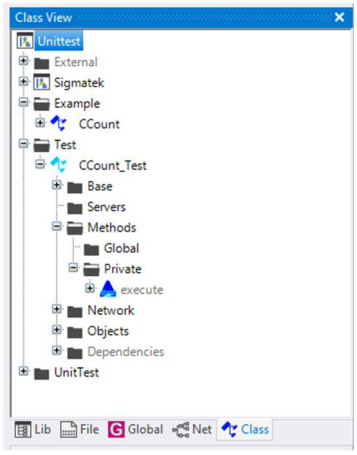


Vorgehensweise

1. Testbaustein anlegen
Dafür eine neue (Test)Klasse von der Klasse CTestBase ableiten.



2. Notwendige Methoden überschreiben



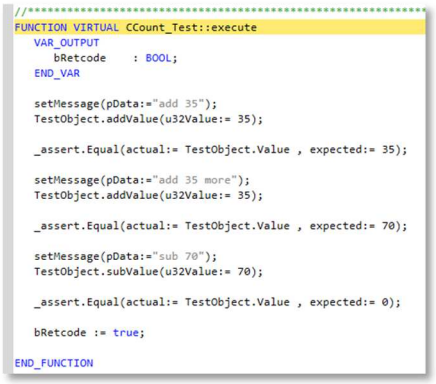
Um den Testbaustein zu verwenden muß die Methode `execute` *(TestAusführung)* überschrieben werden.

Im ersten Schritt reicht es aus `prepare` und `cleanup` jeweils `auszulassen`.
In diesen beiden Methoden kann Code eingetragen werden der vor (`prepare`) oder nach (`cleanup`) dem Testfall ausgeführt werden soll.

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

3. Test implementieren

Für die Einstufung des Testfalls ("passed" oder "failed") müssen sogenannte Asserts aufgerufen werden. Diese bestehen immer aus einem aktuellen und einem erwarteten Wert.



```
//*****************************************************************
FUNCTION VIRTUAL CCount_Test::execute
    VAR_OUTPUT
        bRetcode    : BOOL;
    END_VAR

    setMessage(pData:="add 35");
    TestObject.addValue(u32Value:= 35);

    _assert.Equal(actual:= TestObject.Value , expected:= 35);

    setMessage(pData:="add 35 more");
    TestObject.addValue(u32Value:= 35);

    _assert.Equal(actual:= TestObject.Value , expected:= 70);

    setMessage(pData:="sub 70");
    TestObject.subValue(u32Value:= 70);

    _assert.Equal(actual:= TestObject.Value , expected:= 0);

    bRetcode := true;

END_FUNCTION
```

Ein Assert ist eine Behauptung das der aktuelle dem erwarteten Wert

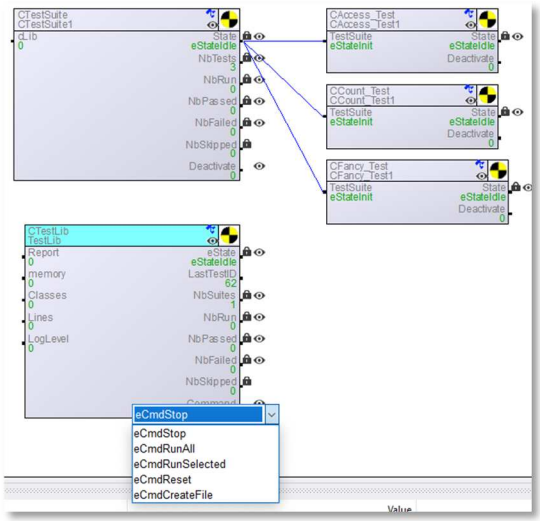
entpricht : `Assert.assertTrue`, oder
nicht entpricht : `Assert.assertFalse`.
bzw.
entpricht : `Assert.Equal`, oder
nicht entpricht : `Assert.NotEqual`.

- Zahlen werden als DINT oder REAL übergeben
- Es müssen für beide Parameter die gleichen Datentypen verwendet werden.

Auf das Beispiel bezogen:
Die Funktion erwartet einen DINT Wert und addiert oder subtrahiert diesen Wert vom aktuellen Wert der Instanz.

Zusätzlich empfiehlt es sich vor dem Aufruf über setMessage(...) einen Hinweis zu speichern falls der Assert fehlschlägt um einen Hinweis zu erhalten an welcher Stelle genau der Fehler aufgetreten ist.

4. Test ausführen



Um die Testfälle ausführen zu können muss das Kommando eCmdRunAll an die Testlib geschickt werden. Wenn die Testautomatisierung verwendet wird, wird dieser Schritt durch den Generator ausgeführt. Dieser erzeugt auch die Instanz TestLib.

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

Testbausteine Instanzieren

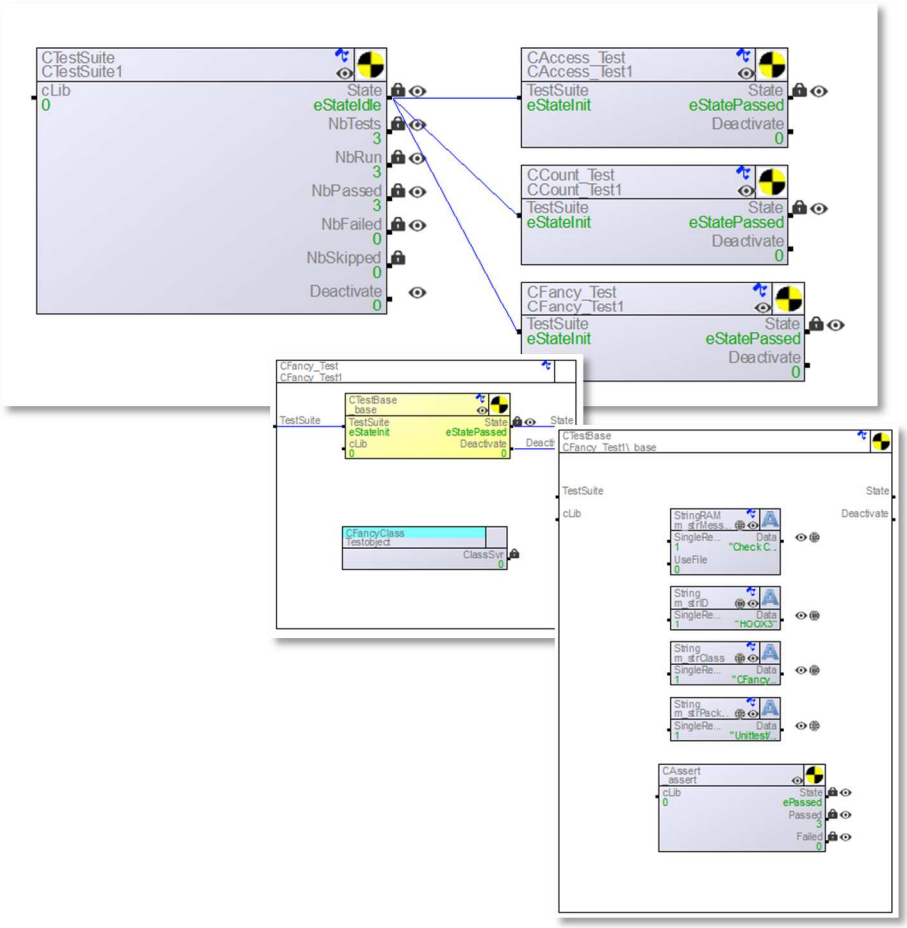
Die Testbausteine können über verschiedene Netzwerke instanziert werden. Für die Ausführung wird zur besseren Übersicht eine Testsuite benötigt die mehrere Tests verwaltet.
Für die spätere Auswertung müssen noch weitere Informationen der Instanz übergeben werden:

Testsuite

Parameter	Beschreibung
`m_strName`	Bezeichnung der Testsuite

Testfall

Parameter	Beschreibung
`m_strId`	eindeutige Id des Tests
`m_strClass`	Klasse des Testobjekts
`m_strPackage`	Pfad zum Quellcode der enthaltenen Testfälle
`m_strMessage`	Text der im Falle eines Fehlers Informationen liefern kann



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

HOOX

[SOFTWARE]

www.hoox.software