

Schnellstart: C++ Unit Tests mit Microsoft CppUnitTest

```
#include "stdafx.h"
#include "CppUnitTest.h" ← Include the Test Framework

#include "LocCounter.h" ← Include the System Under Test

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace examples
{
    TEST_CLASS(LocCounter_Tests) ← "This is a class containing tests"
    {
        TEST_METHOD(Drei_Codezeilen) ← Name of the test class
        {
            TEST_METHOD(Drei_Codezeilen) ← Name of the test
            {
                LocCounter sut;
                char* lines[] = { "abc", "d", "efgh" }; ← 1. Arrange

                int loc = sut.LOCCount(lines); ← 2. Act

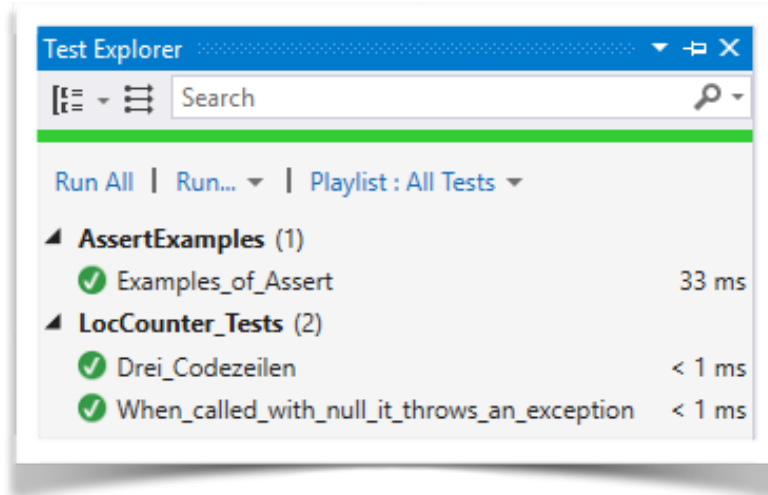
                Assert::AreEqual(3, loc); ← 3. Assert
            }
        }
    };
}
```

Tests ausführen

Visual Studio:

Ausführen: Im Menü unter *Test - Run - All Tests*

Anzeigen: Im Menü unter *Test - Windows - Test Explorer*



Assert Beispiele

Vergleiche

```
Assert::AreEqual(x, y);
Assert::AreSame(x, *z);
Assert::AreNotEqual(x, w);
Assert::AreNotSame(x, y);
```

Null Prüfung

```
Assert::IsNull(z);
Assert::IsNotNull(z);
```

Boolesche Ausdrücke

```
Assert::IsTrue(b);
Assert::IsFalse(b);
```

double und float

```
Assert::AreEqual(42.0, d, 0.001);
```

Strings

```
Assert::AreEqual(L"hello", L"HELLO", true);
```

Exceptions

```
TEST_METHOD(When_called_with_null_it_throws_an_exception)
{
    LocCounter* sut = new LocCounter();
    char** lines = nullptr;
    auto f = [sut, lines] { sut->LOCCount(lines); };
    Assert::ExpectException<std::invalid_argument>(f);
}
```

Assert Ergänzungen

Message und Line Info

```
Assert::AreEqual(x, y, L"x should equal y", LINE_INFO());
```

- Eine Meldung kann ergänzt werden. Sie wird ausgegeben, wenn die Annahme fehlschlägt.
- Durch das `LINE_INFO` Makro wird beim Fehlschlagen die Zeilennummer auch ohne `.pdb` Datei ausgegeben.

Testprojekt vs. Implementationsprojekt

- Implementationsprojekt muss ein `.dll` Projekt sein
- Testprojekt in Visual Studio hinzufügen mit:
Other Languages / Visual C++ / Test / Native Unit Test Project
- Referenz auf das Implementationsprojekt hinzufügen mit:
References / Add Reference
- Im `.h` File `__declspec(dllimport)` an die zu testenden Methoden ergänzen.

