Root namespace of Moq (pronounced "Mock-you" or just "Mock"), the only mocking library for .NET developed from scratch to take full advantage of .NET 3.5 (i.e. Linq expression trees) and C# 3.0 features (i.e. lambda expressions) that make it the most productive, simple and refactoring-friendly mocking library available.

See the [online quickstarts](#) for more examples than those available in this code documentation.

The Mock<T> class is the core of the library, so it's a good place to start.

You can also read [blog entries](#) from around the world about Moq.
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>It</strong></td>
<td>Allows the specification of a matching condition for an argument in a method invocation, rather than a specific argument value. &quot;It&quot; refers to the argument being matched.</td>
</tr>
<tr>
<td><strong>MatcherAttribute</strong></td>
<td>Marks a method as a matcher, which allows complete replacement of the built-in <strong>It</strong> class with your own argument matching rules.</td>
</tr>
<tr>
<td><strong>Mock</strong></td>
<td>Base class for mocks and static helper class with methods that apply to mocked objects, such as $\text{Get&lt;}(&lt;\text{T}&gt;)(\text{T})$ to retrieve a $\text{Mock&lt;}(&lt;\text{T}&gt;)(\text{T})$ from an object instance.</td>
</tr>
<tr>
<td><strong>Mock&lt;Of</strong></td>
<td>Core implementation of the $\text{IMock&lt;}(&lt;\text{T}&gt;)$ interface.</td>
</tr>
<tr>
<td><strong>Mock&lt;T&gt;</strong></td>
<td>Represents a generic event that has been mocked and can be raised.</td>
</tr>
<tr>
<td><strong>MockedEvent</strong></td>
<td>Provides a typed <strong>MockedEvent</strong> for a specific type of $\text{EventArgs}$.</td>
</tr>
<tr>
<td><strong>MockedException</strong></td>
<td>Exception thrown by mocks when expectations are not met, the mock is not properly setup, etc.</td>
</tr>
<tr>
<td><strong>MockFactory</strong></td>
<td>Utility factory class to use to construct multiple mocks when consistent verification is desired for all of them.</td>
</tr>
</tbody>
</table>
## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMock</td>
<td>Base mock interface exposing non-generic members.</td>
</tr>
<tr>
<td>IMock&lt;Of&lt;(T)&gt;</td>
<td>Provides a mock implementation of T.</td>
</tr>
</tbody>
</table>
## Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DefaultValue</code></td>
<td>Determines the way default values are generated calculated for loose mocks.</td>
</tr>
<tr>
<td><code>MockBehavior</code></td>
<td>Options to customize the behavior of the mock.</td>
</tr>
<tr>
<td><code>Range</code></td>
<td>Kind of range to use in a filter specified through <code>IsInRange&lt;(Of&lt;(TValue)&gt;))(TValue,TValue,Range)</code>.</td>
</tr>
</tbody>
</table>

Send comments on this topic to [moqdisc@googlegroups.com](mailto:moqdisc@googlegroups.com)
Determines the way default values are generated calculated for loose mocks.

**Namespace:** Moq

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public enum DefaultValue
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Default behavior, which generates empty values for value types (i.e. <code>default(int)</code>), empty array and enumerables, and nulls for all other reference types.</td>
</tr>
<tr>
<td>Mock</td>
<td>Whenever the default value generated by <code>Empty</code> is null, replaces this value with a mock (if the type can be mocked).</td>
</tr>
</tbody>
</table>
See Also

Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Base mock interface exposing non-generic members.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface IMock
See Also

IMock Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The **IMock** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateEventHandler</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Verify</td>
<td>Verifies that all verifiable expectations have been met.</td>
</tr>
<tr>
<td>VerifyAll</td>
<td>Verifies all expectations regardless of whether they have been flagged as verifiable.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CallBase</strong></td>
<td>Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to true (True in Visual Basic).</td>
</tr>
<tr>
<td><strong>DefaultValue</strong></td>
<td>Determines how to generate default values for loose mocks on unexpected invocations.</td>
</tr>
<tr>
<td><strong>Object</strong></td>
<td>The mocked object instance.</td>
</tr>
</tbody>
</table>
See Also

IMock Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The **IMock** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CreateEventHandler</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>Verify</code></td>
<td>Verifies that all verifiable expectations have been met.</td>
</tr>
<tr>
<td><code>VerifyAll</code></td>
<td>Verifies all expectations regardless of whether they have been flagged as verifiable.</td>
</tr>
</tbody>
</table>
See Also

IMock Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CreateEventHandler&lt;TEventArgs&gt;</code></td>
<td>Creates a handler that can be associated to an event receiving the given TEventArgs and can be used to raise the event.</td>
</tr>
<tr>
<td><code>CreateEventHandler()</code></td>
<td>Creates a handler that can be associated to an event receiving a generic EventArgs and can be used to raise the event.</td>
</tr>
</tbody>
</table>
See Also

**IMock Interface**
**IMock Members**
**Moq Namespace**

Send comments on this topic to [moqdisc@googlegroups.com](mailto:moqdisc@googlegroups.com)
**C#**

**Moq**

IMock...::CreateEventHandler<(Of <(TEventArgs)>)> Method

**IMock Interface  Example  See Also  Send Feedback**

Creates a handler that can be associated to an event receiving the given TEventArgs and can be used to raise the event.

**Namespace:** Moq

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

MockedEvent<TEventArgs> CreateEventHandler<TEventArgs>()
where TEventArgs : EventArgs
Type Parameters

TEventArgs
   Type of EventArgs data passed in to the event.
Examples

This example shows how to invoke an event with a custom event arguments class in a view that will cause its corresponding presenter to react by changing its state:

C#

```csharp
var mockView = new Mock<IOrdersView>();
var mockedEvent = mockView.CreateEventHandler<OrderEventArgs>();

var presenter = new OrdersPresenter(mockView.Object);

// Check that the presenter has no selection by default
Assert.Null(presenter.SelectedOrder);

// Create a mock event handler of the appropriate type
var handler = mockView.CreateEventHandler<OrderEventArgs>();
// Associate it with the event we want to raise
mockView.Object.Cancel += handler;
// Finally raise the event with a specific arguments data
handler.Raise(new OrderEventArgs { Order = new Order("moq", 500) });

// Now the presenter reacted to the event, and we have a selected or
Assert.NotNull(presenter.SelectedOrder);
Assert.Equal("moq", presenter.SelectedOrder.ProductName);
```
See Also

IMock Interface
CreateEventHandler Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

IMock...::CreateEventHandler Method

See Also  Send Feedback

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)

Creates a handler that can be associated to an event receiving a generic EventArgs and can be used to raise the event.
Syntax

C#

MockedEvent<EventArgs> CreateEventHandler()
**Examples**

This example shows how to invoke a generic event in a view that will cause its corresponding presenter to react by changing its state:

```csharp
var mockView = new Mock<IOrdersView>();
var mockedEvent = mockView.CreateEventHandler();

var presenter = new OrdersPresenter(mockView.Object);

// Check that the presenter is not in the "Canceled" state
Assert.False(presenter.IsCanceled);

// Create a mock event handler of the appropriate type
var handler = mockView.CreateEventHandler();
// Associate it with the event we want to raise
mockView.Object.Cancel += handler;
// Finally raise the event
handler.Raise(EventArgs.Empty);

// Now the presenter reacted to the event, and changed its state
Assert.True(presenter.IsCanceled);
```
See Also

IMock Interface
CreateEventHandler Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

IMock...::Verify Method

IMock Interface  Example  See Also  Send Feedback

Verifies that all verifiable expectations have been met.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

void Verify()
Examples

This example sets up an expectation and marks it as verifiable. After the mock is used, a Verify() call is issued on the mock to ensure the method in the expectation was invoked:

C#

```csharp
var mock = new Mock<IWarehouse>();
mock.Expect(x => x.HasInventory(TALISKER, 50)).Verifiable().Returns(...
    // other test code
    ...
    // Will throw if the test code has didn't call HasInventory.
    mock.Verify();
```
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Moq::MockException</code></td>
<td>Not all verifiable expectations were met.</td>
</tr>
</tbody>
</table>
See Also

IMock Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Verifies all expectations regardless of whether they have been flagged as verifiable.

**Namespace:** Moq

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
C#

void VerifyAll()
Examples

This example sets up an expectation without marking it as verifiable. After the mock is used, a VerifyAll() call is issued on the mock to ensure that all expectations are met:

C#  

```csharp
var mock = new Mock<I Warehouse> ();
mock.Expect (x => x. HasInventory (TALISKER, 50 )). Returns (true );
...
// other test code
...
// Will throw if the test code has didn't call HasInventory, even
// that expectation was not marked as verifiable.
mock.VerifyAll ();
```
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq:::MockException</td>
<td>At least one expectation was not met.</td>
</tr>
</tbody>
</table>
See Also

IMock Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The **IMock** type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CallBase</strong></td>
<td>Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to <code>true</code> (true in Visual Basic).</td>
</tr>
<tr>
<td><strong>DefaultValue</strong></td>
<td>Determines how to generate default values for loose mocks on unexpected invocations.</td>
</tr>
<tr>
<td><strong>Object</strong></td>
<td>The mocked object instance.</td>
</tr>
</tbody>
</table>
See Also

IMock Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
IMock...::CallBase Property

IMock Interface  See Also  Send Feedback

Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to true True True True (True in Visual Basic).

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

bool CallBase { get; set; }
See Also

**IMock Interface**
**Moq Namespace**

Send comments on this topic to [moqdisc@googlegroups.com](mailto:moqdisc@googlegroups.com)
Determines how to generate default values for loose mocks on unexpected invocations.

Namespace: **Moq**
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

DefaultValue  DefaultValue { get; set; }
See Also

IMock Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The mocked object instance.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

Object Object { get; }
See Also

IMock Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Provides a mock implementation of T.

**Namespace:** Moq
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface IMock<T>
where T : class
Type Parameters

T
Type to mock, which can be an interface or a class.
Remarks

Only abstract and virtual members of classes can be mocked.

The behavior of the mock with regards to the expectations and the actual calls is determined by the optional MockBehavior that can be passed to the Mock<Of <(T)>>(MockBehavior) constructor.
Examples

The following example shows setting expectations with specific values for method invocations:

C#  

```csharp
//setup - data
var order = new Order(TALISKER, 50);
var mock = new Mock<IWarehouse>();

//setup - expectations
mock.Expect(x => x.HasInventory(TALISKER, 50)).Returns(true);

//exercise
order.Fill(mock.Object);

//verify
Assert.True(order.IsFilled);
```

The following example shows how to use the It class to specify conditions for arguments instead of specific values:

C#  

```csharp
//setup - data
var order = new Order(TALISKER, 50);
var mock = new Mock<IWarehouse>();

//setup - expectations
//shows how to expect a value within a range
mock.Expect(x => x.HasInventory(
    It.IsAny<string>(),
    It.IsInRange(0, 100, Range.Inclusive)))
  .Returns(false);

//shows how to throw for unexpected calls. contrast with the "verify
mock.Expect(x => x.Remove(
    It.IsAny<string>(),
    It.IsAny<int>()))
  .Throws(new InvalidOperationException());

//exercise
order.Fill(mock.Object);
```
//verify
Assert.False(order.IsFilled);
See Also

IMock<(Of <(T)>)> Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `IMock<(Of <(T)>)>` type exposes the following members.
<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>As&lt;(&lt;TInterface&gt;)&gt;</td>
<td>Adds an interface implementation to the mock, allowing expectations to be set for it.</td>
</tr>
<tr>
<td>Expect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ExpectGet&lt;(&lt;TProperty&gt;)&gt;</td>
<td>Sets an expectation on the mocked type for a call to a property getter.</td>
</tr>
<tr>
<td>ExpectSet</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Verify</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>VerifyAll</td>
<td>Implements <strong>VerifyAll()</strong>.</td>
</tr>
<tr>
<td>VerifyGet&lt;(&lt;TProperty&gt;)&gt;</td>
<td>Verifies that a property was read on the mock. Use in conjunction with the default <strong>Loose</strong>.</td>
</tr>
<tr>
<td>VerifySet</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior</td>
<td>Behavior of the mock, according to the value set in the constructor.</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Specifies the behavior to use when returning default values for unexpected invocations.</td>
</tr>
<tr>
<td>Object</td>
<td>Exposes the mocked object instance.</td>
</tr>
</tbody>
</table>
See Also

IMock<OF (T)> Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `IMock<Of <(T)>>` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>As&lt;Of &lt;TInterface&gt;&gt;</td>
<td>Adds an interface implementation to the mock, allowing expectations to be set for it.</td>
</tr>
<tr>
<td>Expect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ExpectGet&lt;Of &lt;TProperty&gt;&gt;</td>
<td>Sets an expectation on the mocked type for a call to a property getter.</td>
</tr>
<tr>
<td>ExpectSet</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Verify</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>VerifyAll</td>
<td>Implements <code>VerifyAll()</code>.</td>
</tr>
<tr>
<td>VerifyGet&lt;Of &lt;TProperty&gt;&gt;</td>
<td>Verifies that a property was read on the mock. Use in conjunction with the default Loose.</td>
</tr>
<tr>
<td>VerifySet</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

IMock<Of<T>> Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Add an interface implementation to the mock, allowing expectations to be set for it.

**Namespace:** [Moq](https://moq.net)

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IMock<TInterface> As<TInterface>()
where TInterface : class
Type Parameters

TInterface
Type of interface to cast the mock to.
Remarks

This method can only be called before the first use of the mock Object property, at which point the runtime type has already been generated and no more interfaces can be added to it.

Also, TInterface must be an interface and not a class, which must be specified when creating the mock instead.
Examples

The following example creates a mock for the main interface and later adds `IDisposable` to it to verify it's called by the consumer code:

```csharp
var mock = new Mock<IProcessor>();
mock.Expect(x => x.Execute("ping"));

// add IDisposable interface
var disposable = mock.As<IDisposable>();
disposable.Expect(d => d.Dispose()).Verifiable();
```
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidOperationException</td>
<td>The mock type has already been generated by accessing the <strong>Object</strong> property.</td>
</tr>
<tr>
<td>ArgumentException</td>
<td>The TInterface specified is not an interface.</td>
</tr>
</tbody>
</table>
See Also

IMock<(Of <(T)>)> Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members
Moq
IMock<Of <(T)>>.:.:.Expect Method
IMock<Of <(T)>>) Interface  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expect(Expression&lt;Of (Action&lt;Of (T)&gt;)&gt;)</td>
<td>Sets an expectation on the mocked type for a call to to a void method.</td>
</tr>
<tr>
<td>Expect(Of (TResult))</td>
<td>Sets an expectation on the mocked type for a call to to a value returning method.</td>
</tr>
<tr>
<td>(Expression&lt;Of &lt;(Func&lt;Of (T, TResult)&gt;)&gt;&gt;)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IMock(Of (T)>) Interface
IMock(Of (T)>) Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Moq

IMock<Of (T)>...:..Expect Method (<(Of <(Of (T)>)>))

IMock<Of (T)> Interface  Example  See Also  Send Feedback

Sets an expectation on the mocked type for a call to to a void method.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IExpect Expect(
    Expression<Action<T>> expression
)

Parameters

equation
    Type: Expression<(Of (<Action<(Of (T)>))>)
    Lambda expression that specifies the expected method invocation.
Remarks

If more than one expectation is set for the same method or property, the latest one wins and is the one that will be executed.
Examples

C#

```csharp
var mock = new Mock<IProcessor>();
mock.Expect(x => x.Execute("ping"));
```
See Also

IMock<Of <(T)>>) Interface
Expect Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Sets an expectation on the mocked type for a call to a value returning method.

**Namespace:**  Moq  
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
C#

IExpect<TResult> Expect<TResult>(
    Expression<Func<T, TResult>> expression
)

Parameters

expression
    Type: Expression<(Of <(Func<(Of <(T, TResult)>))>)>
    Lambda expression that specifies the expected method invocation.
Type Parameters

TResult
Type of the return value. Typically omitted as it can be inferred from the expression.
Remarks

If more than one expectation is set for the same method or property, the latest one wins and is the one that will be executed.
 Examples

C#

mock.Expect(x => x.HasInventory("Talisker", 50)).Returns(true);
See Also

IMock<Of<(<T>)>> Interface
Expect Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Sets an expectation on the mocked type for a call to a property getter.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

```csharp
IExpectGetter<TProperty> ExpectGet<TProperty>(
    Expression<Func<T, TProperty>> expression
)
```

**Parameters**

expression
  Type: Expression<(Of (Func<(Of (T, TProperty)>))>)
  Lambda expression that specifies the expected property getter.
Type Parameters

TPROPERTY
  Type of the property. Typically omitted as it can be inferred from the expression.
Remarks

If more than one expectation is set for the same property getter, the latest one wins and is the one that will be executed.
Examples

C#

```csharp
mock.ExpectGet(x => x.Suspended)
    .Returns(true);
```
See Also

IMock<(Of <(T)>)> Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members
Include Inherited Members
Moq
IMock(Of (T)>):::ExpectSet Method
IMock(Of (T)>) Interface
See Also
Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ExpectSet&lt;(Of &lt;(TProperty)&gt;)&gt;</code></td>
<td>Sets an expectation on the mocked type for a call to a property setter.</td>
</tr>
<tr>
<td><code>ExpectSet&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;))</code></td>
<td>Sets an expectation on the mocked type for a call to a property setter with a specific value.</td>
</tr>
</tbody>
</table>
See Also

IMock(Of (T)) Interface
IMock(Of (T)) Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Sets an expectation on the mocked type for a call to a property setter.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
### Syntax

**C#**

```csharp
IExpectSetter<TProperty> ExpectSet<TProperty>(
    Expression<Func<T, TProperty>> expression
)
```

**Parameters**

- `expression`
  ```csharp
  Type: Expression<
    (Of
      (Func<
        (Of
          (T
            ,
            TProperty
          )
        ,
        TProperty
      )
    )
  )

  Lambda expression that specifies the expected property setter.
  ```
Type Parameters

TProperty
Type of the property. Typically omitted as it can be inferred from the expression.
Remarks

If more than one expectation is set for the same property setter, the latest one wins and is the one that will be executed.
Examples

C#

mock.ExpectSet(x => x.Suspended);
See Also

IMock<(Of <(T)>>) Interface
ExpectSet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Sets an expectation on the mocked type for a call to a property setter with a specific value.

**Namespace:**  Moq
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
IExpectSetter<TProperty> ExpectSet<TProperty>(
    Expression<Func<T, TProperty>> expression,
    TProperty value
)
```

**Parameters**

**expression**
- Type: `Expression<Func<T, TProperty>>`
  - Lambda expression that specifies the expected property setter.

**value**
- Type: `TProperty`
  - The value expected to be set for the property.
Type Parameters

TProperty
Type of the property. Typically omitted as it can be inferred from the expression.
Remarks

More than one expectation can be set for the setter with different values.
Examples

C#

mock.ExpectSet(x => x.Suspended, true);
See Also

IMock<Of<(T)>>) Interface
ExpectSet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members

Moq

IMock<Of <(T)>)::.Verify Method

IMock<Of <(T)>) Interface  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Verify&lt;&gt;</code></td>
<td>Implements <code>Verify&lt;&gt;</code>. Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjunction with the default <code>Loose</code>.</td>
</tr>
<tr>
<td><code>Verify(Expression&lt;(Of &lt;(Action&lt;(Of &lt;(T)&gt;)&gt;)&gt;))</code></td>
<td>Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjunction with the default <code>Loose</code>.</td>
</tr>
<tr>
<td><code>Verify&lt;(Of &lt;(TResult)&gt;)(Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TResult)&gt;)&gt;)&gt;))</code></td>
<td>Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjunction with the default <code>Loose</code>.</td>
</tr>
</tbody>
</table>
See Also

IMock(Of (T)> ) Interface
IMock(Of (T)> ) Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Moq

IMock<(Of <(T)>)>...::Verify Method

IMock<(Of <(T)>)> Interface  See Also  Send Feedback

Implements  Verify()().

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

void Verify()
See Also

IMock<(Of (T)>)> Interface
Verify Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjunction with the default Loose.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
void Verify(
    Expression<Action<T>> expression
)
```

Parameters

expression
  Type: Expression<(Of <(Action<(Of <(T)>)>)>))
  Expression to verify.
Examples

This example assumes that the mock has been used, and later we want to verify that a given invocation with specific parameters was performed:

C#

```csharp
var mock = new Mock<IProcessor>();
// exercise mock
//...
// Will throw if the test code didn't call Execute with a "ping" str
mock.Verify(proc => proc.Execute("ping"));
```
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Moq:::MockException</code></td>
<td>The invocation was not performed on the mock.</td>
</tr>
</tbody>
</table>
See Also

IMock<(Of <(T)>)> Interface
Verify Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
IMock(Of (T)>)\ldots\text{Verify}(Of (TResult)>) Method (Of (Of (Of (T, TResult)>)>)>)

IMock(Of (T)>) Interface  Example  See Also  Send Feedback

Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjunction with the default Loose.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

void Verify<TResult>(
    Expression<Func<T, TResult>> expression
)

Parameters

description
   Type: Expression<(Of <(Func<(Of <(T, TResult)>))>))
   Expression to verify.
Type Parameters

TResult
  Type of return value from the expression.
**Examples**

This example assumes that the mock has been used, and later we want to verify that a given invocation with specific parameters was performed:

```csharp
var mock = new Mock<IWarehouse>();
// exercise mock
//...  
// Will throw if the test code didn't call HasInventory.
mock.Verify(warehouse => warehouse.HasInventory(TALISKER, 50));
```
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq::MockException</td>
<td>The invocation was not performed on the mock.</td>
</tr>
</tbody>
</table>
See Also

IMock<where T> Interface
Verify Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
IMock<Of <(T)>>...:.VerifyAll Method
IMock<Of <(T)>> Interface  See Also  Send Feedback
Implements VerifyAll<QQQ>.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

void VerifyAll()
See Also

IMock<Of <(T)>>) Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Verifies that a property was read on the mock. Use in conjunction with the default Loose.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

void VerifyGet<TProperty>(
    Expression<Func<T, TProperty>> expression
)

Parameters

expression
    Type: Expression<(Of <(Func<(Of <(T, TProperty)>)>)>)
    Expression to verify.
Type Parameters

TProperty
Type of the property to verify. Typically omitted as it can be inferred from the expression's return type.
Examples

This example assumes that the mock has been used, and later we want to verify that a given property was retrieved from it:

```csharp
var mock = new Mock<IWarehouse>();
// exercise mock
//...
// Will throw if the test code didn't retrieve the IsClosed property
mock.VerifyGet(warehouse => warehouse.IsClosed);
```
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Moq::MockException</code></td>
<td>The invocation was not performed on the mock.</td>
</tr>
</tbody>
</table>
See Also

**IMock<(Of <(T)>)> Interface**

**Moq Namespace**

Send comments on this topic to [moqdisc@googlegroups.com](mailto:moqdisc@googlegroups.com)
C#

Include Protected Members
Include Inherited Members
Moq

IMock(Of(T))::VerifySet Method

IMock(Of(T)) Interface    See Also    Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerifySet&lt;(Of &lt;(TProperty)&gt;)&gt; (Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;))</td>
<td>Verifies that a property has been set on the mock. Use in conjunction with the default Loose.</td>
</tr>
<tr>
<td>VerifySet&lt;(Of &lt;(TProperty)&gt;)&gt; (Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;), TProperty)</td>
<td>Verifies that a property has been set on the mock to the given value. Use in conjunction with the default Loose.</td>
</tr>
</tbody>
</table>
See Also

IMock(Of (T)>) Interface
IMock(Of (T)>) Members
Mock Namespace

Send comments on this topic to moqdisc@googlegroups.com
Verifies that a property has been set on the mock. Use in conjunction with the default Loose.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

C#

```csharp
void VerifySet<TProperty>(
    Expression<Func<T, TProperty>> expression
)
```

**Parameters**

**expression**

Type: `Expression<((Expression<Func<T, TProperty>>)>>)`
Expression to verify.
Type Parameters

TProperty
Type of the property to verify. Typically omitted as it can be inferred from the expression's return type.
Examples

This example assumes that the mock has been used, and later we want to verify that a given invocation with specific parameters was performed:

```csharp
var mock = new Mock<IWarehouse>();
// exercise mock
//...  
// Will throw if the test code didn't set the IsClosed property.
mock.VerifySet(warehouse => warehouse.IsClosed);
```
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Moq::MockException</code></td>
<td>The invocation was not performed on the mock.</td>
</tr>
</tbody>
</table>
See Also

IMock<(Of <(T)>)> Interface
VerifySet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
IMock<(Of<(T)>)>....:VerifySet<(Of<(T)>)> Method (<(Of<(T)>)>), TProperty)

IMock<(Of<(T)>)> Interface  Example  See Also  Send Feedback

Verifies that a property has been set on the mock to the given value. Use in conjunction with the default Loose.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
### Syntax

**C#**

```csharp
void VerifySet<TProperty>(
    Expression<Func<T, TProperty>> expression,
    TProperty value
)
```

**Parameters**

- **expression**
  - Type: `Expression<Func<T, TProperty>>`
  - Expression to verify.

- **value**
  - Type: `TProperty`
  - The value that should have been set on the property.
Type Parameters

TProperty
Type of the property to verify. Typically omitted as it can be inferred from the expression's return type.
Examples

This example assumes that the mock has been used, and later we want to verify that a given invocation with specific parameters was performed:

```csharp
var mock = new Mock<IWarehouse>();
// exercise mock
... // Will throw if the test code didn't set the IsClosed property to true
mock.VerifySet(warehouse => warehouse.IsClosed, true);
```
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq...MockException</td>
<td>The invocation was not performed on the mock.</td>
</tr>
</tbody>
</table>
See Also

IMock<OF<T>> Interface
VerifySet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `IMock<(Of <T>>)` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior</td>
<td>Behavior of the mock, according to the value set in the constructor.</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Specifies the behavior to use when returning default values for unexpected invocations.</td>
</tr>
<tr>
<td>Object</td>
<td>Exposes the mocked object instance.</td>
</tr>
</tbody>
</table>
See Also

IMock<(Of <(T)>)> Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Behavior of the mock, according to the value set in the constructor.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

MockBehavior Behavior { get; }
See Also

IMock<Of<(T)>)> Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies the behavior to use when returning default values for unexpected invocations.

**Namespace:** [Moq](https://github.com/Moq/Moq)

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```
DefaultValue DefaultValue { get; set; }
```
See Also

IMock<(Of <(T)>)> Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Exposes the mocked object instance.

**Namespace:**  [Moq](https://github.com/Moq/Moq)

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

T Object { get; }
See Also

IMock<(Of (T)>)> Interface
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Allows the specification of a matching condition for an argument in a method invocation, rather than a specific argument value. "It" refers to the argument being matched.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public static class It
Remarks

This class allows the expectation to match a method invocation with an arbitrary value, with a value in a specified range, or even one that matches a given predicate.
Inheritance Hierarchy

Object
Moq...It
See Also

It Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The It type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is&lt;Of TValue&gt;</td>
<td>Matches any value that satisfies the given predicate.</td>
</tr>
<tr>
<td>IsAny&lt;Of TValue&gt;</td>
<td>Matches any value of the given TValue type.</td>
</tr>
<tr>
<td>IsInRange&lt;Of TValue&gt;</td>
<td>Matches any value that is in the range specified.</td>
</tr>
<tr>
<td>IsRegex</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

It Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `It` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Is&lt;(Of &lt;(TValue)&gt;)&gt;</code></td>
<td>Matches any value that satisfies the given predicate.</td>
</tr>
<tr>
<td><code>IsAny&lt;(Of &lt;(TValue)&gt;)&gt;</code></td>
<td>Matches any value of the given TValue type.</td>
</tr>
<tr>
<td><code>IsInRange&lt;(Of &lt;(TValue)&gt;)&gt;</code></td>
<td>Matches any value that is in the range specified.</td>
</tr>
<tr>
<td><code>IsRegex</code></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

It Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Matches any value that satisfies the given predicate.

**Namespace:** [Moq](https://moq.net)

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

```csharp
public static TValue Is<TValue>(
    Expression<Predicate<TValue>> match
)
```

**Parameters**

`match`

Type: `Expression<(Of <(Predicate<(Of <(TValue)>)))>)`

The predicate used to match the method argument.
**Type Parameters**

TValue

Type of the argument to check.
Remarks

Allows the specification of a predicate to perform matching of method call arguments.
Examples

This example shows how to return the value 1 whenever the argument to the Do method is an even number.

```csharp
mock.Expect(x => x.Do(It.Is<int>(i => i % 2 == 0)))
   .Returns(1);
```

This example shows how to throw an exception if the argument to the method is a negative number:

```csharp
mock.Expect(x => x.GetUser(It.Is<int>(i => i < 0)))
   .Throws(new ArgumentException());
```
See Also

It Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Matches any value of the given TValue type.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public static TValue IsAny<TValue>()
Type Parameters

TValue
   Type of the value.
Remarks

Typically used when the actual argument value for a method call is not relevant.
Examples

C#

// Throws an exception for a call to Remove with any string value.
mock.Expect(x => x.Remove(It.IsAny<string>())).Throws(new InvalidOperationException());
See Also

*It Class*
*Moq Namespace*

Send comments on this topic to *moqdisc@googlegroups.com*
Matches any value that is in the range specified.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

```csharp
public static TValue IsInRange<TValue>(
    TValue from,
    TValue to,
    Range rangeKind
)
where TValue : IComparable
```

**Parameters**

from
Type: TValue
The lower bound of the range.

to
Type: TValue
The upper bound of the range.

rangeKind
Type: `Moq:::Range`
The kind of range. See `Range`. 
**Type Parameters**

TValue
Type of the argument to check.
Examples

The following example shows how to expect a method call with an integer argument within the 0..100 range.

```c#
mock.Expect(x => x.HasInventory(
    It.IsAny<string>(),
    It.IsInRange(0, 100, Range.Inclusive)))
  .Returns(false);
```
See Also

It Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members
Moq

It.IsRegex Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsRegex(String)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
<tr>
<td>IsRegex(String, RegexOptions)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
</tbody>
</table>
See Also

It Class
It Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Matches a string argument if it matches the given regular expression pattern.

**Namespace:**  Moq  
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

    public static string IsRegex(
        string regex
    )

Parameters

regex
    Type: String
    The pattern to use to match the string argument value.
Examples

The following example shows how to expect a call to a method where the string argument matches the given regular expression:

C#

```csharp
mock.Expect(x => x.Check(It.IsRegex("[a-z]+"))).Returns(1);
```
See Also

It Class
IsRegex Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Moq

It...:..IsRegex Method (, )

Matches a string argument if it matches the given regular expression pattern.

**Namespace:**  Moq
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public static string IsRegex(
    string regex,
    RegexOptions options
)

Parameters

regex
    Type: String
    The pattern to use to match the string argument value.

options
    Type: RegexOptions
    The options used to interpret the pattern.
Examples

The following example shows how to expect a call to a method where the string argument matches the given regular expression, in a case insensitive way:

C#

```csharp
mock.Expect(x => x.Check(It.IsRegex("[a-z]+", RegexOptions.IgnoreCase))).Returns(
```
See Also

It Class
IsRegex Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Marks a method as a matcher, which allows complete replacement of the built-in `It` class with your own argument matching rules.

**Namespace:**  [Moq](https://github.com/Moq)

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
## Syntax

C#

```csharp
public class MatcherAttribute : Attribute
```
Remarks

The argument matching is used to determine whether a concrete invocation in the mock matches a given expectation. This matching mechanism is fully extensible.

There are two parts of a matcher: the compiler matcher and the runtime matcher.

- Compiler matcherUsed to satisfy the compiler requirements for the argument. Needs to be a method optionally receiving any arguments you might need for the matching, but with a return type that matches that of the argument.

Let's say I want to match a lists of orders that contains a particular one. I might create a compiler matcher like the following:

```csharp
public static class Orders
{
    [Matcher]
    public static IEnumerable<Order> Contains(Order order) {
        return null;
    }
}
```

Now we can invoke this static method instead of an argument in an invocation:

```csharp
var order = new Order { ... };
var mock = new Mock<IRepository<Order>>();
mock.Expect(x => x.Save(Orders.Contains(order))).Throws<ArgumentException>();
```

Note that the return value from the compiler matcher is irrelevant. This method will never be called, and is just used to satisfy the compiler and to signal Moq that this is not a method that we want to be invoked at runtime.
- Runtime matcher The runtime matcher is the one that will actually perform evaluation when the test is run, and is defined by convention to have the same signature as the compiler matcher, but where the return value is the first argument to the call, which contains the object received by the actual invocation at runtime:

```csharp
public static bool Contains(IEnumerable<Order> orders, Order order)
{
    return orders.Contains(order);
}
```

At runtime, the mocked method will be invoked with a specific list of orders. This value will be passed to this runtime matcher as the first argument, while the second argument is the one specified in the expectation (x.Save(Orders.Contains(order))).

The boolean returned determines whether the given argument has been matched. If all arguments to the expected method are matched, then the expectation is verified.

Using this extensible infrastructure, you can easily replace the entire `It` set of matchers with your own. You can also avoid the typical (and annoying) lengthy expressions that result when you have multiple arguments that use generics.
Examples

The following is the complete example explained above:

```csharp
public static class Orders
{
    [Matcher]
    public static IEnumerable<Order> Contains(Order order)
    {
        return null;
    }

    public static bool Contains(IEnumerable<Order> orders, Order order)
    {
        return orders.Contains(order);
    }
}

And the concrete test using this matcher:

```csharp
var order = new Order { ... };
var mock = new Mock<IRepository<Order>>();

mock.Expect(x => x.Save(Orders.Contains(order))).Throws<ArgumentException>();

// use mock, invoke Save, and have the matcher filter.
```
Inheritance Hierarchy

Object
Attribute
   Moq...MatcherAttribute
See Also

MatcherAttribute Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MatcherAttribute` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MatcherAttribute</td>
<td>Initializes a new instance of the MatcherAttribute class</td>
</tr>
</tbody>
</table>
See Also

MatcherAttribute Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MatcherAttribute Constructor

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public MatcherAttribute()
See Also

MatcherAttribute Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Base class for mocks and static helper class with methods that apply to mocked objects, such as `Get<(Of <(T)>)(T)` to retrieve a `Mock<(Of <(T)>))` from an object instance.

**Namespace:**  Moq

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public abstract class Mock : IMock
Inheritance Hierarchy

Object
Moq::Mock

Moq::Mock<Of<^T>>
See Also

Mock Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `Mock` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌷 Mock</td>
<td>Initializes the mock</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CreateEventHandler</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>Get(Of {(T)&gt;)</code></td>
<td>Retrieves the mock object for the given object instance.</td>
</tr>
<tr>
<td><code>GetObject</code></td>
<td>Returns the mocked object value.</td>
</tr>
<tr>
<td><code>Verify</code></td>
<td>Implements <code>VerifyQQQ</code>.</td>
</tr>
<tr>
<td><code>VerifyAll</code></td>
<td>Implements <code>VerifyAllQQQ</code>.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior</td>
<td>Implements Behavior.</td>
</tr>
<tr>
<td>CallBase</td>
<td>Implements CallBase.</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Implements DefaultValue.</td>
</tr>
<tr>
<td>ImplementedInterfaces</td>
<td>Exposes the list of extra interfaces implemented by the mock.</td>
</tr>
<tr>
<td>Object</td>
<td>The mocked object instance. Implements Object.</td>
</tr>
</tbody>
</table>
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

Mock Constructor

*Mock Class*  *See Also*  *Send Feedback*

Initializes the mock

**Namespace:**  *Moq*

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

protected Mock()
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The **Mock** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateEventHandler</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Get&lt;(Of &lt;(T)&gt;)&gt;</td>
<td>Retrieves the mock object for the given object instance.</td>
</tr>
<tr>
<td>GetObject</td>
<td>Returns the mocked object value.</td>
</tr>
<tr>
<td>Verify</td>
<td>Implements <code>VerifyQQQ</code>.</td>
</tr>
<tr>
<td>VerifyAll</td>
<td>Implements <code>VerifyAllQQQ</code>.</td>
</tr>
</tbody>
</table>
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateEventHandler&lt;(Of &lt;(TEventArgs)&gt;)&gt;</td>
<td>Implements CreateEventHandler&lt;(Of &lt;(TEventArgs)&gt;)&gt;</td>
</tr>
<tr>
<td>CreateEventHandler()</td>
<td>Implements CreateEventHandler()</td>
</tr>
</tbody>
</table>
See Also

Mock Class
Mock Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock..:::CreateEventHandler(Of <(TEventArgs)> ) Method
Mock Class  See Also  Send Feedback
Implements CreateEventHandler(Of <(TEventArgs)> )()().

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
C#

```csharp
public MockedEvent<TEventArgs> CreateEventHandler<TEventArgs>()
where TEventArgs : EventArgs
```
Type Parameters

TEventArgs
  Type of event argument class.
See Also

Mock Class
CreateEventHandler Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock...::.CreateEventHandler Method

Mock Class  See Also  Send Feedback

Implements CreateEventHandler()()

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public MockedEvent<EventArgs> CreateEventHandler()
See Also

Mock Class
CreateEventHandler Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Retrieves the mock object for the given object instance.

**Namespace:** [Moq](https://moq.codeplex.com)

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public static IMock<T> Get<T>(
    T mocked
)  
where T : class

Parameters

mocked
    Type: T
    The instance of the mocked object.
Type Parameters

**T**

Type of the mock to retrieve. Can be omitted as it's inferred from the object instance passed in as the mocked instance.

Return Value

The mock associated with the mocked object.
The following example shows how to add a new expectation to an object instance which is not the original `Mock<(Of <(T)>)>` but rather the object associated with it:

C#  

```csharp
// Typed instance, not the mock, is retrieved from some test API.  
HttpContextBase context = GetMockContext();

// context.Request is the typed object from the "real" API  
// so in order to add an expectation to it, we need to get  
// the mock that "owns" it  
Mock<HttpRequestBase> request = Mock.Get(context.Request);
mock.Expect(req => req.AppRelativeCurrentExecutionFilePath)  
  .Returns(tempUrl);
```
# Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>The received mocked instance was not created by Moq.</td>
</tr>
</tbody>
</table>
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Returns the mocked object value.

Namespace: **Moq**
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

protected abstract Object GetObject()
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Mock:::Verify Method

Mock Class  See Also  Send Feedback

Implements Verify().

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public abstract void Verify()

Implements

IMock::<Verify>()
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Mock
Mock...:::VerifyAll Method

Mock Class  See Also  Send Feedback

Implements VerifyAll().

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public abstract void VerifyAll()

Implements

IMock::<>.VerifyAll()()
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `Mock` type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior</td>
<td>Implements <a href="#">Behavior</a>.</td>
</tr>
<tr>
<td>CallBase</td>
<td>Implements <a href="#">CallBase</a>.</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Implements <a href="#">DefaultValue</a>.</td>
</tr>
<tr>
<td>ImplementedInterfaces</td>
<td>Exposes the list of extra interfaces implemented by the mock.</td>
</tr>
<tr>
<td>Object</td>
<td>The mocked object instance. Implements <a href="#">Object</a>.</td>
</tr>
</tbody>
</table>
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock...::Behavior Property

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public MockBehavior Behavior { get; internal set; }
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock..::.CallBase Property

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public bool CallBase { get; set; }
```

Implements

IMock::<CallBase
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock::DefaultValue Property

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public DefaultValue DefaultValue { get; set; }

Implements

IMock...::DefaultValue
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock::ImplementedInterfaces Property

Exposes the list of extra interfaces implemented by the mock.

**Namespace:** Moq

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

protected internal List<Type> ImplementedInterfaces { get; private ;}
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The mocked object instance. Implements Object.

**Namespace:** Moq

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public Object Object { get; }

Implements

IMock::<Object>
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock<(Of <(T)>)> Class

Core implementation of the IMock<(Of <(T)>)> interface.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public class Mock<T> : Mock, IMock<T>
where T : class
```
Type Parameters

T
    Type to mock.
Inheritance Hierarchy

Object
*Moq..::.Mock*
  Moq..::.Mock<Of <(T)>>
See Also

Mock<Of<(T)>>() Members
Moq Namespace
Moq::::IMock<Of<(T)>>()

Send comments on this topic to moqdisc@googlegroups.com
The `Mock(Of T)`) type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mock&lt;()&lt;()&gt;&lt;()</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>As&lt;Of (TInterface)&gt;()</td>
<td>Implements As&lt;Of (TInterface)&gt;()().</td>
</tr>
<tr>
<td>CreateEventHandler&lt;Of (TEventArgs)&gt;()</td>
<td>Implements CreateEventHandler&lt;Of (TEventArgs)&gt;(). () (Inherited from Mock.).</td>
</tr>
<tr>
<td>Expect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ExpectGet&lt;Of (TProperty)&gt;()</td>
<td>Implements ExpectGet&lt;Of (TProperty)&gt;() (Expression&lt;Of ((Func&lt;Of (T, TProperty)&gt;())&gt;)).</td>
</tr>
<tr>
<td>ExpectSet</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetObject</td>
<td>Returns the mocked object value. (Overrides Mock::&lt;::GetObject()()).</td>
</tr>
<tr>
<td>Verify</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>VerifyAll</td>
<td>Implements VerifyAll(). (Overrides Mock::&lt;::VerifyAll()).</td>
</tr>
<tr>
<td>VerifyGet&lt;Of (TProperty)&gt;()</td>
<td>Implements VerifyGet&lt;Of (TProperty)&gt;() (Expression&lt;Of ((Func&lt;Of (T, TProperty)&gt;())&gt;)).</td>
</tr>
<tr>
<td>VerifySet</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior</td>
<td>Implements <strong>Behavior</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Mock</strong>.)</td>
</tr>
<tr>
<td>CallBase</td>
<td>Implements <strong>CallBase</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Mock</strong>.)</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Implements <strong>DefaultValue</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Mock</strong>.)</td>
</tr>
<tr>
<td>ImplementedInterfaces</td>
<td>Exposes the list of extra interfaces implemented by the mock.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Mock</strong>.)</td>
</tr>
<tr>
<td>Object</td>
<td>Exposes the mocked object instance.</td>
</tr>
</tbody>
</table>
See Also

Mock\<(Of \<(T)>\)> Class  
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock(Of T) Constructor

Mock(Of T) Class

See Also

Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mock&lt;(Of (T)&gt;)]()()</td>
<td>Initializes an instance of the mock with default behavior.</td>
</tr>
<tr>
<td>Mock&lt;(Of (T)&gt;)](MockBehavior)</td>
<td>Initializes an instance of the mock with the specified behavior.</td>
</tr>
<tr>
<td>Mock&lt;(Of (T)&gt;)](array&lt;Object&gt;[][])</td>
<td>Initializes an instance of the mock with default behavior and with the given constructor arguments for the class. (Only valid when T is a class)</td>
</tr>
<tr>
<td>Mock&lt;(Of (T)&gt;)](MockBehavior, array&lt;Object&gt;[][])</td>
<td>Initializes an instance of the mock with a specific behavior with the given constructor arguments for the class.</td>
</tr>
</tbody>
</table>
See Also

Mock<(Of <(T)>)> Class
Mock<(Of <(T)>)> Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock\<(\text{Of}<\text{T}\text{)}>\) Constructor

Initialize an instance of the mock with default behavior.

\textbf{Namespace:} Moq
\textbf{Assembly:} Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public Mock()
Examples

C#

```csharp
var mock = new Mock<IFormatProvider>();
```
See Also

Mock<(Of <(T)>)> Class
Mock<(Of <(T)>)> Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

Mock(Of (T)>) Constructor (MockBehavior)

Mock(Of (T)> Class  Example  See Also  Send Feedback

Initializes an instance of the mock with the specified behavior.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public Mock(
    MockBehavior behavior
)
```

Parameters

behavior
  Type: `Moq::MockBehavior`
  Behavior of the mock.
var mock = new Mock<IFormatProvider>(MockBehavior.Relaxed);
See Also

Mock(Of (T)>) Class
Mock(Of (T)>) Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Moq

Mock<(Of <(T)>)> Constructor (array<>[]()[])

Mock<(Of <(T)>)> Class  Example  See Also  Send Feedback

Initializes an instance of the mock with default behavior and with the given constructor arguments for the class. (Only valid when T is a class)

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public Mock(
    params Object[] args
)
```

Parameters

args
  Type: array<Object>[]
  Optional constructor arguments if the mocked type is a class.
Remarks

The mock will try to find the best match constructor given the constructor arguments, and invoke that to initialize the instance. This applies only for classes, not interfaces.
Examples

C#

```csharp
var mock = new Mock<MyProvider>(someArgument, 25);
```
See Also

Mock(Of T) Class
Mock(Of T) Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

Mock<

Mock<

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public Mock(
    MockBehavior behavior,
    params Object[] args
)
```

Parameters

behavior
  Type: `MockBehavior`
  Behavior of the mock.

args
  Type: `array< Object >[]`[
  Optional constructor arguments if the mocked type is a class.
Remarks

The mock will try to find the best match constructor given the constructor arguments, and invoke that to initialize the instance. This applies only to classes, not interfaces.
Examples

C#

var mock = new Mock<MyProvider>(someArgument, 25);
See Also

Mock(Of T>) Class
Mock(Of T>) Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `Mock<Of <(T)>)` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>As&lt;Of (TInterface)&gt;()</code></td>
<td>Implements <code>As&lt;Of (TInterface)&gt;()()()</code>.</td>
</tr>
<tr>
<td><code>CreateEventHandler&lt;Of (EventArgs)&gt;()</code></td>
<td>Implements <code>CreateEventHandler&lt;Of (EventArgs)&gt;()()()</code>. (Inherited from <code>Mock</code>).</td>
</tr>
<tr>
<td><code>Expect</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>ExpectGet&lt;Of (TProperty)&gt;()</code></td>
<td>Implements <code>ExpectGet&lt;Of (TProperty)&gt;()</code> (Expression&lt;Of &lt;(Func&lt;Of &lt;(T, TProperty)&gt;)&gt;)&gt;)).</td>
</tr>
<tr>
<td><code>ExpectSet</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>GetObject</code></td>
<td>Returns the mocked object value. (Overrides <code>Mock::GetObject()()()</code>.</td>
</tr>
<tr>
<td><code>Verify</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>VerifyAll</code></td>
<td>Implements <code>VerifyAll()()()</code>. (Overrides <code>Mock::VerifyAll()()()</code>.</td>
</tr>
<tr>
<td><code>VerifyGet&lt;Of (TProperty)&gt;()</code></td>
<td>Implements <code>VerifyGet&lt;Of (TProperty)&gt;()</code> (Expression&lt;Of &lt;(Func&lt;Of &lt;(T, TProperty)&gt;)&gt;)&gt;)).</td>
</tr>
<tr>
<td><code>VerifySet</code></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

Mock<(Of <(T)>)> Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

C#

```csharp
public virtual IMock<TInterface> As<TInterface>()
where TInterface : class
```
Type Parameters

TInterface
  Type of interface to cast the mock to.
See Also

Mock<(Of <(T)>)> Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members Include Inherited Members Moq

Mock<(Of <(T)>)>...:::Expect Method Mock<(Of <(T)>)> Class See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Expect(Expression&lt;(Of &lt;(Action&lt;(Of &lt;(T)&gt;)&gt;)&gt;))</code></td>
<td>Implements <code>Expect(Expression&lt;(Of &lt;(Action&lt;(Of &lt;(T)&gt;)&gt;)&gt;))</code>.</td>
</tr>
<tr>
<td><code>Expect&lt;(Of &lt;(TResult)&gt;)(Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TResult)&gt;)&gt;)&gt;&gt;))</code></td>
<td>Implements <code>Expect&lt;(Of &lt;(TResult)&gt;)(Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TResult)&gt;)&gt;)&gt;&gt;))</code>.</td>
</tr>
</tbody>
</table>
See Also

Mock(Of (T)>) Class
Mock(Of (T)>) Members
Mock Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq
Mock<(Of <(T)>)>...: Expect Method (<(Of <<(Of <(T)>)>)>))
Mock<(Of <(T)>)> Class  See Also  Send Feedback
Implements Expect(Expression<(Of <(Action<(Of <(T)>)>)>))).

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
### Syntax

**C#**

```csharp
public IExpect Expect(
    Expression<Action<T>> expression
)
```

### Parameters

**expression**
Type: `Expression<Action<T>>`
Lambda expression that specifies the expected method invocation.

### Implements

```csharp
IMock<(Of <(T)>).;.;; Expect<(Of <(TResult)>))(Expression<(Of <(Func<(Of <(T, TResult)>)>))>)
```
See Also

Mock(Of T>) Class
Expect Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Moq
Mock(Of T>).::.Expect(Of (TResult)>) Method (Of (Of (T, TResult)>)>))
Mock(Of T>) Class  See Also  Send Feedback
Implements Expect(Of (TResult)>(Expression(Of (Func(Of (T, TResult)>)>)>)).

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

```csharp
public IExpect<TResult> Expect<TResult>(
    Expression<Func<T, TResult>> expression
)
```

**Parameters**

expression
Type: Expression(Of (Func(Of (T, TResult))))>
Lambda expression that specifies the expected method invocation.
Type Parameters

TResult
Type of the return value. Typically omitted as it can be inferred from the expression.
See Also

Mock(Of T>) Class
Expect Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
Mock<Of<(T)>).::.ExpectGet<Of<(TProperty)>>) Method
Mock<Of<(T)>>) Class  See Also  Send Feedback
Implements ExpectGet<Of<(TProperty)>>(Expression<Of<(Func<Of<(Of<(T,
TProperty)>)>)>))>.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
### Syntax

**C#**

```csharp
public IExpectGetter<TProperty> ExpectGet<TProperty>(
  Expression<Func<T, TProperty>> expression
)
```

#### Parameters

- **expression**
  - Type: `Expression<Func<T, TProperty>>`
  - Lambda expression that specifies the expected property getter.
Type Parameters

TProperty
Type of the property. Typically omitted as it can be inferred from the expression.
See Also

Mock(Of (T)>) Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock<Of<(T)>>).::..ExpectSet Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ExpectSet&lt;(Of &lt;(TProperty)&gt;)&gt; (Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;))</code></td>
<td>Implements <code>ExpectSet&lt;(Of &lt;(TProperty)&gt;)&gt; (Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;), TProperty)</code>.</td>
</tr>
<tr>
<td><code>ExpectSet&lt;(Of &lt;(TProperty)&gt;)&gt; (Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;), TProperty)&gt;)</code></td>
<td>Implements <code>ExpectSet&lt;(Of &lt;(TProperty)&gt;)&gt; (Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;), TProperty)&gt;), TProperty)</code>.</td>
</tr>
</tbody>
</table>
See Also

Mock<(Of <(T)>)> Class
Mock<(Of <(T)>)> Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

Mock<(T)> Method (Mock<(T, TProperty)>)

Mock<(T)> Class

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

```csharp
public IExpectSetter<TProperty> ExpectSet<TProperty>(
    Expression<Func<T, TProperty>> expression
)
```

**Parameters**

- `expression`
  - Type: `Expression<((Func<(Func<T, TProperty>))>)>`
  - Lambda expression that specifies the expected property setter.
Type Parameters

TProperty
Type of the property. Typically omitted as it can be inferred from the expression.
See Also

Mock(Of(Of(T)>)>) Class
ExpectSet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
Mock<Of (T)>.:::.ExpectSet<Of (TProperty)> Method (Of (Of (T, TProperty))), TProperty)
Mock<Of (T)>:: Class  See Also  Send Feedback

Implements ExpectSet<Of (TProperty)>(Expression<Of (Func<Of (T, TProperty)>)), TProperty>.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public IExpectSetter<TPropertyValue> ExpectSet<TPropertyValue>(
    Expression<Func<T, TPropertyValue>> expression,
    TPropertyValue value
)
```

**Parameters**

**expression**
Type: Expression<(Of <(Func<(Of <(T, TPropertyValue)>)>))>
Lambda expression that specifies the expected property setter.

**value**
Type: TPropertyValue
The value expected to be set for the property.
Type Parameters

TProperty
Type of the property. Typically omitted as it can be inferred from the expression.
See Also

Mock(Of T) Class
ExpectSet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Returns the mocked object value.

**Namespace:**  [Moq](https://github.com/Moq/Moq)

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

protected override Object GetObject()
See Also

Mock(Of T) Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock<(Of <(T)>>)...::Verify Method
Mock<(Of <(T)>)> Class
See Also
Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Verify()</code></td>
<td>Implements <code>Verify()</code>. (Overrides <code>Mock::&lt;Mock::VerifyQQ</code>).</td>
</tr>
<tr>
<td><code>Verify(Expression&lt;Of (Action&lt;Of &lt;T&gt;&gt;)&gt;)</code></td>
<td>Implements <code>Verify(Expression&lt;Of (Action&lt;Of &lt;T&gt;&gt;)&gt;)).</code></td>
</tr>
<tr>
<td><code>Verify&lt;(Of &lt;(TResult)&gt;)(Expression&lt;(Of &lt;(Func&lt;Of &lt;(T, TResult)&gt;)&gt;))&gt;</code></td>
<td>Implements <code>Verify&lt;(Of &lt;(TResult)&gt;)(Expression&lt;(Of &lt;(Func&lt;Of &lt;(T, TResult)&gt;)&gt;)))).</code></td>
</tr>
</tbody>
</table>
See Also

Mock(Of (T)>)
Mock(Of (T)>) Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock<(Of <(T)>)> Class

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public override void Verify()

Implements

IMock<(Of <(T)>)>..::..Verify()
IMock[..::..Verify()}
See Also

Mock<(Of <(T)>)> Class
Verify Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Moq
Mock<(T)>::Verify Method (<(Of <(Of <(T)>)>)>)
Mock<(T)> Class  See Also  Send Feedback
Implements Verify(Expression<(Of <(Action<(Of <(T)>)>)>)>).

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
## Syntax

**C#**

```csharp
public virtual void Verify(
    Expression<Action<T>> expression
)
```

### Parameters

- **expression**
  - Type: `Expression<Action<T>>`
  - Expression to verify.

### Implements

- `IMock<(Of <(T)>)>.IMock<((Of <(Action<(Of <(T)>)>)))>.Verify(Expression<(Of <(Action<(Of <(T)>)>))>)`
See Also

Mock<(Of (T)>)> Class
Verify Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
```csharp
using Moq;

Mock<
(Where T)
>
.VERIFY<
(When TResult)
>
Method
(Where T, TResult)
;

Mock<
(Where T)
>
Class
;

See Also
;
Send Feedback

Implements
Verify<
(When TResult)
>
<Expression<
(Where Func<
(With T, TResult)
>
>
>
>
>

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
```
## Syntax

**C#**

```csharp
public virtual void Verify<TResult>(
    Expression<Func<T, TResult>> expression
)
```

### Parameters

**expression**

Type: `Expression<(Of <(Func<(Of `<(T, TResult)>))>>))>

Expression to verify.
Type Parameters

TResult
Type of return value from the expression.

Implements

IMock(Of (T)>),..,Verify(Expression<(Of <(Action<(Of (T)>)>))>))
See Also

Mock<(Of <(T)>)> Class
Verify Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

Mock(Of (T)>).::.VerifyAll Method

Mock(Of (T)>) Class  See Also  Send Feedback

Implements VerifyAll().

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public override void VerifyAll()

Implements

IMock(Of T)...:::VerifyAll
IMock:::VerifyAll


See Also

Mock<Of<(T)>) Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
```
C#
Moq
Mock<(Of <(T)>).::.VerifyGet<(Of <(T)Property)>)> Method
Mock<(Of <(T)>)> Class  See Also  Send Feedback

Implements VerifyGet<(Of <(T)Property)>)(Expression<(Of <(Func<(Of <(T, TProperty)>)>)>)>).

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
```
Syntax

C#

public virtual void VerifyGet<TProperty>(
    Expression<Func<T, TProperty>> expression
)

Parameters

equation
  Type: Expression<(Of ((Of (Of (T, TProperty)>)>))>)
  Expression to verify.
Type Parameters

TProperty
  Type of the property to verify. Typically omitted as it can be inferred from the expression's return type.

Implements

IMock<(Of <(T)>).::.VerifyGet<(Of <(TProperty)>))(Expression<(Of <(Func<(Of <(T, TProperty)>)>)>))>
See Also

Mock<(Of <(T)>)> Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock(Of (T)>) ...::: VerifySet Method

Mock(Of (T)>) Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerifySet&lt;(Of &lt;(TProperty)&gt;)(Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;))&gt;</td>
<td>Implements VerifySet&lt;(Of &lt;(TProperty)&gt;)(Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;), TProperty)&gt;</td>
</tr>
<tr>
<td>VerifySet&lt;(Of &lt;(TProperty)&gt;)(Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;), TProperty)&gt;))&gt;</td>
<td>Implements VerifySet&lt;(Of &lt;(TProperty)&gt;)(Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;), TProperty)&gt;</td>
</tr>
</tbody>
</table>
See Also

Mock<Of<(T)>) Class
Mock<Of<(T)>) Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

Mock<Of <(T)>>::VerifySet<Of <(TProperty)>>() Method <(Of <(Of <(Of <(T, TProperty)>)>)>)>)

Mock<Of <(T)>>() Class  See Also  Send Feedback

Implements VerifySet<Of <(TProperty)>>()(Expression<Of <(Func<Of <(Of <(T, TProperty)>)>)>)>).

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public virtual void VerifySet<TProperty>(
    Expression<Func<T, TProperty>> expression
)

Parameters

expression
    Type: Expression<(Of <(Func<(Of <(T, TProperty)>)>))>)
    Expression to verify.
Type Parameters

TProperty
  Type of the property to verify. Typically omitted as it can be inferred from
  the expression's return type.

Implements

IMock(Of (T)>).:::VerifySet(Of (TProperty>)(Expression(Of
  (Func(Of (T, TProperty>))>))
See Also

Mock(Of (T)> ) Class
VerifySet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

Mock<Of <(T)>).::.VerifySet<Of <(TProperty)>>) Method (<<Of <<-((Of <(T, TProperty)>)))>>), TProperty)

Mock<Of <(T)>>) Class  See Also  Send Feedback

Implements VerifySet<Of <(TProperty)>>(Expression<Of <<(Func<<Of <(T, TProperty)>)>>>>), TProperty).

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

C#

```csharp
public virtual void VerifySet<TProperty>(
    Expression<Func<T, TProperty>> expression,
    TProperty value
)
```

**Parameters**

**expression**
Type: Expression<(Of <(Func<(Of <(T, TProperty)>))>)>)
Expression to verify.

**value**
Type: TProperty
The value that should have been set on the property.
Type Parameters

TProperty

Type of the property to verify. Typically omitted as it can be inferred from the expression's return type.

Implements

IMock<Of<(T)>...::VerifySet<Of<(TProperty)>>(Expression<Of<br.Func<Of<(T, TProperty)>>>), TProperty)
See Also

Mock(Of T) Class
VerifySet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `Mock<Of<T>>` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior</td>
<td>Implements <a href="#">Behavior</a>. (Inherited from <a href="#">Mock</a>.)</td>
</tr>
<tr>
<td>CallBase</td>
<td>Implements <a href="#">CallBase</a>. (Inherited from <a href="#">Mock</a>.)</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Implements <a href="#">DefaultValue</a>. (Inherited from <a href="#">Mock</a>.)</td>
</tr>
<tr>
<td>ImplementedInterfaces</td>
<td>Exposes the list of extra interfaces implemented by the mock.</td>
</tr>
<tr>
<td>Object</td>
<td>Exposes the mocked object instance.</td>
</tr>
</tbody>
</table>
See Also

Mock<(Of <(T)>)> Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Exposes the mocked object instance.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public T Object { get; }

Implements

IMock(Of (Of T)<>).::Object
See Also

Mock(Of T>) Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockBehavior Enumeration

Options to customize the behavior of the mock.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public enum MockBehavior
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strict</td>
<td>Causes the mock to always throw an exception for invocations that don't have a corresponding expectation. Will never throw exceptions, returning default values when necessary (null for reference types, zero for value types or empty enumerables and arrays).</td>
</tr>
<tr>
<td>Loose</td>
<td>Default mock behavior, which equals Loose.</td>
</tr>
<tr>
<td>Default</td>
<td>Default mock behavior, which equals Loose.</td>
</tr>
</tbody>
</table>
See Also

Mpq Namespace

Send comments on this topic to mqdisc@googlegroups.com
MockedEvent Class

Represented a generic event that has been mocked and can be raised.

**Namespace:**  Moq

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C# public class MockedEvent
Inheritance Hierarchy

Object

Moq:::MockedEvent

Moq:::MockedEvent<(Of <(TEventArgs)>)>
See Also

MockedEvent Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockedEvent` type exposes the following members.
## Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Provides support for attaching a MockedEvent to a generic EventHandler event.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Raised</td>
<td>Event raised whenever the mocked event is raised.</td>
</tr>
</tbody>
</table>
See Also

MockedEvent Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockedEvent` type exposes the following members.
## Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Provides support for attaching a <a href="#">MockedEvent</a> to a generic <code>EventHandler</code> event.</td>
</tr>
</tbody>
</table>
See Also

MockedEvent Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockedEvent...:::Implicit Operator

MockedEvent Class  See Also  Send Feedback

Provides support for attaching a MockedEvent to a generic EventHandler event.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public static implicit operator EventHandler (MockedEvent mockEvent)

Parameters

mockEvent
  Type: Moq::MockedEvent
  Event to convert.
See Also

MockedEvent Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The MockedEvent type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raised</td>
<td>Event raised whenever the mocked event is raised.</td>
</tr>
</tbody>
</table>
See Also

MockedEvent Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockedEvent...:::Raised Event

MockedEvent Class  See Also  Send Feedback

Event raised whenever the mocked event is raised.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public event EventHandler Raised
See Also

MockedEvent Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Provides a typed `MockedEvent` for a specific type of `EventArgs`.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public class MockedEvent<TEventArgs> : MockedEvent
where TEventArgs : EventArgs
```
Type Parameters

TEventArgs

The type of event arguments required by the event.
Remarks

The mocked event can either be a EventHandler or custom event handler which follows .NET practice of providing object sender, EventArgs args kind of signature.
Inheritance Hierarchy

Object

Moq::MockedEvent

Moq::MockedEvent<((TEventArgs))>
See Also

MockedEvent<\(\text{Of } \{\text{TEventArgs}\}\)> Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockedEvent<(Of <(TEventArgs)>)>`) type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raise</td>
<td>Raises the associated event with the given event argument data.</td>
</tr>
</tbody>
</table>
## Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Provides support for attaching a <code>MockedEvent&lt;...(TEventArgs)&gt;&gt;</code> to a generic EventHandler event.</td>
</tr>
</tbody>
</table>
# Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raised</td>
<td>Event raised whenever the mocked event is raised. (Inherited from MockedEvent.)</td>
</tr>
</tbody>
</table>
See Also

MockedEvent<(Of <(TEventArgs)>)> Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockedEvent<Of <(TEventArgs)>>` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raise</td>
<td>Raises the associated event with the given event argument data.</td>
</tr>
</tbody>
</table>
## Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Provides support for attaching a \texttt{MockedEvent&lt;\text{Of \text{TEventArgs}&gt;}} to a generic \texttt{EventHandler} event.</td>
</tr>
</tbody>
</table>
See Also

MockedEvent<(Of <(TEventArgs)>)> Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Mock
MockedEvent<Of <(TEventArgs)>>).::.Implicit Operator
MockedEvent<Of <(TEventArgs)>>) Class  See Also  Send Feedback
Provides support for attaching a MockedEvent<Of <(TEventArgs)>>) to a
generic EventHandler event.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public static implicit operator EventHandler<TEventArgs> ( 
    MockedEvent<TEventArgs> mockEvent
 )
```

Parameters

mockEvent

Type: `MockedEvent<TEventArgs>(Of (TEventArgs))`
Event to convert.
See Also

MockedEvent<(Of <(TEventArgs)>)> Class

Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Raises the associated event with the given event argument data.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
### Syntax

**C#**

```
public void Raise(
    TEventArgs args
)
```

### Parameters

- **args**
  - **Type:** `TEventArgs`
  - Data to pass to the event.
See Also

MockedEvent<(Of <(TEventArgs)>)> Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockedEvent<Of <(TEventArgs)>>` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raised</td>
<td>Event raised whenever the mocked event is rised. (Inherited from MockedEvent.)</td>
</tr>
</tbody>
</table>
See Also

MockedEvent<(Of <(TEventArgs)>)> Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Exception thrown by mocks when expectations are not met, the mock is not properly setup, etc.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

[SerializableAttribute]
public class MockException : Exception
Remarks

A distinct exception type is provided so that exceptions thrown by the mock can be differentiated in tests that expect other exceptions to be thrown (i.e. ArgumentException).

Richer exception hierarchy/types are not provided as tests typically should not catch or expect exceptions from the mocks. These are typically the result of changes in the tested class or its collaborators implementation, and result in fixes in the mock setup so that they disappear and allow the test to pass.
Inheritance Hierarchy

Object
  Exception
    Moq....MockException
See Also

MockException Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockException` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MockException</strong></td>
<td>Supports the serialization infrastructure.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Supports the serialization infrastructure. (Overrides ExceptionGetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
</tbody>
</table>


See Also

MockException Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockException Constructor

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
## Syntax

**C#**

```csharp
protected MockException(
    SerializationInfo info,
    StreamingContext context
)
```

### Parameters

- **info**
  - Type: `SerializationInfo`
  - Serialization information.

- **context**
  - Type: `StreamingContext`
  - Streaming context.
See Also

MockException Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.GetObjectData</td>
<td>Supports the serialization infrastructure. (Overrides ExceptionGetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
</tbody>
</table>
See Also

MockException Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockException..::.GetObjectData Method

MockException Class  See Also  Send Feedback

Supports the serialization infrastructure.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

**C#**

```csharp
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)
```

**Parameters**

- `info`  
  Type: `SerializationInfo`  
  Serialization information.

- `context`  
  Type: `StreamingContext`  
  Streaming context.

**Implements**

- `ISerializableGetObjectData(SerializationInfo, StreamingContext)`
- `_ExceptionGetObjectData(SerializationInfo, StreamingContext)`
See Also

MockException Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockFactory Class

Utility factory class to use to construct multiple mocks when consistent verification is desired for all of them.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public class MockFactory
Remarks

If multiple mocks will be created during a test, passing the desired MockBehavior (if different than the Default or the one passed to the factory constructor) and later verifying each mock can become repetitive and tedious.

This factory class helps in that scenario by providing a simplified creation of multiple mocks with a default MockBehavior (unless overriden by calling Create<Of <(T)>)(MockBehavior)) and posterior verification.
Examples

The following is a straightforward example on how to create and automatically verify strict mocks using a MockFactory:

```csharp
var factory = new MockFactory(MockBehavior.Strict);

var foo = factory.Create<IFoo>();
var bar = factory.Create<IBar>();

// no need to call Verifiable() on the expectation
// as we'll be validating all expectations anyway.
foo.Expect(f => f.Do());
bar.Expect(b => b.Redo());

// exercise the mocks here
factory.VerifyAll();
// At this point all expectations are already checked
// and an optional MockException might be thrown.
// Note also that because the mocks are strict, any invocation
// that doesn't have a matching expectation will also throw a MockEx
```

The following examples shows how to setup the factory to create loose mocks and later verify only verifiable expectations:

```csharp
var factory = new MockFactory(MockBehavior.Loose);

var foo = factory.Create<IFoo>();
var bar = factory.Create<IBar>();

// this expectation will be verified at the end of the "using" block
foo.Expect(f => f.Do()).Verifiable();

// this expectation will NOT be verified
foo.Expect(f => f.Calculate());

// this expectation will be verified at the end of the "using" block
bar.Expect(b => b.Redo()).Verifiable();
```
// exercise the mocks here
// note that because the mocks are Loose, members
// called in the interfaces for which no matching
// expectations exist will NOT throw exceptions,
// and will rather return default values.

factory.Verify();
// At this point verifiable expectations are already checked
// and an optional MockException might be thrown.

The following examples shows how to setup the factory with a default strict
behavior, overriding that default for a specific mock:

C#

```csharp
var factory = new MockFactory(MockBehavior.Strict);

// this particular one we want loose
var foo = factory.Create<IFoo>(MockBehavior.Loose);
var bar = factory.Create<IBar>();

// set expectations

// exercise the mocks here

factory.Verify();
```
Inheritance Hierarchy

Object
Moq..:..MockFactory
See Also

MockFactory Members
Moq Namespace
Moq::MockBehavior

Send comments on this topic to moqdisc@googlegroups.com
The `MockFactory` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MockFactory</td>
<td>Initializes the factory with the given defaultBehavior for newly created mocks from the factory.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>CreateMock&lt;(Of&lt;T&gt;)&gt;</td>
<td>Implements creation of a new mock within the factory.</td>
</tr>
<tr>
<td>Verify</td>
<td>Verifies all verifiable expectations on all mocks created by this factory.</td>
</tr>
<tr>
<td>VerifyAll</td>
<td>Verifies all verifiable expectations on all mocks created by this factory.</td>
</tr>
<tr>
<td>VerifyMocks</td>
<td>Invokes verifyAction for each mock in Mocks, and accumulates the resulting MockVerificationException that might be thrown from the action.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallBase</td>
<td>Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to true (True in Visual Basic).</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Specifies the behavior to use when returning default values for unexpected invocations on loose mocks.</td>
</tr>
<tr>
<td>Mocks</td>
<td>Gets the mocks that have been created by this factory and that will get verified together.</td>
</tr>
</tbody>
</table>
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockFactory Constructor

Initializes the factory with the given defaultBehavior for newly created mocks from the factory.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public MockFactory(
  MockBehavior defaultBehavior
)

Parameters

defaultBehavior
  Type: Moq::MockBehavior
  The behavior to use for mocks created using the Create(Of <T>)() overload.
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockFactory` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>CreateMock&lt;Of&lt;T&gt;&gt;</td>
<td>Implements creation of a new mock within the factory.</td>
</tr>
<tr>
<td>Verify</td>
<td>Verifies all verifiable expectations on all mocks created by this factory.</td>
</tr>
<tr>
<td>VerifyAll</td>
<td>Verifies all verifiable expectations on all mocks created by this factory.</td>
</tr>
<tr>
<td>VerifyMocks</td>
<td>Invokes verifyAction for each mock in <code>Mocks</code>, and accumulates the resulting MockVerificationException that might be thrown from the action.</td>
</tr>
</tbody>
</table>
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members Include Inherited Members Moq MockFactory...:::Create Method MockFactory Class See Also Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Create&lt;(Of &lt;(T)&gt;)&gt;(){}</code></td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time.</td>
</tr>
<tr>
<td><code>Create&lt;(Of &lt;(T)&gt;)&gt; (MockBehavior)</code></td>
<td>Creates a new mock with the given behavior.</td>
</tr>
<tr>
<td><code>Create&lt;(Of &lt;(T)&gt;)&gt; (array&lt;Object&gt;[][])</code></td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time and with the given constructor arguments for the class.</td>
</tr>
<tr>
<td><code>Create&lt;(Of &lt;(T)&gt;)&gt; (MockBehavior, array&lt;Object&gt;[][])</code></td>
<td>Creates a new mock with the given behavior and with the given constructor arguments for the class.</td>
</tr>
</tbody>
</table>
See Also

MockFactory Class
MockFactory Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

MockFactory..::.Create(Of (T)> ) Method

MockFactory Class  Example  See Also  Send Feedback

Creates a new mock with the default MockBehavior specified at factory construction time.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public Mock<T> Create<T>()
where T : class
Type Parameters

T
Type to mock.

Return Value
A new `Mock<(Of <(T)>).`
Examples

C#

```csharp
var factory = new MockFactory(MockBehavior.Strict);
var foo = factory.Create<IFoo>();
// use mock on tests
factory.VerifyAll();
```
See Also

MockFactory Class
Create Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq
MockFactory...:::Create<(Of <(T)>)> Method (MockBehavior)

MockFactory Class  Example  See Also  Send Feedback

Creates a new mock with the given behavior.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
### Syntax

**C#**

```csharp
public Mock<T> Create<T>(
    MockBehavior behavior
)
where T : class
```

### Parameters

- **behavior**
  - Type: `Moq::MockBehavior`
  - Behavior to use for the mock, which overrides the default behavior specified at factory construction time.
Type Parameters

T
  Type to mock.

Return Value

A new `Mock<Of<T>>`. 
Examples

The following example shows how to create a mock with a different behavior to that specified as the default for the factory:

C#

```csharp
var factory = new MockFactory(MockBehavior.Strict);
var foo = factory.Create<IFoo>(MockBehavior.Loose);
```
See Also

MockFactory Class
Create Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

MockFactory..::.Create(Of (T)>)) Method (array<>[][])  
MockFactory Class  Example  See Also  Send Feedback

Creates a new mock with the default MockBehavior specified at factory construction time and with the the given constructor arguments for the class.

Namespace:  Moq  
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public Mock<T> Create<T>(
    params Object[] args
)
where T : class

Parameters

args
    Type: array<Object>[]()
    Constructor arguments for mocked classes.
Type Parameters

T
  Type to mock.

Return Value

A new Mock<Of <(T)>>.
Remarks

The mock will try to find the best match constructor given the constructor arguments, and invoke that to initialize the instance. This applies only to classes, not interfaces.
**Examples**

C#  

```csharp
var factory = new MockFactory(MockBehavior.Default);

var mock = factory.Create<MyBase>("Foo", 25, true);
// use mock on tests

factory.Verify();
```
See Also

MockFactory Class
Create Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
MockFactory...:::Create<(Of <(T)>)> Method (MockBehavior, array<>[])[]
MockFactory Class  Example  See Also  Send Feedback

Creates a new mock with the given behavior and with the given constructor arguments for the class.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
# Syntax

**C#**

```csharp
public Mock<T> Create<T>(
    MockBehavior behavior,
    params Object[] args
)
```

**Parameters**

**behavior**

Type: `MockBehavior`

Behavior to use for the mock, which overrides the default behavior specified at factory construction time.

**args**

Type: `array<Object>[]`[]

Constructor arguments for mocked classes.
Type Parameters

T
  Type to mock.

Return Value

A new `Mock<Of<(T)>`. 
Remarks

The mock will try to find the best match constructor given the constructor arguments, and invoke that to initialize the instance. This applies only to classes, not interfaces.
Examples

The following example shows how to create a mock with a different behavior to that specified as the default for the factory, passing constructor arguments:

C#

```csharp
var factory = new MockFactory(MockBehavior.Default);

var mock = factory.Create<MyBase>(MockBehavior.Strict, "Foo", 25, true);
```
See Also

MockFactory Class
Create Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockFactory..::.CreateMock<(Of <(T)>)> Method

 Implements creation of a new mock within the factory.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
protected virtual Mock<T> CreateMock<T>(
    MockBehavior behavior,
    Object[] args
)
where T : class
```

Parameters

behavior
   Type: Moq::MockBehavior
   The behavior for the new mock.

args
   Type: array<Object>[]
   Optional arguments for the construction of the mock.
Type Parameters

T
  Type to mock.
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockFactory..::.Verify Method

Verifies all verifiable expectations on all mocks created by this factory.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public virtual void Verify()
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Moq::MockException</code></td>
<td>One or more mocks had expectations that were not satisfied.</td>
</tr>
</tbody>
</table>
See Also

MockFactory Class
Moq Namespace
Mock<Of<(T)>>.:::Verify()()
C#

Moq
MockFactory..::.VerifyAll Method

Moq
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)

Verifies all verifiable expectations on all mocks created by this factory.
Syntax

C#

public virtual void VerifyAll()
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Moq::MockException</code></td>
<td>One or more mocks had expectations that were not satisfied.</td>
</tr>
</tbody>
</table>
See Also

MockFactory Class
Moq Namespace
Mock<Of <(T)>).::.Verify()()
C# Moq MockFactory..::.VerifyMocks Method

MockFactory Class  See Also  Send Feedback

Invokes verifyAction for each mock in Moqs, and accumulates the resulting MockVerificationException that might be thrown from the action.

Namespace:  Moq
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

protected virtual void VerifyMocks(
    Action<IMock> verifyAction
)

Parameters

verifyAction
    Type: Action<(Of <(IMock)>))
    The action to execute against each mock.
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The MockFactory type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallBase</td>
<td>Whether the base member virtual implementation will be called for mocked</td>
</tr>
<tr>
<td></td>
<td>classes if no expectation is met.</td>
</tr>
<tr>
<td></td>
<td>Defaults to true (True in Visual Basic).</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Specifies the behavior to use when returning default values for unexpected</td>
</tr>
<tr>
<td></td>
<td>invocations on loose mocks.</td>
</tr>
<tr>
<td>Mocks</td>
<td>Gets the mocks that have been created by this factory and that will get</td>
</tr>
<tr>
<td></td>
<td>verified together.</td>
</tr>
</tbody>
</table>
See Also

MockFactory Class  
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to true (True in Visual Basic).

**Namespace:** Moq

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public bool CallBase { get; set; }
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockFactory...:::DefaultValue Property

MockFactory Class  See Also  Send Feedback

Specifies the behavior to use when returning default values for unexpected invocations on loose mocks.

Namespace:  Moq  
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public DefaultValue DefaultValue { get; set; }
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

MockFactory..::.Mocks Property

**MockFactory Class**  **See Also**  **Send Feedback**

Gets the mocks that have been created by this factory and that will get verified together.

**Namespace:**  **Moq**

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

protected IEnumerable<IMock> Mocks { get; }
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Kind of range to use in a filter specified through `IsInRange<Of <(TValue)>>(TValue, TValue, Range)`. 

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public enum Range
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inclusive</td>
<td>The range includes the to and from values.</td>
</tr>
<tr>
<td>Exclusive</td>
<td>The range does not include the to and from values.</td>
</tr>
</tbody>
</table>
See Also

Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
This namespace defines the interfaces that contain the methods available in Moq fluent API, such as Expect, Callback, Returns, Throws, etc.

Review the documentation of the available methods on these interfaces for code examples. The way these interfaces are composed, grouped and made visible at different stages during an expectation (i.e. Verifiable is the last "verb" and can't be specified before the Returns) is internal but the API will naturally lead you to the proper way of using it, so don't worry too much about who (and where) exposes these language interfaces. They will show up in Intellisense when it's appropriate.

Do use their documentation to learn about options (especially when there are several overloads available) that may be a better fit for a particular scenario.
# Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICallback</td>
<td>Defines the Callback verb and overloads.</td>
</tr>
<tr>
<td>ICallback&lt;(Of&lt;TResult&gt;)&gt;</td>
<td>Defines the Callback verb and overloads for callbacks on expectations that return a value.</td>
</tr>
<tr>
<td>ICallbackGetter&lt;(Of&lt;TProperty&gt;)&gt;</td>
<td>Defines the Callback verb for property getter expectations.</td>
</tr>
<tr>
<td>ICallbackSetter&lt;(Of&lt;TProperty&gt;)&gt;</td>
<td>Defines the Callback verb for property setter expectations.</td>
</tr>
<tr>
<td>INever</td>
<td>Defines the Never verb.</td>
</tr>
<tr>
<td>IOccurrence</td>
<td>Defines occurrence members to constraint expectations.</td>
</tr>
<tr>
<td>IRaise</td>
<td>Defines the Raises verb.</td>
</tr>
</tbody>
</table>
| IReturns                   | Base interface for IReturns<(Of<TResult>)>.
| IReturns<(Of<TResult>)>    | Defines the Returns verb.                                                  |
| IReturnsGetter<(Of<TProperty>)> | Defines the Returns verb for property get expectations.                   |
| IThrows                    | Defines the Throws verb.                                                   |
| IVerifies                  | Defines the Verifiable verb.                                               |

Send comments on this topic to moqdisc@googlegroups.com
ICallback Interface

**Namespace:** Moq.Language

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)

Defines the Callback verb and overloads.
Public interface ICallback
See Also

ICallback Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members
Moq
ICallback Members

ICallback Interface  Methods  See Also  Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

ICallback Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members
Include Inherited Members
Moq ICallback Methods
ICallback Interface  See Also  Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

ICallback Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Callback(Action)</code></td>
<td>Specifies a callback to invoke when the method is called.</td>
</tr>
<tr>
<td><code>Callback&lt;(Of &lt;(T)&gt;)&gt;(Action&lt;(Of &lt;(T)&gt;)&gt;)</code></td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td><code>Callback&lt;(Of &lt;(T1, T2, T3)&gt;)&gt;(Action&lt;(Of &lt;(T1, T2, T3)&gt;)&gt;)</code></td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td><code>Callback&lt;(Of &lt;(T1, T2, T3, T4)&gt;)&gt;(Action&lt;(Of &lt;(T1, T2, T3, T4)&gt;)&gt;)</code></td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
</tbody>
</table>
See Also

ICallback Interface
ICallback Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback Callback Method ()

ICallback Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the method is called.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

ICallbackResult Callback(
    Action callback
)

Parameters

callback
    Type: Action
    Callback method to invoke.
Examples

The following example specifies a callback to set a boolean value that can be used later:

```csharp
bool called = false;
mock.Expect(x => x.Execute())
    .Callback(() => called = true);
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback..::.Callback(Of (T)>)) Method ((Of (T)>))
ICallback Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the method is called that receives the original arguments.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

ICallbackResult Callback<T>(
    Action<T> callback
)

Parameters

callback
    Type: Action<(Of <(T)>))
    Callback method to invoke.
Type Parameters

`T`

Argument type of the invoked method.
Examples

Invokes the given callback with the concrete invocation argument value.

Notice how the specific string argument is retrieved by simply declaring it as part of the lambda expression for the callback:

C#

```csharp
mock.Expect(x => x.Execute(It.IsAny<string>()))
    .Callback((string command) => Console.WriteLine(command));
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
ICallback...::Callback<Of <(T1, T2)>> Method (<(Of <(T1, T2)>>))
ICallback Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the method is called that receives the original arguments.

**Namespace:**  [Moq.Language](#)  
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

ICallbackResult Callback<T1, T2>(
    Action<T1, T2> callback
)

Parameters

callback
    Type: Action<(Of <(T1, T2)>))
    Callback method to invoke.
Type Parameters

T1
Type of the first argument of the invoked method.

T2
Type of the second argument of the invoked method.
Examples

Invokes the given callback with the concrete invocation arguments values.

Notice how the specific arguments are retrieved by simply declaring them as part of the lambda expression for the callback:

C#
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback::Callback<Of<(T1, T2, T3)>> Method (<(Of<((T1, T2, T3))>))

ICallback Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the method is called that receives the original arguments.

**Namespace:**  Moq.Language

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

ICallbackResult Callback<T1, T2, T3>(
    Action<T1, T2, T3> callback
)

Parameters

callback
    Type: Action<(Of <(T1, T2, T3)>))
    Callback method to invoke.
Type Parameters

T1
Type of the first argument of the invoked method.

T2
Type of the second argument of the invoked method.

T3
Type of the third argument of the invoked method.
Examples

Invokes the given callback with the concrete invocation arguments values.

Notice how the specific arguments are retrieved by simply declaring them as part of the lambda expression for the callback:

```csharp
mock.Expect(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<int>())
    .Callback((string arg1, string arg2, int arg3) => Console.WriteLine(arg1 + arg2 + arg3));
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback...:.Callback<(Of <(T1, T2, T3, T4)>)> Method (<(Of <(T1, T2, T3, T4)>)>)

ICallback Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the method is called that receives the original arguments.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

ICallbackResult Callback<T1, T2, T3, T4>(
    Action<T1, T2, T3, T4> callback
)

Parameters

callback
    Type: Action<(Of ((T1, T2, T3, T4)>)>)
    Callback method to invoke.
Type Parameters

T1
Type of the first argument of the invoked method.

T2
Type of the second argument of the invoked method.

T3
Type of the third argument of the invoked method.

T4
Type of the fourth argument of the invoked method.
Examples

Invokes the given callback with the concrete invocation arguments values.

Notice how the specific arguments are retrieved by simply declaring them as part of the lambda expression for the callback:

```c#
mock.Expect(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<int>(),
    It.IsAny<bool>()))
    .Callback((string arg1, string arg2, int arg3, bool arg4) => Console.WriteLine(arg1 + arg2 + arg3 + arg4));
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
IViewHolder interface

Represents a control that displays a view.

Members | See Also | Send Feedback

Defined in `Moq` (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

C#

```csharp
public interface ICallback<TResult>
```
Type Parameters

TResult
 Type of the return value of the expectation.
See Also

ICallback<(Of <(TResult)>)> Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members

Moq

ICallback(Of TResult>) Members

ICallback(Of TResult>) Interface  Methods  See Also  Send Feedback
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

ICallback<(Of <(TResult)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members
Include Inherited Members
Moq
ICallback<((TResult)>>) Methods
ICallback<((TResult)>>) Interface  See Also  Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

ICallback<(Of <(TResult)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
- Include Protected Members
- Include Inherited Members
Moq
ICallback(Of TResult)

ICallback(Of TResult) Interface  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Callback(Action)</strong></td>
<td>Specifies a callback to invoke when the method is called.</td>
</tr>
<tr>
<td><strong>Callback&lt;(Of &lt;(T)&gt;)&gt; (Action&lt;(Of &lt;(T)&gt;)&gt;)</strong></td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td><strong>Callback&lt;(Of &lt;(T1, T2)&gt;)&gt; (Action&lt;(Of &lt;(T1, T2)&gt;)&gt;)</strong></td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td><strong>Callback&lt;(Of &lt;(T1, T2, T3)&gt;)(Action&lt;(Of &lt;(T1, T2, T3)&gt;)&gt;))</strong></td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td><strong>Callback&lt;(Of &lt;(T1, T2, T3, T4)&gt;)(Action&lt;(Of &lt;(T1, T2, T3, T4)&gt;)&gt;))</strong></td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
</tbody>
</table>
See Also

ICallback<(Of <(TResult)>)> Interface
ICallback<(Of <(TResult)>)> Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq
ICallback<Of<TResult>>).:.:.Callback Method()
ICallback<Of<TResult>>) Interface Example See Also Send Feedback

Specifies a callback to invoke when the method is called.

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IReturnsThrows<TResult> Callback(
    Action callback
)

Parameters

callback
    Type: Action
    Callback method to invoke.
Examples

The following example specifies a callback to set a boolean value that can be used later:

C#

```csharp
bool called = false;
mock.Expect(x => x.Execute())
  .Callback(() => called = true)
  .Returns(true);
```

Note that in the case of value-returning methods, after the Callback call you can still specify the return value.
See Also

ICallback<(Of <(TResult>)>) Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback<(Of <(TResult)>)> ... Callback<(Of <(T)>)> Method <(Of <(T)>)>)
ICallback<(Of <(TResult)>)> Interface Example See Also Send Feedback

Specifies a callback to invoke when the method is called that receives the original arguments.

**Namespace:** Moq.Language
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IReturnsThrows<TResult> Callback<T>(
    Action<T> callback
);

Parameters

callback
    Type: Action(Of (Of (T)>))
    Callback method to invoke.
Type Parameters

T
Type of the argument of the invoked method.
Examples

Invokes the given callback with the concrete invocation argument value.

Notice how the specific string argument is retrieved by simply declaring it as part of the lambda expression for the callback:

C#  

```csharp
mock.Expect(x => x.Execute(It.IsAny<string>())
  .Callback((string command) => Console.WriteLine(command))
  .Returns(true);
```
See Also

ICallback(Of TResult) Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback<(Of<TResult>)>...: Callback<(Of<(T1, T2)>)> Method<(Of<(T1, T2)>)>

ICallback<(Of<TResult>)> Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the method is called that receives the original arguments.

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IReturnsThrows<TResult> Callback<T1, T2>(
    Action<T1, T2> callback
)

Parameters

callback
    Type: Action<(Of<(T1, T2)>))
    Callback method to invoke.
Type Parameters

T1
Type of the first argument of the invoked method.

T2
Type of the second argument of the invoked method.
**Examples**

Invokes the given callback with the concrete invocation arguments values.

Notice how the specific arguments are retrieved by simply declaring them as part of the lambda expression for the callback:

```csharp
mock.Expect(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>()))
    .Callback((string arg1, string arg2) => Console.WriteLine(arg1 + arg2))
    .Returns(true);
```
See Also

ICallback<(Of <(TResult)>)> Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback<(Of <(TResult)>)>...Callback<(Of <(T1, T2, T3)>)> Method <(Of <(T1, T2, T3)>))

ICallback<(Of <(TResult)>)> Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the method is called that receives the original arguments.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

C#

```csharp
IReturnsThrows<TResult> Callback<T1, T2, T3>(
    Action<T1, T2, T3> callback
)
```

**Parameters**

callback
  Type: `Action<(Of <(T1, T2, T3)>)>`
  Callback method to invoke.
Type Parameters

T1
Type of the first argument of the invoked method.

T2
Type of the second argument of the invoked method.

T3
Type of the third argument of the invoked method.
Examples

Invokes the given callback with the concrete invocation arguments values.

Notice how the specific arguments are retrieved by simply declaring them as part of the lambda expression for the callback:

C#  

```csharp
mock.Expect(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<int>()))
  .Callback((string arg1, string arg2, int arg3) => Console.WriteLine(arg1 + arg2 + arg3))
  .Returns(true);
```
See Also

ICallback<(Of <(TResult)>)> Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback(Of TResult)> ... Callback(Of T1, T2, T3, T4>)> Method (Of (T1, T2, T3, T4>))

ICallback(Of TResult>) Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the method is called that receives the original arguments.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IReturnsThrows<TResult> Callback<T1, T2, T3, T4>(
    Action<T1, T2, T3, T4> callback
)

Parameters

callback
    Type: Action<(Of <(T1, T2, T3, T4)>))
    Callback method to invoke.
Type Parameters

T1
Type of the first argument of the invoked method.

T2
Type of the second argument of the invoked method.

T3
Type of the third argument of the invoked method.

T4
Type of the fourth argument of the invoked method.
Examples

Invokes the given callback with the concrete invocation arguments values.

Notice how the specific arguments are retrieved by simply declaring them as part of the lambda expression for the callback:

```csharp
mock.Expect(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<int>(),
    It.IsAny<bool>())
    .Callback(((string arg1, string arg2, int arg3, bool arg4) => Console.WriteLine(arg1 + arg2 + arg3 + arg4))
    .Returns(true);
```
See Also

ICallback<(Of <(TResult)>)> Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Defines the Callback verb for property getter expectations.

**Namespace:**  [Moq.Language](#)

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface ICallbackGetter<TProperty>
Type Parameters

TProperty
   Type of the property.
See Also

ICallbackGetter<(Of <(TProperty)>)> Members
Moq.Language Namespace
Mock<(Of <(T)>),..::ExpectGet<(Of <(TProperty)>)(Expression<(Of
<Func<(Of <(T, TProperty)>)>>)>))

Send comments on this topic to moqdisc@googlegroups.com
The `ICallbackGetter<Of <(TProperty)>)` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback</td>
<td>Specifies a callback to invoke when the property is retrieved.</td>
</tr>
</tbody>
</table>
See Also

ICallbackGetter(Of TProperty) Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The ICallbackGetter(Of TProperty) type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌟 Callback</td>
<td>Specifies a callback to invoke when the property is retrieved.</td>
</tr>
</tbody>
</table>
See Also

ICallbackGetter<(Of <(TProperty)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallbackGetter<(Of <(TProperty)>)>..::.Callback Method

ICallbackGetter<(Of <(TProperty)>)> Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the property is retrieved.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

```csharp
IReturnsThrowsGetter<TProperty> Callback(
    Action callback
)
```

**Parameters**

callback
   - Type: Action
   - Callback method to invoke.
Examples

Invokes the given callback with the property value being set.

C#

```csharp
mock.ExpectGet(x => x.Suspended)
    .Callback(() => called = true)
    .Returns(true);
```
See Also

ICallbackGetter<(Of <(TProperty>>>)) Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallbackSetter<(Of <(TProperty)>)> Interface

Defines the Callback verb for property setter expectations.

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface ICallbackSetter<TProperty>
Type Parameters

TProperty
   Type of the property.
See Also

ICallbackSetter(Of <TProperty>) Members

Moq.Language Namespace

IMock(Of <T>).ExpectSet(Of <TProperty>)(Expression(Of Func(Of <T, TProperty>), TProperty))

Send comments on this topic to moqdisc@googlegroups.com
The `ICallbackSetter<Of <(TProperty)▻)` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback</td>
<td>Specifies a callback to invoke when the property is set that receives the property value being set.</td>
</tr>
</tbody>
</table>
See Also

ICallbackSetter<(Of <(TProperty)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The ICallbackSetter<(Of <(TProperty)>)> type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback</td>
<td>Specifies a callback to invoke when the property is set that receives the property value being set.</td>
</tr>
</tbody>
</table>
See Also

ICallbackSetter<(Of <(TProperty)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallbackSetter<(Of <(TProperty)>)>..::.Callback Method

ICallbackSetter<(Of <(TProperty)>)> Interface  Example  See Also  Send Feedback

Specifies a callback to invoke when the property is set that receives the property value being set.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

ICallbackResult Callback(
    Action<TProperty> callback
)

Parameters

callback
    Type: Action<(Of <TProperty>)>
    Callback method to invoke.
Examples

Invokes the given callback with the property value being set.

C#

```csharp
mock.ExpectSet(x => x.Suspended)
    .Callback((bool state) => Console.WriteLine(state));
```
See Also

ICallbackSetter<(Of (TProperty)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Moq

INever Interface

Defines the Never verb.

**Namespace:** Moq.Language

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface INever
See Also

INever Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The INever type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td>The expected invocation is never expected to happen.</td>
</tr>
</tbody>
</table>
See Also

INever Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The \texttt{INever} type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td>The expected invocation is never expected to happen.</td>
</tr>
</tbody>
</table>
See Also

INever Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The expected invocation is never expected to happen.

**Namespace:**  Moq.Language
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

void Never();
Remarks

Never()() is always verified immediately as the invocations are performed, like strict mocks do with unexpected invocations.
Examples

C#

```csharp
var mock = new Mock<ICommand>();
mock.Expect(foo => foo.Execute("ping"))
    .Never();
```
See Also

INever Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Defines occurrence members to constraint expectations.

**Namespace:** [Moq.Language](#)  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface IOccurrence
See Also

IOccurrence Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `IOccurrence` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtMost</td>
<td>The expected invocation can happen at most specified number of times.</td>
</tr>
<tr>
<td>AtMostOnce</td>
<td>The expected invocation can happen at most once.</td>
</tr>
</tbody>
</table>
See Also

IOccurrence Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The **IOccurrence** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtMost</td>
<td>The expected invocation can happen at most specified number of times.</td>
</tr>
<tr>
<td>AtMostOnce</td>
<td>The expected invocation can happen at most once.</td>
</tr>
</tbody>
</table>
See Also

IOccurrence Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The expected invocation can happen at most specified number of times.

**Namespace:** Moq.Language

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
IVerifies AtMost(
    int callCount
)
```

Parameters

callCount
    Type: Int32

[Missing <param name="callCount"/> documentation for "M:Moq.Language.IOccurrence.AtMost(System.Int32)" ]
Examples

C#

```csharp
var mock = new Mock<ICommand>();
mock.Expect(foo => foo.Execute("ping"))
    .AtMost(5);
```
See Also

IOccurrence Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The expected invocation can happen at most once.

**Namespace:** [Moq.Language](#)  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

`IVerifies AtMostOnce()`
Examples

C#

```csharp
var mock = new Mock<ICommand>();
mock.Expect(foo => foo.Execute("ping"))
    .AtMostOnce();
```
See Also

IOccurrence Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Defines the Raises verb.

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface IRaise
See Also

IRaise Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌹 <strong>Raises</strong></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

IRaise Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Include Protected Members
Include Inherited Members
Moq
IRaise Methods

IRaise Interface  See Also  Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raises</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

IRaise Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members
Moq

IRaise...:::::: Raises Method

IRaise Interface  See Also  Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raises(MockedEvent, EventArgs)</td>
<td>Specifies the mocked event that will be raised when the expectation is met.</td>
</tr>
<tr>
<td>Raises(MockedEvent, Func&lt;(Of EventArgs)&gt;)</td>
<td>Specifies the mocked event that will be raised when the expectation is met.</td>
</tr>
<tr>
<td>Raises&lt;(Of (T)&gt;)(MockedEvent, Func&lt;(Of (T, EventArgs)&gt;)&gt;)</td>
<td>Specifies the mocked event that will be raised when the expectation is met.</td>
</tr>
<tr>
<td>Raises&lt;(Of (T1, T2)&gt;)(MockedEvent, Func&lt;(Of (T1, T2, EventArgs)&gt;)&gt;)</td>
<td>Specifies the mocked event that will be raised when the expectation is met.</td>
</tr>
<tr>
<td>Raises&lt;(Of (T1, T2, T3)&gt;)(MockedEvent, Func&lt;(Of (T1, T2, T3, EventArgs)&gt;)&gt;)</td>
<td>Specifies the mocked event that will be raised when the expectation is met.</td>
</tr>
<tr>
<td>Raises&lt;(Of (T1, T2, T3, T4)&gt;)(MockedEvent, Func&lt;(Of (T1, T2, T3, T4, EventArgs)&gt;)&gt;)</td>
<td>Specifies the mocked event that will be raised when the expectation is met.</td>
</tr>
</tbody>
</table>
See Also

IRaise Interface
IRaise Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
IRaise...::: Raises Method (MockedEvent, )

Namespaces: Moq, Language
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)

Specifies the mocked event that will be raised when the expectation is met.
Syntax

C#

```csharp
IVerifies Raises(
    MockedEvent eventHandler,
    EventArgs args
)
```

Parameters

eventHandler
    Type: Moq::MockedEvent
    The mocked event, retrieved from CreateEventHandler() or
    CreateEventHandler<Of <(TEventArgs)>>.

args
    Type: EventArgs
    The event args to pass when raising the event.
Examples

The following example shows how to set an expectation that will raise an event when it's met:

```csharp
var mock = new Mock<IContainer>();
// create handler to associate with the event to raise
var handler = mock.CreateEventHandler();
// associate the handler with the event to raise
mock.Object.Added += handler;
// set the expectation and the handler to raise
mock.Expect(add => add.Add(It.IsAny<string>(), It.IsAny<object>()))
    .Raises(handler, EventArgs.Empty);
```
See Also

IRaise Interface
Raises Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

IRaise:::Raises Method (MockedEvent, <(Of <(>)>))

IRaise Interface  See Also  Send Feedback

Specifies the mocked event that will be raised when the expectation is met.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
### Syntax

C#

```csharp
IVerifies Raises(
    MockedEvent eventHandler,
    Func<EventArgs> func
)
```

### Parameters

**eventHandler**

Type: `Moq...MockedEvent`

The mocked event, retrieved from `CreateEventHandler()` or `CreateEventHandler<TEventArgs>()`.

**func**

Type: `Func<EventArgs>`

A function that will build the `EventArgs` to pass when raising the event.
See Also

IRaise Interface
Raises Overload
Moq.Language Namespace
IRaise::::Raise(MockedEvent, EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
C# Moq
IRaise...:::Raises<(Of <(T)>>) Method (MockedEvent, <(Of <(T, )>)>)
IRaise Interface  See Also  Send Feedback

Specifies the mocked event that will be raised when the expectation is met.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

```csharp
IVerifies Raises<T>(
    MockedEvent eventHandler,
    Func<T, EventArgs> func
)
```

**Parameters**

**eventHandler**
Type: `Moq::MockedEvent`
The mocked event, retrieved from `CreateEventHandler()` or `CreateEventHandler<>()`.

**func**
Type: `Func<Of <(T, EventArgs)>)`
A function that will build the EventArgs to pass when raising the event.
Type Parameters

\( T \)

Type of the argument received by the expected invocation.
See Also

IRaise Interface
Raises Overload
Moq.Language Namespace
IRaise:::Rises(MockedEvent, EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

IRaise...:::Raises<(Of <(T1, T2)>)> Method (MockedEvent, <(Of <(T1, T2, >)>))

IRaise Interface  See Also  Send Feedback

Specifies the mocked event that will be raised when the expectation is met.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IVerifies Raises<T1, T2>(
    MockedEvent eventHandler,
    Func<T1, T2, EventArgs> func
)

Parameters

eventHandler
    Type: Moq::MockedEvent
    The mocked event, retrieved from CreateEventHandler() or CreateEventHandler<(Of <(TEventArgs)>)>().

func
    Type: Func<(Of <(T1, T2, EventArgs)>))
    A function that will build the EventArgs to pass when raising the event.
Type Parameters

T1
Type of the first argument received by the expected invocation.

T2
Type of the second argument received by the expected invocation.
See Also

IRaise Interface
Raises Overload
Moq.Language Namespace
IRaise::<::Raises(MockedEvent, EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

IRaise...:::::Raises<(Of <(T1, T2, T3)>)) Method (MockedEvent, <(Of <(T1, T2, T3, >)>))

IRaise Interface  See Also  Send Feedback

Specifies the mocked event that will be raised when the expectation is met.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

**IVerifies** Raises<T1, T2, T3>(
  [MockedEvent] eventHandler,
  Func<T1, T2, T3, EventArgs> func
)

Parameters

eventHandler
  Type: `Mock<..MockedEvent`
  The mocked event, retrieved from `CreateEventHandler()` or `CreateEventHandler<Of<(TEventArgs)>>()`.  

func
  Type: `Func<(Of<(T1, T2, T3, EventArgs)>))`
  A function that will build the EventArgs to pass when raising the event.
Type Parameters

T1
Type of the first argument received by the expected invocation.

T2
Type of the second argument received by the expected invocation.

T3
Type of the third argument received by the expected invocation.
See Also

IRaise Interface
 Raises Overload
 Moq.Language Namespace
 IRaise::<:: Raises(MockedEvent, EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
IRaise...:::Raises<(Of <(T1, T2, T3, T4)>)> Method (MockedEvent, <(Of <(T1, T2, T3, T4, )>))

IRaise Interface  See Also  Send Feedback

Specifies the mocked event that will be raised when the expectation is met.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
IVerifies Raises<T1, T2, T3, T4>(
    MockedEvent eventHandler,
    Func<T1, T2, T3, T4, EventArgs> func
)
```

Parameters

**eventHandler**
Type: `Moq..::MockedEvent`
The mocked event, retrieved from `CreateEventHandler()` or `CreateEventHandler<Of<TEventArgs>>()`.

**func**
Type: `Func<Of<(T1, T2, T3, T4, EventArgs)>)`
A function that will build the EventArgs to pass when raising the event.
Type Parameters

T1
Type of the first argument received by the expected invocation.

T2
Type of the second argument received by the expected invocation.

T3
Type of the third argument received by the expected invocation.

T4
Type of the fourth argument received by the expected invocation.
See Also

IRaise Interface
Raises Overload
Moq.Language Namespace
IRaise:::Raises(MockedEvent, EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

IReturns Interface

See Also  Send Feedback

Base interface for IReturns<(Of <(TResult)>).

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface IReturns
See Also

Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
IReturns<(Of <(TResult)>)> Interface

Defines the Returns verb.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface IReturns<TResult> : IReturns
Type Parameters

TResult
  Type of the return value from the expression.
See Also

IReturns<(Of <(TResult)>)> Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Include Protected Members
Include Inherited Members
Moq
IReturns<(Of <(TResult)>>) Members

IReturns<(Of <(TResult)>>) Interface  Methods  See Also  Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

IReturns<(Of <(TResult)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members
Moq

IReturns<Of <(TResult)>)> Methods

IReturns<Of <(TResult)>)> Interface  See Also  Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

IReturns<(Of <(TResult)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Include Protected Members
Include Inherited Members
Moq
IReturns<((TResult))>::Returns Method
IReturns<((TResult))> Interface
See Also
Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Returns(Func&lt;Of&lt;TResult&gt;&gt;)</strong></td>
<td>Specifies a function that will calculate the value to return from the method.</td>
</tr>
<tr>
<td><strong>Returns&lt;Of&lt;(T)&gt;(Func&lt;Of&lt;((T, TResult)&gt;)&gt;)</strong></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><strong>Returns&lt;Of&lt;(T1, T2)&gt;&gt;(Func&lt;Of&lt;((T1, T2, TResult)&gt;)&gt;)</strong></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><strong>Returns&lt;Of&lt;(T1, T2, T3)&gt;&gt;(Func&lt;Of&lt;((T1, T2, T3, TResult)&gt;)&gt;)</strong></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><strong>Returns&lt;Of&lt;(T1, T2, T3, T4)&gt;&gt;(Func&lt;Of&lt;((T1, T2, T3, T4, TResult)&gt;)&gt;)</strong></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><strong>Returns(TResult)</strong></td>
<td>Specifies the value to return.</td>
</tr>
</tbody>
</table>
See Also

IReturns<(Of (TResult)>>) Interface
IReturns<(Of (TResult)>>) Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies a function that will calculate the value to return from the method.

**Namespace:**  Moq.Language  
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IReturnsResult Returns(
    Func<TResult> valueFunction
)

Parameters

valueFunction
    Type: Func<(Of <(TResult)>))
    The function that will calculate the return value.
Examples

Return a calculated value when the method is called:

C#?

```csharp
mock.Expect(x => x.Execute("ping"))
  .Returns(() => returnValues[0]);
```

The lambda expression to retrieve the return value is lazy-executed, meaning that its value may change depending on the moment the method is executed and the value the returnValues array has at that moment.
See Also

IReturns<(Of <(TResult)>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

IReturns<(Of <(TResult)>)>.::.Returns<(Of <(T)>)> Method (<(Of <(T, TResult)>)>)

IReturns<(Of <(TResult)>)> Interface  Example  See Also  Send Feedback

Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
## Syntax

### C#

```csharp
IReturnsResult Returns<T>(
    Func<T, TResult> valueFunction
)
```

### Parameters

valueFunction
- **Type:** Func<(Of <(T, TResult)>)
- The function that will calculate the return value.
Type Parameters

T

Type of the argument of the invoked method.
Examples

Return a calculated value which is evaluated lazily at the time of the invocation.

The lookup list can change between invocations and the expectation will return different values accordingly. Also, notice how the specific string argument is retrieved by simply declaring it as part of the lambda expression:

```csharp
mock.Expect(x => x.Execute(It.IsAny<string>()))
    .Returns((string command) => returnValues[command]);
```
See Also

IReturns<(Of <(TResult)>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.

**Namespace:** [Moq.Language](#)  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

```csharp
IReturnsResult Returns<T1, T2>(
    Func<T1, T2, TResult> valueFunction
)
```

**Parameters**

valueFunction
- Type: Func<(Of <(T1, T2, TResult>>))
- The function that will calculate the return value.
Type Parameters

T1
Type of the first argument of the invoked method.

T2
Type of the second argument of the invoked method.
Examples

Return a calculated value which is evaluated lazily at the time of the invocation.

The return value is calculated from the value of the actual method invocation arguments. Notice how the arguments are retrieved by simply declaring them as part of the lambda expression:

C#  

```csharp
mock.Expect(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>()))
    .Returns((string arg1, string arg2) => arg1 + arg2);
```
See Also

IReturns<(Of <(TResult)>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.

**Namespace:**  [Moq.Language](#)  
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IReturnsResult Returns<T1, T2, T3>(
    Func<T1, T2, T3, TResult> valueFunction
)

Parameters

valueFunction
    Type: Func<(Of <(T1, T2, T3, TResult)>)>
    The function that will calculate the return value.
Type Parameters

T1  
Type of the first argument of the invoked method.

T2  
Type of the second argument of the invoked method.

T3  
Type of the third argument of the invoked method.
Examples

Return a calculated value which is evaluated lazily at the time of the invocation.

The return value is calculated from the value of the actual method invocation arguments. Notice how the arguments are retrieved by simply declaring them as part of the lambda expression:

```csharp
mock.Expect(x => x.Execute(  
    It.IsAny<string>(),  
    It.IsAny<string>(),  
    It.IsAny<int>())  
  .Returns((string arg1, string arg2, int arg3) => arg1 + arg2 + a
```
See Also

**I\text{Returns}<(\text{Of }<(\text{TResult}>))>\text{ Interface}\n**

**Returns Overload**

**Moq.Language Namespace**

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

IReturns<(Of <(TResult)>)>..:::Returns<(Of <(T1, T2, T3, T4)>)> Method

((Of <(T1, T2, T3, T4, TResult)>))

IReturns<(Of <(TResult)>)> Interface Example See Also Send Feedback

Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IReturnsResult Returns<T1, T2, T3, T4>(
    Func<T1, T2, T3, T4, TResult> valueFunction
)

Parameters

valueFunction
    Type: Func<(Of <(T1, T2, T3, T4, TResult)>))
    The function that will calculate the return value.
Type Parameters

T1
Type of the first argument of the invoked method.

T2
Type of the second argument of the invoked method.

T3
Type of the third argument of the invoked method.

T4
Type of the fourth argument of the invoked method.
Examples

Return a calculated value which is evaluated lazily at the time of the invocation.

The return value is calculated from the value of the actual method invocation arguments. Notice how the arguments are retrieved by simply declaring them as part of the lambda expression:

```csharp
mock.Expect(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<int>(),
    It.IsAny<bool>())
   .Returns((string arg1, string arg2, int arg3, bool arg4) => arg1
```

See Also

IReturns<(Of <(TResult>)>) Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq
IReturns<(<TResult>)>.::: Returns Method (TResult)
IReturns<(Of <(TResult)>)> Interface  Example  See Also  Send Feedback

Specifies the value to return.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IReturnsReturnsResult Returns(
    TResult value
)

Parameters

value
    Type: TResult
    The value to return, or nullNothingnullptr null reference (Nothing in Visual Basic).
Examples

Return a true value from the method call:

C#

```csharp
mock.Expect(x => x.Execute("ping"))
    .Returns(true);
```
See Also

I\text{Returns}\langle\text{Of }\langle\text{TResult}\rangle\rangle\text{ Interface}
\text{Returns Overload}
\text{Moq.Language Namespace}

Send comments on this topic to moqdisc@googlegroups.com
C# Moq

IReturnsGetter<(Of<(TProperty)>)> Interface

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)

Defines the Returns verb for property get expectations.
### Syntax

**C#**

```csharp
public interface IReturnsGetter<TProperty> : IReturns
```
Type Parameters

TProperty
  Type of the property.
See Also

IReturnsGetter<(Of <(TProperty)>)> Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members Include Inherited Members Moq IReturnsGetter(Of (TProperty)) Members IReturnsGetter(Of (TProperty)) Interface Methods See Also Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

IReturns Getter<Of <(TProperty)>>) Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
- Include Protected Members
- Include Inherited Members

Moq
IReturnsGetter(Of (TProperty)>) Methods

IReturnsGetter(Of (TProperty)>) Interface  See Also  Send Feedback
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

IReturnsGetter(Of <TProperty>>) Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members

Moq

IReturnsGetter<(Of <(TProperty)>)>...:::Returns Method

IReturnsGetter<(Of <(TProperty)>)> Interface  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Returns(Func&lt;Of TProperty&gt;)</code></td>
<td>Specifies a function that will calculate the value to return for the property.</td>
</tr>
<tr>
<td><code>Returns(TProperty)</code></td>
<td>Specifies the value to return.</td>
</tr>
</tbody>
</table>
See Also

IReturnsGetter<(Of <(TProperty)>)> Interface
IReturnsGetter<(Of <(TProperty)>)> Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

IReturnsGetter<Of <(TProperty)> >.:::Returns Method (<(Of <(TProperty) )>))

IReturnsGetter<Of <(TProperty) )> > Interface  Example  See Also  Send Feedback

Specifies a function that will calculate the value to return for the property.

Namespace:  Moq.Language
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
### Syntax

C#

```csharp
IReturnsResult Returns(
    Func<TProperty> valueFunction
)
```

### Parameters

valueFunction

- **Type:** Func<(Of (TProperty)>)
- The function that will calculate the return value.
Examples

Return a calculated value when the property is retrieved:

C#

```csharp
mock.ExpectGet(x => x.Suspended)
    .Returns(() => returnValues[0]);
```

The lambda expression to retrieve the return value is lazy-executed, meaning that its value may change depending on the moment the property is retrieved and the value the returnValues array has at that moment.
See Also

IReturnsGetter<(Of TProperty)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq
IReturnsGetter<Of (TProperty)>
:: Returns Method (TProperty)
IReturnsGetter<Of (TProperty)>) Interface

Example
See Also
Send Feedback

Specifies the value to return.

**Namespace:** Moq.Language
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IReturnsResult Returns(
    TProperty value
)

Parameters

value
Type: TProperty
The value to return, or nullNothingnullptr null reference (Nothing in Visual Basic).
Examples

Return a true value from the property getter call:

C#  

```csharp
mock.ExpectGet(x => x.Suspended)
    .Returns(true);
```
See Also

IReturnsGetter<Of <(TProperty)>>) Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#  
Moq  
IThrows Interface  
Members  See Also  Send Feedback  

Defines the Throws verb.  

**Namespace:**  Moq.Language  
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface IThrows
See Also

**IThrows Members**
**Moq.Language Namespace**

Send comments on this topic to moqdisc@googlegroups.com
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

IThrows Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members
Moq

IThrows Methods

See Also  Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thrown</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

IThrows Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members Include Inherited Members Moq

IThrows...:::Throws Method

IThrows Interface  See Also  Send Feedback
Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Throws&lt;Of</code></td>
<td>Specifies the type of exception to throw when the method is invoked.</td>
</tr>
<tr>
<td><code>&lt;(TException)&gt;)(QQ)</code></td>
<td></td>
</tr>
<tr>
<td><code>Throws(Exception)</code></td>
<td>Specifies the exception to throw when the method is invoked.</td>
</tr>
</tbody>
</table>
See Also

IThrows Interface
IThrows Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Moq

IThrows...::Throws(Of TException

Method

IThrows Interface  Example  See Also  Send Feedback

Specifies the type of exception to throw when the method is invoked.

**Namespace:** Moq.Language

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IThrowsResult Throws<TException>()
where TException : new(), Exception
Type Parameters

TException
Type of exception to instantiate and throw when the expectation is met.
Examples

This example shows how to throw an exception when the method is invoked with an empty string argument:

C#

```csharp
mock.Expect(x => x.Execute(""))
    .Throws<ArgumentException>();
```
See Also

IThrows Interface
Throws Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies the exception to throw when the method is invoked.

**Namespace:**  [Moq.Language](#)

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IThrowsResult Throws(
    Exception exception
)

Parameters

exception
    Type: Exception
    Exception instance to throw.
This example shows how to throw an exception when the method is invoked with an empty string argument:

```csharp
mock.Expect(x => x.Execute(""))
    .Throws(new ArgumentException());
```
See Also

IThrows Interface
Throws Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Defines the Verifiable verb.

**Namespace:**  [Moq.Language](#)

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface IVerifies
See Also

IVerifies Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `IVerifies` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifiable</td>
<td>Marks the expectation as verifiable, meaning that a call to VerifyQQ will check if this particular expectation was met.</td>
</tr>
</tbody>
</table>
See Also

IVerifies Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The IVerifies type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Verifiable</strong></td>
<td>Marks the expectation as verifiable, meaning that a call to Verify() will check if this particular expectation was met.</td>
</tr>
</tbody>
</table>
See Also

`IVerifies Interface`
`Moq.Language Namespace`

Send comments on this topic to `moqdisc@googlegroups.com`
C#

Moq

IVerifies...::Verifiable Method

IVerifies Interface  Example  See Also  Send Feedback

Marks the expectation as verifiable, meaning that a call to `Verify()` will check if this particular expectation was met.

**Namespace:**  Moq.Language

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

void Verifiable()
Examples

The following example marks the expectation as verifiable:

C#

```csharp
mock.Expect(x => x.Execute("ping"))
    .Returns(true)
    .Verifiable();
```
See Also

IVerifies Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Importing this namespace will enable the `Protected()` method on mocks, enabling expectations on protected members by specifying the member name as a string.
### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ItExpr</td>
<td>Allows the specification of a matching condition for an argument in a protected member expectation, rather than a specific argument value. &quot;ItExpr&quot; refers to the argument being matched.</td>
</tr>
<tr>
<td>ProtectedExtension</td>
<td>Enables the Protected() method on <code>Mock&lt;Of &lt;(T)&gt;</code>, allowing expectations to be set for protected members by using their name as a string, rather than strong-typing them which is not possible due to their visibility.</td>
</tr>
</tbody>
</table>
# Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IProtectedMock</td>
<td>Allows expectations to be set for protected members by using their name as a string, rather than strong-typing them which is not possible due to their visibility.</td>
</tr>
</tbody>
</table>

Send comments on this topic to [moqdisc@googlegroups.com](mailto:moqdisc@googlegroups.com)
IProtectedMock Interface

Allows expectations to be set for protected members by using their name as a string, rather than strong-typing them which is not possible due to their visibility.

**Namespace:** Moq.Protected
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public interface IProtectedMock
See Also

IProtectedMock Members
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `IProtectedMock` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ExpectGet&lt;(Of(TProperty))&gt;</td>
<td>Sets an expectation on a property getter with the given propertyName.</td>
</tr>
<tr>
<td>ExpectSet&lt;(Of(TProperty))&gt;</td>
<td>Sets an expectation on a property setter with the given propertyName.</td>
</tr>
</tbody>
</table>
See Also

IProtectedMock Interface
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `IProtectedMock` type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expect</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ExpectGet&lt;TProperty&gt;</td>
<td>Sets an expectation on a property getter with the given propertyName.</td>
</tr>
<tr>
<td>ExpectSet&lt;TProperty&gt;</td>
<td>Sets an expectation on a property setter with the given propertyName.</td>
</tr>
</tbody>
</table>
See Also

IProtectedMock Interface
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Include Protected Members
Include Inherited Members

Moq

IProtectedMock....Expect Method

IProtectedMock Interface  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Expect&lt;(Of TResult)&gt;(String, array&lt;Object&gt;[])[]</code></td>
<td>Sets an expectation on a property or a non void method with the given methodOrPropertyName, optionally specifying arguments for the method call.</td>
</tr>
<tr>
<td><code>Expect(String, array&lt;Object&gt;[])[]</code></td>
<td>Sets an expectation on the void method with the given voidMethodName, optionally specifying arguments for the method call.</td>
</tr>
</tbody>
</table>
See Also

IProtectedMock Interface
IProtectedMock Members
Mock.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
Sets an expectation on a property or a non-void method with the given methodOrPropertyName, optionally specifying arguments for the method call.

**Namespace:**  Moq.Protected  
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

IExpect<TResult> Expect<TResult>(
    string methodOrPropertyName,
    params Object[] args
)

Parameters

methodOrPropertyName
    Type: String
    Name of the method or property to be invoke.

args
    Type: array<Object>[]
    Optional arguments for the invocation.
Type Parameters

TResult
   Return type of the method or property.
See Also

IProtectedMock Interface
Expect Overload
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
IProtectedMock...:::Expect Method (, array<>[]0[])
IProtectedMock Interface  See Also  Send Feedback

Sets an expectation on the void method with the given voidMethodName, optionally specifying arguments for the method call.

**Namespace:**  [Moq.Protected](https://github.com/Moq/Moq)
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
## Syntax

### C#

```csharp
IExpect Expect(
    string voidMethodName,
    params Object[] args
)
```

## Parameters

**voidMethodName**
- **Type:** String
- **Description:** Name of the void method to be invoke.

**args**
- **Type:** array< Object >
- **Description:** Optional arguments for the invocation.
See Also

IProtectedMock Interface
Expect Overload
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
Sets an expectation on a property getter with the given propertyName.

**Namespace:** Moq.Protected

**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

C#

```csharp
IExpectGetter<TProperty> ExpectGet<TProperty>(
    string propertyName
)
```

**Parameters**

propertyName
- Type: String
- Name of the property.
Type Parameters

TProperty
  Type of the property.
See Also

IProtectedMock Interface
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Moq
IProtectedMock...:::ExpectSet<(Of <(TProperty)>)> Method

IProtectedMock Interface  See Also  Send Feedback

Sets an expectation on a property setter with the given propertyName.

Namespace:  Moq.Protected
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
### Syntax

**C#**

```csharp
IExpectSetter<TProperty> ExpectSet<TProperty>(
    string propertyName
)
```

### Parameters

- **propertyName**
  - Type: String
  - Name of the property.
Type Parameters

TProperty
   Type of the property.
See Also

IProtectedMock Interface
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
ItExpr Class

Allows the specification of a matching condition for an argument in a protected member expectation, rather than a specific argument value. "ItExpr" refers to the argument being matched.

Namespace: Moq.Protected
Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public static class ItExpr
Remarks

Use this variant of argument matching instead of It for protected expectations.

This class allows the expectation to match a method invocation with an arbitrary value, with a value in a specified range, or even one that matches a given predicate.
Inheritance Hierarchy

Object
    Moq.Protected...::ItExpr
See Also

ItExpr Members
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `ItExpr` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Is&lt;(Of &lt;(TValue)&gt;)&gt;</code></td>
<td>Matches any value that satisfies the given predicate.</td>
</tr>
<tr>
<td><code>IsAny&lt;(Of &lt;(TValue)&gt;)&gt;</code></td>
<td>Matches any value of the given TValue type.</td>
</tr>
<tr>
<td><code>IsInRange&lt;(Of &lt;(TValue)&gt;)&gt;</code></td>
<td>Matches any value that is in the range specified.</td>
</tr>
<tr>
<td><code>IsRegex</code></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

ItExpr Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `ItExpr` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Is&lt;(Of &lt;(TValue)&gt;)&gt;</code></td>
<td>Matches any value that satisfies the given predicate.</td>
</tr>
<tr>
<td><code>IsAny&lt;(Of &lt;(TValue)&gt;)&gt;</code></td>
<td>Matches any value of the given TValue type.</td>
</tr>
<tr>
<td><code>IsInRange&lt;(Of &lt;(TValue)&gt;)&gt;</code></td>
<td>Matches any value that is in the range specified.</td>
</tr>
<tr>
<td><code>IsRegex</code></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

ItExpr Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
Matches any value that satisfies the given predicate.

**Namespace:**  [Moq.Protected](https://www.nuget.org/packages/Moq.Protected)

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public static Expression Is<TValue>(
    Expression<Predicate<TValue>> match
)
```

Parameters

match
Type: Expression<(Of <(Predicate<(Of <(TValue)>)>)>))>
The predicate used to match the method argument.
Type Parameters

TValue
Type of the argument to check.
Remarks

Allows the specification of a predicate to perform matching of method call arguments.
Examples

This example shows how to return the value 1 whenever the argument to the Do method is an even number.

```csharp
mock.Protected()
    .Expect("Do", ItExpr.Is<int>(i => i % 2 == 0))
    .Returns(1);
```

This example shows how to throw an exception if the argument to the method is a negative number:

```csharp
mock.Protected()
    .Expect("GetUser", ItExpr.Is<int>(i => i < 0))
    .Throws(new ArgumentException());
```
See Also

ItExpr Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
ItExpr::IsAny<(Of <(TValue)>)> Method

Matches any value of the given TValue type.

Namespace:  Moq.Protected
Assembly:   Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public static Expression IsAny<TValue>()
Type Parameters

TValue
    Type of the value.
Remarks

Typically used when the actual argument value for a method call is not relevant.
Examples

C#

```csharp
// Throws an exception for a call to Remove with any string value.
mock.Protected()
    .Expect("Remove", ItExpr.IsAny<string>())
    .Throws(new InvalidOperationException());
```
See Also

ItExpr Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

ItExpr..::.InRange(Of tvb> Method

ItExpr Class  Example  See Also  Send Feedback

Matches any value that is in the range specified.

**Namespace:**  Moq.Protected

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
## Syntax

### C#

```csharp
public static Expression IsInRange<TValue>(
    TValue from,
    TValue to,
    Range rangeKind
)
where TValue : IComparable
```

### Parameters

**from**
- Type: TValue
  - The lower bound of the range.

**to**
- Type: TValue
  - The upper bound of the range.

**rangeKind**
- Type: **Moq::Range**
  - The kind of range. See **Range**.
Type Parameters

TValue
Type of the argument to check.
Examples

The following example shows how to expect a method call with an integer argument within the 0..100 range.

C#

```csharp
mock.Protected()
    .Expect("HasInventory",
           ItExpr.IsAny<string>(),
           ItExpr.IsInRange(0, 100, Range.Inclusive))
    .Returns(false);
```
See Also

ItExpr Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
ItExpr...::IsRegex Method

ItExpr Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsRegex(String)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
<tr>
<td>IsRegex(String, RegexOptions)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
</tbody>
</table>
See Also

ItExpr Class
ItExpr Members
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
Matches a string argument if it matches the given regular expression pattern.

**Namespace:**  Moq.Protected
**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public static Expression IsRegex(
    string regex
)
```

Parameters

regex
  Type: String
  The pattern to use to match the string argument value.
Examples

The following example shows how to expect a call to a method where the string argument matches the given regular expression:

```csharp
mock.Protected()
    .Expect("Check", ItExpr.IsRegex("[a-z]+"))
    .Returns(1);
```
See Also

ItExpr Class
IsRegex Overload
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
Matches a string argument if it matches the given regular expression pattern.

**Namespace:** Moq.Protected  
**Assembly:** Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public static Expression IsRegex(
    string regex,
    RegexOptions options
)
```

Parameters

regex
  Type: String
  The pattern to use to match the string argument value.

options
  Type: RegexOptions
  The options used to interpret the pattern.
Examples

The following example shows how to expect a call to a method where the string argument matches the given regular expression, in a case insensitive way:

```c#
mock.Protected()
    .Returns(1);
```
See Also

ItExpr Class
IsRegex Overload
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
Enables the Protected() method on `Mock<Of<(<T)>>>`, allowing expectations to be set for protected members by using their name as a string, rather than strongly typing them which is not possible due to their visibility.

**Namespace:**  Moq.Protected

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public static class ProtectedExtension
Inheritance Hierarchy

Object
    Moq.Protected...::ProtectedExtension
See Also

ProtectedExtension Members
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
The *ProtectedExtension* type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Protected&lt;Of&lt;((T))&gt;&gt;</code></td>
<td>Enable protected expectations for the mock.</td>
</tr>
</tbody>
</table>
See Also

ProtectedExtension Class
Moq.Protection Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `ProtectedExtension` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected&lt;(&lt;T&gt;)&gt;</td>
<td>Enable protected expectations for the mock.</td>
</tr>
</tbody>
</table>
See Also

ProtectedExtension Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
Enable protected expectations for the mock.

**Namespace:**  
Moq.Protected  

**Assembly:**  
Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
**Syntax**

**C#**

```csharp
public static IProtectedMock Protected<T>(
    Mock<T> mock
)
where T : class
```

**Parameters**

- `mock`<br>  Type: `Moq..::Mock<Of <(T)>>`<br>  The mock to set the protected expectations on.
Type Parameters

T

Mocked object type. Typically omitted as it can be inferred from the mock instance.
See Also

- ProtectedExtension Class
- Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
[Missing <summary> documentation for N:Moq.Stub]
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StubExtensions</td>
<td>Adds Stub extension method to a mock so that you can stub properties.</td>
</tr>
</tbody>
</table>

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

StubExtensions Class

Members  See Also  Send Feedback

Adds Stub extension method to a mock so that you can stub properties.

**Namespace:**  Moq.Stub

**Assembly:**  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

public static class StubExtensions
Inheritance Hierarchy

Object
Moq.Stub..:::StubExtensions
See Also

StubExtensions Members
Moq.Stub Namespace

Send comments on this topic to moqdisc@googlegroups.com
The **StubExtensions** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stub</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>StubAll&lt;Of&lt;(T)&gt;</td>
<td>Stubs all properties on the mock, setting the default value to the one generated as specified by the <code>DefaultValue</code> property.</td>
</tr>
</tbody>
</table>
See Also

StubExtensions Class
Moq.Stub Namespace

Send comments on this topic to moqdisc@googlegroups.com
The **StubExtensions** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stub</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>StubAll(Of&lt;T&gt;)</strong></td>
<td>Stubs all properties on the mock, setting the default value to the one generated as specified by the <code>DefaultValue</code> property.</td>
</tr>
</tbody>
</table>
See Also

StubExtensions Class
Moq.Stub Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
- Include Protected Members
- Include Inherited Members
Moq

StubExtensions...:::Stub Method

StubExtensions Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stub&lt;(Of &lt;(T, TProperty)&gt;)(Mock&lt;(Of &lt;(T)&gt;), Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;))&gt;</td>
<td>Specifies that the given property should have stub behavior, meaning that setting its value will cause it to be saved and later returned when the property is requested.</td>
</tr>
<tr>
<td>Stub&lt;(Of &lt;(T, TProperty)&gt;)(Mock&lt;(Of &lt;(T)&gt;), Expression&lt;(Of &lt;(Func&lt;(Of &lt;(T, TProperty)&gt;)&gt;)&gt;), TProperty)&gt;</td>
<td>Specifies that the given property should have stub behavior, meaning that setting its value will cause it to be saved and later returned when the property is requested. This overload allows setting the initial value for the property.</td>
</tr>
</tbody>
</table>
See Also

StubExtensions Class
StubExtensions Members
Moq.Stub Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies that the given property should have stub behavior, meaning that setting its value will cause it to be saved and later returned when the property is requested.

**Namespace:** `Moq.Stub
` **Assembly:** `Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)`
C#

```csharp
public static void Stub<T, TProperty>(
    Mock<T> mock,
    Expression<Func<T, TProperty>> property
)
where T : class
```

**Parameters**

- **mock**
  - Type: `Mock<>()` (Of `<T>`)  
  - The instance to stub.

- **property**  
  - Type: `Expression<>()` (Of `<(Func<>()<T, TProperty>)>`)  
  - Property expression to stub.
**Type Parameters**

T  
Mocked type, inferred from the object where this method is being applied (does not need to be specified).

TProperty  
Type of the property, inferred from the property expression (does not need to be specified).
**Examples**

If you have an interface with an int property Value, you might stub it using the following straightforward call:

```
C#

var mock = new Mock<IHaveValue>();
mock.Stub(v => v.Value);
```

After the Stub call has been issued, setting and retrieving the object value will behave as expected:

```
C#

IHaveValue v = mock.Object;

v.Value = 5;
Assert.Equal(5, v.Value);
```
See Also

StubExtensions Class
Stub Overload
Moq.Stub Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

StubExtensions...:::Stub<(T, TProperty)> Method (Mock<(T)>,
<(Of <(Of (T, TProperty)>)>), TProperty)

Specifies that the given property should have stub behavior, meaning that setting
its value will cause it to be saved and later returned when the property is
requested. This overload allows setting the initial value for the property.

Namespace:  Moq.Stub
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public static void Stub<T, TProperty>(
    Mock<T> mock,
    Expression<Func<T, TProperty>> property,
    TProperty initialValue
)
where T : class
```

Parameters

mock
Type: `Moq::Mock<Of (T)>`
The instance to stub.

property
Type: `Expression<Of (Func<Of (T, TProperty)>)>`
Property expression to stub.

initialValue
Type: `TProperty`
Initial value for the property.
Type Parameters

T
Mocked type, inferred from the object where this method is being applied (does not need to be specified).

TProperty
Type of the property, inferred from the property expression (does not need to be specified).
Examples

If you have an interface with an int property Value, you might stub it using the following straightforward call:

```csharp
var mock = new Mock<IHaveValue>();
mock.Stub(v => v.Value, 5);
```

After the Stub call has been issued, setting and retrieving the object value will behave as expected:

```csharp
IHaveValue v = mock.Object;
// Initial value was stored
Assert.Equal(5, v.Value);

// New value set which changes the initial value
v.Value = 6;
Assert.Equal(6, v.Value);
```
See Also

StubExtensions Class
Stub Overload
Moq.Stub Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#

Moq

StubExtensions..:::.StubAll<(Of <(T)>)> Method

StubExtensions Class  See Also  Send Feedback

Stubs all properties on the mock, setting the default value to the one generated as specified by the DefaultValue property.

Namespace:  Moq.Stub
Assembly:  Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)
Syntax

C#

```csharp
public static void StubAll<T>(
    Mock<T> mock
)
where T : class
```

Parameters

`mock`

Type: `Moq::Mock<(Of <(T)>)>`
The mock to stub.
Type Parameters

T

Mocked type, typically omitted as it can be inferred from the mock argument.
Remarks

If the mock `DefaultValue` is set to `Mock`, the mocked default values will also be stubbed recursively.
See Also

StubExtensions Class
Moq.Stub Namespace

Send comments on this topic to moqdisc@googlegroups.com