

**Basic Setup**

<p><b>Mocking a method</b> (static, instance, private and public)</p> <pre>Mock mock = MockManager.Mock(typeof(SomeType)); mock.ExpectCall("AnyMethod");  SomeType mockedInstance = new SomeType();</pre>	<p><b>Mocking an abstract class or interface</b></p> <pre>MockObject mock = MockManager.MockObject(typeof(AbstractClass)); mock.ExpectCall("AnyMethod");  AbstractClass mockedClass = (AbstractClass)mock.Object;</pre>
<p><b>Mocking Properties</b></p> <pre>mock.ExpectSet("Property"); mock.ExpectGet("Property", &lt;...value...&gt;);</pre>	<p><b>Mocking all instances</b></p> <pre>Mock mock = MockManager.MockAll(typeof(SomeType)); mock.ExpectCall("AnyMethod");  SomeType mockedInstance1 = new SomeType(); SomeType mockedInstance2 = new SomeType();</pre>
<p><b>Verification</b></p> <pre>MockManager.Verify(); MockManager.VerifyWithTimeout();</pre>	<p><b>Mocking Firing Events</b>*</p> <pre>Mock mock = MockManager.Mock(typeof(Button)); MockedEvent handle = mock.ExpectAddEvent("Button");  handle.Fire(this, EventArgs.Empty);</pre>

**Faking Behavior**

<p><b>Returning fake values</b></p> <pre>Mock mock =     MockManager.Mock(typeof(SomeType));  mock.ExpectAndReturn("AnyMethod",     &lt;...fake_return_value...&gt;);</pre>	<p><b>Changing Behavior</b></p> <pre>mock.ExpectAndReturn(...); mock.ExpectAndThrow(...);  mock.ExpectUnmockedCall();</pre>	<p><b>Mocking "Out" values</b><sup>†</sup></p> <pre>mock.ExpectCall("AnyMethod").Args(     new Assign("NewValue").     AndCheck("OldValue"));</pre>
<p><b>Repeating Expections</b></p> <pre>mock.ExpectCall("Method", times); mock.ExpectAndReturn("Method", &lt;...&gt;, times); mock.AlwaysReturn("Method", ...); mock.AlwaysThrow("Method", ...); mock.ExpectAlways("Method");</pre>	<p><b>Custom Return Values</b></p> <pre>mock.ExpectAndReturn("AnyMethod",     new DynamicReturnValue(         CustomReturnDelegate));</pre>	<p><b>Conditional Expectations</b><sup>†</sup></p> <pre>mock.ExpectCall("AnyMethod").     When(Check...); // PreDefined</pre>

**Validating Arguments**

```
mock.ExpectCall("AnyMethod").Args(<value>, Check...); // Build in Checks
```

**Built in Checks**

Check.IsAny	Check.IsIn	Check.Or	Check.IsEqualIgnoreCase	Check.IsMatch
Check.IsEqual	Check.IsSame	Check.NotEqual	Check.IsEqualIgnoreWhiteSpace	Check.IsCloseTo
Check.IsNull	Check.Not	Check.NotNull	Check.StartsWith	Check.IsMock
Check.IsTypeOf	Check.And	Check.NotIn	Check.EndsWith	Check.CustomChecker

**Custom Checks**

```
mock.ExpectCall("AnyMethod").Args(new ParameterCheckerEx(CustomCheckDelegate));
mock.ExpectCall("AnyMethod").Args(Check.CustomChecker(new ParameterCheckerEx(CustomCheckDelegate), ...));
```

**Stubst† Asynchronous Tests Rollback Field Values**

<pre>mock.StartBlock(     VerifyMode.PassIfNotCalled);  mock.EndBlock();</pre>	<pre>mock.ValidateArgsOnVerify = true; MockManager.VerifyWithTimeout();</pre>	<pre>ObjectState state = new ObjectState(test); // Test ...  // reset the test object fields state.Reset();</pre>
--	---	---

\* Professional  
† Enterprise