NMock2 QUICK REFERENCE

**Declaration**

Mockery mocks = new Mockery();

InterfaceToBeMocked aMock =

(InterfaceToBeMocked) mocks.NewMock(typeof(InterfaceToBeMocked));

|  |  |
| --- | --- |
| **Setting Basic Expectation**  Expect.Once.On(aMock)  .Method( … )  .With( … )  .Will(Return.Value( … )); | **Setting Expectation That Throws Exception**  Expect.Once.On(aMock)  .Method( … )  .With( … )  .Will(**Throw.Exception( … )**); |

|  |  |
| --- | --- |
| **Setting Expectation On a Getter**  Expect.Once.On(aMock)  **.GetProperty( … )**  .Will(Return.Value( … ); | **Setting Expectation On a Setter**  Expect.Once.On(aMock)  **.SetProperty( … )**  .To( … ); |

|  |  |
| --- | --- |
| **Stubs**  Expect can be replaced with Stub which essentially means ‘zero or more’. Behavior of the stub will be defined and invoked if called, but the stub will not cause the test to fail.  **Stub.On(aMock)**  .Method( … )  .With( … )  .Will(Return.Value( … ); | **Constraining Order**  Mocks by default can be in any order. To constrain the order of a set of expectations, wrap the expectations with Using:  **using (mocks.Ordered)**  **{**  Expect.Once.On( …  Expect.Once.On( …  **}** |

**Possible Method Call Expectations**

|  |  |  |
| --- | --- | --- |
| Expect.Once | Expect.Never | Expect.AtLeastOnce |
| Expect.AtLeast(<# times>) | Expect.AtMost(<# times>) | Expect.Exactly(<# times>) |
| Expect.Between(<# times>, <# times>) | |  |

**Arguments**

.With( … ) can be replaced with .withAnyArguments() or .withNoArguments()

**Verificiation**

Mockery mocks = new Mockery();

…

mocks.VerifyAllExpectationsHaveBeenMet();