■ □ C# Moq Moq Namespace <u>Send Feedback</u>

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 <u>online quickstarts</u> 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 <u>blog entries</u> from around the world about Moq.

Classes

	Class	Description
~ \$	<u>It</u>	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.
* \$	<u>MatcherAttribute</u>	Marks a method as a matcher, which allows complete replacement of the built-in <u>It</u> class with your own argument matching rules.
~ \$	<u>Mock</u>	Base class for mocks and static helper class with methods that apply to mocked objects, such as $Get \le (Of \le (T \ge) \ge)(T)$ to retrieve a $Mock \le (Of \le (T \ge) \ge)$ from an object instance.
\$ \$	<u>Mock≤(Of</u> ≤ <u>(T≥)≥)</u>	Core implementation of the $\underline{IMock \leq (Of \leq (T \geq) >)}$ interface.
\$ \$	<u>MockedEvent</u>	Represents a generic event that has been mocked and can be rised.
\$ \$	<pre>MockedEvent<(Of <(TEventArgs>)>)</pre>	Provides a typed <u>MockedEvent</u> for a specific type of EventArgs.
\$ \$	<u>MockException</u>	Exception thrown by mocks when expectations are not met, the mock is not properly setup, etc.
A \$	<u>MockFactory</u>	Utility factory class to use to construct multiple mocks when consistent verification is desired for all of them.

Interfaces

Interface	Description
> <u>IMock</u>	Base mock interface exposing non-generic members.
$ \frac{\text{IMock} \leq (Of}{\leq (T \geq) \geq} $	Provides a mock implementation of T.

Enumerations

Enumeration	Description
	Determines the way default values are generated calculated for loose mocks.
MockBehavior	Options to customize the behavior of the mock.
Ranoe	Kind of range to use in a filter specified through <u>IsInRange<(Of <(TValue>)>)(TValue, TValue, Range)</u> .

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq DefaultValue Enumeration <u>See Also</u> <u>Send Feedback</u>

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

Member name	Description
Empty	Default behavior, which generates empty values for value types (i.e. default(int)), empty array and enumerables, and nulls for all other reference types.
	Whenever the default value generated by Empty is null, replaces this value with a mock (if the type can be mocked).



Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IMock Interface <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

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



IMock Members Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IMock Members
 IMock Interface Methods Properties See Also Send Feedback

The <u>IMock</u> type exposes the following members.

Methods

	Name	Description
-	<u>CreateEventHandler</u>	Overloaded.
-	• <u>Verify</u>	Verifies that all verifiable expectations have been met.
-	🖬 Αργιτν Α Π	Verifies all expectations regardless of whether they have been flagged as verifiable.

Properties

	Name	Description
₽ <u>C</u>	<u>allBase</u>	Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to trueTruetruetrue (True in Visual Basic).
	<u>efaultValue</u>	Determines how to generate default values for loose mocks on unexpected invocations.
<u>0</u>	<u>bject</u>	The mocked object instance.



IMock Interface Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IMock Methods
 <u>IMock Interface See Also Send Feedback</u>

The <u>IMock</u> type exposes the following members.

Methods

	Name	Description
-	<u>CreateEventHandler</u>	Overloaded.
-	• <u>Verify</u>	Verifies that all verifiable expectations have been met.
-	🖬 Αργιτν Α Π	Verifies all expectations regardless of whether they have been flagged as verifiable.



IMock Interface Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IMock..::.CreateEventHandler Method
 IMock Interface See Also Send Feedback

Overload List

Name

- <u>CreateEventHandler<(Of</u> = 🏟 <(TEventArgs>)>)()()()
- =Ŵ

Description

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

Creates a handler that can be associated to an <u>CreateEventHandler()()()</u> event receiving a generic EventArgs and can be used to raise the event.



IMock Interface IMock Members Moq Namespace

Send comments on this topic to 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# Copy 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 <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IMock..::.CreateEventHandler Method <u>IMock Interface Example See Also Send Feedback</u>

Creates a handler that can be associated to an event receiving a generic EventArgs 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#

MockedEventCreateEventHandler()

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:

Copy

C#
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 <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IMock..::.Verify Method <u>IMock Interface Example See Also Send Feedback</u> 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# Copy
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

Exception Moq..::.MockException **Condition** Not all verifiable expectations were met.

See Also

IMock Interface Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IMock..::.VerifyAll Method <u>IMock Interface Example See Also Send Feedback</u>

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)

Syntax

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# Copy
var mock = new Mock<IWarehouse>();
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

Exception Moq..::.MockException Condition

At least one expectation was not met.

See Also

IMock Interface Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IMock Properties
 <u>IMock Interface See Also Send Feedback</u>

The <u>IMock</u> type exposes the following members.

Properties

	Name	Description
₽ <u>C</u>	<u>allBase</u>	Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to trueTruetruetrue (True in Visual Basic).
	<u>efaultValue</u>	Determines how to generate default values for loose mocks on unexpected invocations.
<u>0</u>	<u>bject</u>	The mocked object instance.



IMock Interface Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IMock..::.CallBase Property <u>IMock Interface</u> <u>See Also</u> <u>Send Feedback</u>

Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to trueTruetruetrue (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

■ □ C# Moq IMock..::.DefaultValue Property <u>IMock Interface</u> <u>See Also</u> <u>Send Feedback</u>

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 { get; set; }

See Also

IMock Interface Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IMock..::.Object Property <u>IMock Interface See Also Send Feedback</u> 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

□ C#
Moq
IMock<(Of <(T>)>) Interface
Members Example See Also Send Feedback
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

Т

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 <u>MockBehavior</u> that can be passed to the <u>Mock<(Of <(T>)>)(MockBehavior</u>) constructor.

Examples

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

C#

Copy

```
//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 <u>It</u> class to specify conditions for
arguments instead of specific values:
C#
                                                                  Copy
//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

<u>IMock<(Of <(T≥)≥) Members</u> <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IMock<(Of <(T>)>) Members
 IMock<(Of <(T>)>) Interface Methods Properties See Also Send Feedback

The $\underline{IMock} \leq (Of \leq (T \geq))$ type exposes the following members.

Methods

	Name	Description
=	<u>As<(Of</u>	Adds an interface implementation to the mock,
	<u><(TInterface≥)≥)</u>	allowing expectations to be set for it.
-=•	<u>Expect</u>	Overloaded.
=	<u>ExpectGet<(Of</u>	Sets an expectation on the mocked type for a call to
	< <u>(TProperty>)>)</u>	to a property getter.
=	ExpectSet	Overloaded.
=	<u>Verify</u>	Overloaded.
-=•	<u>VerifyAll</u>	Implements <u>VerifyAll()()()</u> .
=0	<u>VerifyGet<(Of</u>	Verifies that a property was read on the mock. Use
	< <u>(TProperty≥)></u>)	in conjuntion with the default Loose.
-=•	<u>VerifySet</u>	Overloaded.
I		

Properties

	Name	Description
*	Benavior	Behavior of the mock, according to the value set in the constructor.
2	<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations.
2	<u>Object</u>	Exposes the mocked object instance.

See Also

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IMock<(Of <(T>)>) Methods
 IMock<(Of <(T>)>) Interface See Also Send Feedback

The $\underline{IMock} \leq (Of \leq (T \geq))$ type exposes the following members.

Methods

	Name	Description
=	<u>As<(Of</u>	Adds an interface implementation to the mock,
	<u><(TInterface≥)≥)</u>	allowing expectations to be set for it.
-=•	<u>Expect</u>	Overloaded.
=	<u>ExpectGet<(Of</u>	Sets an expectation on the mocked type for a call to
	< <u>(TProperty>)>)</u>	to a property getter.
=	ExpectSet	Overloaded.
=	<u>Verify</u>	Overloaded.
-=•	<u>VerifyAll</u>	Implements <u>VerifyAll()()()</u> .
=0	<u>VerifyGet<(Of</u>	Verifies that a property was read on the mock. Use
	< <u>(TProperty≥)></u>)	in conjuntion with the default Loose.
-=•	<u>VerifySet</u>	Overloaded.
I		

See Also

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IMock<(Of <(T>)>)..::.As<(Of <(TInterface>)>) Method IMock<(Of <(T>)>) Interface Example See Also Send Feedback

Adds an interface implementation to the mock, allowing expectations to be set for it.

Namespace: Moq 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 <u>Object</u> 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:

C#

Copy

```
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

Exception	Condition
InvalidOperationExceptior	The mock type has already been generated by
invalueoperationException	accessing the <u>Object</u> property.
ArgumentException	The TInterface specified is not an interface.

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

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

Overload List

Name

- Expect(Expression≤(Of ≤(Action≤(Of ≤(T≥)≥)≥)))
 Expect≤(Of ≤(TResult≥)≥)
- (Expression < (Of < (Func < (Of < (T, TResult >) >) >)))

Description

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

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 FeedbackSets 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

expression

Type: Expression<(Of <(Action<(Of $<(\underline{T}>)>)>)>$) 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#

Сору

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

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C# Moq IMock<(Of <(T>)>)..::.Expect<(Of <(TResult>)>) Method (<(Of <(<(Of <(T, TResult>)>)>))) IMock<(Of <(T>)>) Interface Example See Also Send Feedback

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

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

Syntax

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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#

Сору

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

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C# Moq IMock<(Of <(T>)>)..::.ExpectGet<(Of <(TProperty>)>) Method IMock<(Of <(T>)>) Interface Example See Also Send FeedbackSets an expectation on the mocked type for a call to to a property getter.

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

Syntax

C#

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#

Сору

mock.ExpectGet(x => x.Suspended)
 .Returns(true);

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

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

Overload List

Name

ExpectSet<(Of <(TProperty>)>)

- (Expression<(Of <(Func<(Of <(T, TProperty>)>)>)))
 - <u>ExpectSet<(Of <(TProperty>)>)</u>
- (Expression<(Of <(Func<(Of <(T,</pre>
 - <u>TProperty>)>)>)), TProperty)</u>

Description

Sets an expectation on the mocked type for a call to to a property setter.

Sets an expectation on the mocked type for a call to to a property setter with a specific value.

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

Send comments on this topic to moqdisc@googlegroups.com

-

□ C#

Moq

IMock<(Of <(T>)>)..::.ExpectSet<(Of <(TProperty>)>) Method (<(Of <(<(Of <((T, TProperty>)>)>)))

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

Sets an expectation on the mocked type for a call to to a property setter.

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

Syntax

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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#

Сору

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

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

IMock<(Of <(T>)>)..::ExpectSet<(Of <(TProperty>)>) Method (<(Of <(<(Of <(T, TProperty>)>)>)>), TProperty) IMock<(Of <(T>)>) Interface Example See Also Send Feedback

Sets an expectation on the mocked type for a call to 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#

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

Parameters

expression

Type: Expression<(Of <(Func<(Of <(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#

Сору

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

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

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

Overload List

Name

•• <u>Verify()()()</u>

=ŵ

 $\underbrace{ Verify(Expression < (Of < (Action < (Of < (T >) >) >)))}_{\leq (Action < (Of < (T >) >) >))}$

<u>Verify<(Of <(TResult>)>)</u> (Expression<(Of <(Func<(Of

<u> <(T, TResult≥)≥)≥)</u>)

Description

Implements Verify()()(). Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjuntion with the default Loose. Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjuntion with the default Loose.

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

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IMock<(Of <(T>)>)..::.Verify Method <u>IMock<(Of <(T>)>) Interface See Also Send Feedback</u> Implements <u>Verify()()()</u>.

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

Syntax

C#

void Verify()

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IMock<(Of <(T>)>)..::.Verify Method (<(Of <(<(Of <(T>)>)>)>))) 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 conjuntion with the default <u>Loose</u>.

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

Syntax

C#

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

Parameters

expression

Type: Expression<(Of <(Action<(Of <(<u>T</u>>)>)>)>) 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# Copy

```
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

Exception Moq..::.MockException

Condition

The invocation was not performed on the mock.

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IMock<(Of <(T>)>)..::.Verify<(Of <(TResult>)>) Method (<(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 conjuntion with the default <u>Loose</u>.

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

Syntax

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, TResult>)>)>)>) Expression to verify.

Type Parameters

TResult

Type of return value from the expression.

Examples

C#

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

Сору

```
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

Exception Moq..::.MockException

Condition

The invocation was not performed on the mock.

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IMock<(Of <(T>)>)..::.VerifyAll Method <u>IMock<(Of <(T>)>) Interface See Also Send Feedback</u> Implements <u>VerifyAll()()()</u>.

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

Syntax

C#

void VerifyAll()

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IMock<(Of <(T>)>)..::.VerifyGet<(Of <(TProperty>)>) Method IMock<(Of <(T>)>) Interface Example See Also Send Feedback

Verifies that a property was read on the mock. Use in conjuntion with the default Loose.

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

Syntax

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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:

C# Copy

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

Exception Moq..::.MockException

Condition

The invocation was not performed on the mock.

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

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

Overload List

Name

VerifySet<(Of <(TProperty>)>)

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

Description

Verifies that a property has been set on the mock. Use in conjuntion with the default <u>Loose</u>.

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

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

Send comments on this topic to moqdisc@googlegroups.com

□ C# Moq IMock<(Of <(T>)>)..::.VerifySet<(Of <(TProperty>)>) Method (<(Of <(<(Of <((T, TProperty>)>)>))))

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

Verifies that a property has been set on the mock. Use in conjuntion with the default <u>Loose</u>.

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

Syntax

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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:

C# Copy

```
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

Exception Moq..::.MockException

Condition

The invocation was not performed on the mock.

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

IMock<(Of <(T>)>).::.VerifySet<(Of <(TProperty>)>) Method (<(Of <(<(Of <(T, TProperty>)>)>)>), 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 conjuntion with the default <u>Loose</u>.

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

Syntax

C#

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

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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:

C# Copy

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

Exceptions

Exception Moq..::.MockException

Condition

The invocation was not performed on the mock.

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IMock<(Of <(T>)>) Properties
 IMock<(Of <(T>)>) Interface See Also Send Feedback

The $\underline{IMock} \leq (Of \leq (T \geq))$ type exposes the following members.

Properties

	Name	Description
*	Benavior	Behavior of the mock, according to the value set in the constructor.
2	<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations.
2	<u>Object</u>	Exposes the mocked object instance.

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IMock<(Of <(T>)>)..::.Behavior Property <u>IMock<(Of <(T>)>) Interface</u> <u>See Also</u> <u>Send Feedback</u>

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; }

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IMock<(Of <(T>)>)..::.DefaultValue Property <u>IMock<(Of <(T>)>) Interface</u> <u>See Also</u> <u>Send Feedback</u>

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

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

Syntax

C#

DefaultValue { get; set; }

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IMock<(Of <(T>)>)..::.Object Property <u>IMock<(Of <(T>)>) Interface See Also Send Feedback</u> Exposes the mocked object instance.

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

Syntax

C#

T Object { get; }

<u>IMock<(Of <(T≥)≥)</u> Interface <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq It Class <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

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

<u>It Members</u> <u>Moq Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 It Members
 It Class Methods See Also Send Feedback

The <u>It</u> type exposes the following members.

Methods

	Name	Description
≓≬S	<u> Is<(Of <(TValue>)>)</u>	Matches any value that satisfies the given predicate.
≡≬S	<pre>IsAny<(Of <(TValue>)>)</pre>	Matches any value of the given TValue type.
≓≬S	<u>IsInRange<(Of</u> < <u>(TValue>)></u>)	Matches any value that is in the range specified.
= \$	<u>IsRegex</u>	Overloaded.

<u>It Class</u> <u>Moq Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 It Methods
 <u>It Class See Also Send Feedback</u>

The <u>It</u> type exposes the following members.

Methods

	Name	Description
≓≬S	<u> Is<(Of <(TValue>)>)</u>	Matches any value that satisfies the given predicate.
≡≬S	<pre>IsAny<(Of <(TValue>)>)</pre>	Matches any value of the given TValue type.
≓≬S	<u>IsInRange<(Of</u> < <u>(TValue>)></u>)	Matches any value that is in the range specified.
= \$	<u>IsRegex</u>	Overloaded.

<u>It Class</u> <u>Moq Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 It..::.Is<(Of <(TValue>)>) Method
 It Class Example See Also Send Feedback
 Matches any value that satisfies the given predicate.

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

Syntax

C#

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.

C#	Сору
----	------

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:

C#

Copy

mock.Expect(x => x.GetUser(It.Is<int>(i => i < 0)))
.Throws(new ArgumentException());</pre>

<u>It Class</u> <u>Moq Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 It..::.IsAny<(Of <(TValue>)>) Method
 It Class Example See Also Send Feedback
 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#

Сору

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

<u>It Class</u> <u>Moq Namespace</u>

□ C#
 Moq
 It..::.IsInRange<(Of <(TValue>)>) Method
 It Class Example See Also Send Feedback
 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#

public static TValue IsInRange<TValue>(
 TValue from,
 TValue to,
 <u>Range</u> 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: <u>Moq..::Range</u> The kind of range. See <u>Range</u>.

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# Copy mock.Expect(x => x.HasInventory(

```
It.IsAny<<u>string</u>>(),
It.IsInRange(0, 100, Range.Inclusive)))
.Returns(false);
```

<u>It Class</u> <u>Moq Namespace</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 It.::IsRegex Method
 It Class See Also Send Feedback

Overload List

Name

Description

■ IsRegex(String) IsRegex(String, Regex(Dptions) Matches a string argument if it matches the given regular expression pattern.

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

<u>It Class</u> <u>It Members</u> <u>Moq Namespace</u>

■ □ C# Moq It..::.IsRegex Method () <u>It Class Example See Also Send Feedback</u>

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#

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#

Сору

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

<u>It Class</u> <u>IsRegex Overload</u> <u>Moq Namespace</u>

■ □ C# Moq It..::.IsRegex Method (,) <u>It Class Example See Also Send Feedback</u>

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# Copy

mock.Expect(x => x.Check(It.IsRegex("[a-z]+", RegexOptions.IgnoreCas

<u>It Class</u> <u>IsRegex Overload</u> <u>Moq Namespace</u>

■ □ C# Moq MatcherAttribute Class <u>Members Example See Also Send Feedback</u>

Marks a method as a matcher, which allows complete replacement of the built-in <u>It</u> class with your own argument matching rules.

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

Syntax

C#

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:

```
C#
```

C#

```
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:

Copy

Copy

```
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:

```
C# Copy

public static bool Contains(IEnumerable<Order> orders, Order orc

{

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 <u>It</u> 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

C#

The following is the complete example explained above:

Copy

```
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:

```
C# Copy
```

```
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

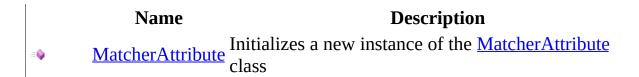
Object Attribute Moq..::.MatcherAttribute

MatcherAttribute Members Moq Namespace

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MatcherAttribute Members
 MatcherAttribute Class Constructors See Also Send Feedback

The <u>MatcherAttribute</u> type exposes the following members.

Constructors



MatcherAttribute Class Moq Namespace

C#
 Moq
 MatcherAttribute Constructor
 MatcherAttribute Class See Also Send Feedback
 Initializes a new instance of the MatcherAttribute class

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

Syntax

C#

public MatcherAttribute()

MatcherAttribute Class Moq Namespace

■ □ C# Moq Mock Class <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

Base class for mocks and static helper class with methods that apply to mocked objects, such as $Get \le (Of \le (T \ge) \ge)(T)$ to retrieve a $Mock \le (Of \le (T \ge) \ge)$ 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 <u>Moq..::.Mock<(Of <(T>)>)</u>

<u>Mock Members</u> <u>Moq Namespace</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock Members
 Mock Class Constructors Methods Properties See Also Send Feedback

The <u>Mock</u> type exposes the following members.

Constructors

Name Description

• <u>Mock</u> Initializes the mock

Methods

	Name	Description
- =	<u>CreateEventHandle</u>	<u>r</u> Overloaded.
≓≬S	<u>Get<(Of <(T>)>)</u>	Retrieves the mock object for the given object instance.
<u>ş</u>	<u>GetObject</u>	Returns the mocked object value.
-=•	<u>Verify</u>	Implements <u>Verify()()(</u>).
≡♦	<u>VerifyAll</u>	Implements <u>VerifyAll()()()</u> .

Properties

	Name	Description
*	<u>Behavior</u>	Implements <u>Behavior</u> .
P	<u>CallBase</u>	Implements <u>CallBase</u> .
P	<u>DefaultValue</u>	Implements <u>DefaultValue</u> .
5	ImplementedInterfaces	Exposes the list of extra interfaces implemented by the mock.
*	<u>Object</u>	The mocked object instance. Implements Object.

<u>Mock Class</u> <u>Moq Namespace</u>

□ C#
 Moq
 Mock Constructor
 <u>Mock Class See Also Send Feedback</u>
 Initializes the mock

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

Syntax

C#

protected Mock()

<u>Mock Class</u> <u>Moq Namespace</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock Methods
 Mock Class See Also Send Feedback

The <u>Mock</u> type exposes the following members.

Methods

	Name	Description
- =	<u>CreateEventHandle</u>	<u>r</u> Overloaded.
≓≬S	<u>Get<(Of <(T>)>)</u>	Retrieves the mock object for the given object instance.
<u>ş</u>	<u>GetObject</u>	Returns the mocked object value.
-=•	<u>Verify</u>	Implements <u>Verify()()(</u>).
≡♦	<u>VerifyAll</u>	Implements <u>VerifyAll()()()</u> .

<u>Mock Class</u> <u>Moq Namespace</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock..::CreateEventHandler Method
 Mock Class See Also Send Feedback

Overload List

Name

- <u>CreateEventHandler<(Of</u>
 <u>CTEventArge>)>)()()</u>
- <(TEventArgs>)>)()()()
- CreateEventHandler()()()

Description

Implements <u>CreateEventHandler<(Of</u> <<u>(TEventArgs>)>)()()()</u>. Implements <u>CreateEventHandler()()()</u>



Mock Class Mock Members Moq Namespace

■ □ C# Moq Mock..::.CreateEventHandler<(Of <(TEventArgs>)>) Method <u>Mock Class See Also Send Feedback</u> Implements <u>CreateEventHandler<(Of <(TEventArgs>)>)()()()</u>.

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



C#

public MockedEvent<TEventArgs> CreateEventHandler<TEventArgs>()
where TEventArgs : EventArgs

Type Parameters

TEventArgs

Type of event argument class.

<u>Mock Class</u> <u>CreateEventHandler Overload</u> <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq Mock..::.CreateEventHandler Method <u>Mock Class See Also Send Feedback</u> Implements <u>CreateEventHandler()()()</u>

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

Syntax

C#

public MockedEvent<EventArgs> CreateEventHandler()

<u>Mock Class</u> <u>CreateEventHandler Overload</u> <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C#
 Moq
 Mock..::.Get<(Of <(T>)>) Method
 Mock Class Example See Also Send Feedback
 Retrieves the mock object for the given object instance.

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

Syntax

C#

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

Parameters

mocked Type: T The instance of the mocked object.

Type Parameters

Т

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.

Examples

The following example shows how to add a new expectation to an object instance which is not the original $Mock \le (Of \le (T \ge) \ge)$ but rather the object associated with it:

C# Copy
// 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

Exception

Condition

ArgumentException The received mocked instance was not created by Moq.

<u>Mock Class</u> <u>Moq Namespace</u>

□ C#
 Moq
 Mock..::.GetObject Method
 Mock Class See Also Send Feedback
 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()

<u>Mock Class</u> <u>Moq Namespace</u>

■ □ C# Moq Mock..::.Verify Method <u>Mock Class See Also Send Feedback</u> Implements <u>Verify()()()</u>.

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()()()

<u>Mock Class</u> <u>Moq Namespace</u>

■ □ C# Moq Mock..::.VerifyAll Method <u>Mock Class See Also Send Feedback</u> Implements <u>VerifyAll()()()</u>.

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()()()

<u>Mock Class</u> <u>Moq Namespace</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock Properties
 <u>Mock Class See Also Send Feedback</u>

The <u>Mock</u> type exposes the following members.

Properties

	Name	Description
*	<u>Behavior</u>	Implements <u>Behavior</u> .
P	<u>CallBase</u>	Implements <u>CallBase</u> .
P	<u>DefaultValue</u>	Implements <u>DefaultValue</u> .
5	ImplementedInterfaces	Exposes the list of extra interfaces implemented by the mock.
*	<u>Object</u>	The mocked object instance. Implements Object.

Mock Class Moq Namespace

□ C#
 Moq
 Mock..::.Behavior Property
 Mock Class See Also Send Feedback
 Implements Behavior.

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

Syntax

C#

public MockBehavior Behavior { get; internal set; }

<u>Mock Class</u> <u>Moq Namespace</u>

□ C#
 Moq
 Mock..::.CallBase Property
 Mock Class See Also Send Feedback
 Implements CallBase.

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

Syntax

C#

public bool CallBase { get; set; }

Implements

IMock..::.CallBase

<u>Mock Class</u> <u>Moq Namespace</u>

□ C#
 Moq
 Mock..::.DefaultValue Property
 Mock Class See Also Send Feedback
 Implements DefaultValue.

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

Syntax

C#

public DefaultValue { get; set; }

Implements

IMock..::.DefaultValue

<u>Mock Class</u> <u>Moq Namespace</u>

■ □ C# Moq Mock..::.ImplementedInterfaces Property <u>Mock Class See Also Send Feedback</u>

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 :

<u>Mock Class</u> <u>Moq Namespace</u>

□ C#
 Moq
 Mock..::.Object Property
 Mock Class See Also Send Feedback
 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

<u>Mock Class</u> <u>Moq Namespace</u>

■ □ C# Moq Mock<(Of <(T>)>) Class <u>Members See Also Send Feedback</u> Core implementation of the <u>IMock<(Of <(T>)></u>) interface.

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

Syntax

C#

public class Mock<T> : Mock, IMock<T>
where T : class

Type Parameters

Т

Type to mock.

Inheritance Hierarchy

Object

<u>Moq..::.Mock</u> Moq..::.Mock<(Of <(T>)>)

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock<(Of <(T>)>) Members
 Mock<(Of <(T>)>) Class Constructors Methods Properties See Also Send Feedback

The $Mock \le (Of \le (T \ge))$ type exposes the following members.

Constructors

NameDescription \blacksquare Mock<(Of <(T>)>)Overloaded.

Methods

	Name	Description
≡∲	<u>As≤(Of</u> ≤(TInterface≥)≥)	Implements <u>As<(Of <(TInterface>)>)()()()</u> .
≡∳	<u>CreateEventHandler<(Of</u> < <u>(TEventArgs>)>)</u>	Implements <u>CreateEventHandler<(Of</u> < <u>(TEventArgs>)>)()()()</u> . (Inherited from <u>Mock</u> .)
=	<u>Expect</u>	Overloaded.
≡∳	<u>ExpectGet<(Of</u> < <u>(TProperty>)></u>)	Implements <u>ExpectGet<(Of <(TProperty>)>)</u> (Expression<(Of <(Func<(Of <(T, TProperty>)>)>))).
=0	<u>ExpectSet</u>	Overloaded.
	<u>GetObject</u>	Returns the mocked object value. (Overrides <u>Mock::.GetObject()()(</u>).)
=0	Verify	Overloaded.
≡ 	<u>VerifyAll</u>	Implements <u>VerifyAll()()()</u> . (Overrides <u>Mock::.VerifyAll()()()</u> .)
≡∳	<u>VerifyGet≤(Of</u> ≤ <u>(TProperty≥)≥</u>)	Implements VerifyGet<(Of <(TProperty>)>)(Expression<(Of <(Func<(Of <(T,
=0	<u>VerifySet</u>	Overloaded.

Properties

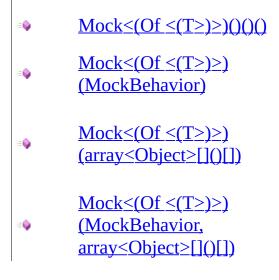
	Name	Description
2	<u>Behavior</u>	Implements <u>Behavior</u> . (Inherited from <u>Mock</u> .)
2	<u>CallBase</u>	Implements <u>CallBase</u> . (Inherited from <u>Mock</u> .)
	<u>DefaultValue</u>	Implements <u>DefaultValue</u> . (Inherited from <u>Mock</u> .)
*	ImplementedInterfaces	Exposes the list of extra interfaces implemented by the mock. (Inherited from <u>Mock</u> .)
	<u>Object</u>	Exposes the mocked object instance.

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

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock<(Of <(T>)>) Constructor
 Mock<(Of <(T>)>) Class See Also Send Feedback

Overload List

Name



Description

Initializes an instance of the mock with <u>default behavior</u>.

Initializes an instance of the mock with the specified <u>behavior</u>.

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

Initializes an instance of the mock with a specific <u>behavior</u> with the given constructor arguments for the class.

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

□ C#
 Moq
 Mock<(Of <(T>)>) Constructor
 Mock<(Of <(T>)>) Class Example See Also Send Feedback
 Initializes an instance of the mock with <u>default behavior</u>.

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

Syntax

C#

public Mock()

Examples

C#

Сору

var mock = new Mock<IFormatProvider>();

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ 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#

Parameters

behavior

Type: <u>Moq..::.MockBehavior</u> Behavior of the mock.

Examples

C#

Сору

var mock = new Mock<IFormatProvider>(MockBehavior.Relaxed);

See Also

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq Mock<(Of <(T>)>) Constructor (array<>[]()[]) Mock<(Of <(T>)>) Class Example See Also Send Feedback

Initializes an instance of the mock with <u>default behavior</u> 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#

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#

Сору

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

See Also

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq Mock<(Of <(T>)>) Constructor (MockBehavior, array<>[]()[]) <u>Mock<(Of <(T>)>) Class</u> <u>Example</u> <u>See Also</u> <u>Send Feedback</u>

Initializes an instance of the mock with a specific <u>behavior</u> 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#

Parameters

behavior

Type: <u>Moq..::.MockBehavior</u> 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#

Сору

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

See Also

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock<(Of <(T>)>) Methods
 Mock<(Of <(T>)>) Class See Also Send Feedback

The $Mock \le (Of \le (T \ge))$ type exposes the following members.

Methods

	Name	Description
≡∲	<u>As≤(Of</u> ≤(TInterface≥)≥)	Implements <u>As<(Of <(TInterface>)>)()()()</u> .
≡∳	<u>CreateEventHandler<(Of</u> < <u>(TEventArgs>)>)</u>	Implements <u>CreateEventHandler<(Of</u> < <u>(TEventArgs>)>)()()()</u> . (Inherited from <u>Mock</u> .)
=	<u>Expect</u>	Overloaded.
≡∳	<u>ExpectGet<(Of</u> < <u>(TProperty>)></u>)	Implements <u>ExpectGet<(Of <(TProperty>)>)</u> (Expression<(Of <(Func<(Of <(T, TProperty>)>)>)).
=0	<u>ExpectSet</u>	Overloaded.
	<u>GetObject</u>	Returns the mocked object value. (Overrides <u>Mock::.GetObject()()(</u>).)
=0	Verify	Overloaded.
≡Q	<u>VerifyAll</u>	Implements <u>VerifyAll()()()</u> . (Overrides <u>Mock::.VerifyAll()()()</u> .)
≡∳	<u>VerifyGet≤(Of</u> ≤ <u>(TProperty≥)≥</u>)	Implements VerifyGet<(Of <(TProperty>)>)(Expression<(Of <(Func<(Of <(T,
=0	<u>VerifySet</u>	Overloaded.

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq Mock<(Of <(T>)>)..::.As<(Of <(TInterface>)>) Method <u>Mock<(Of <(T>)>) Class See Also Send Feedback</u> Implements <u>As<(Of <(TInterface>)>)()()()</u>.

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

Syntax

C#

public virtual <u>IMock</u><TInterface> As<TInterface>()
where TInterface : class

Type Parameters

TInterface

Type of interface to cast the mock to.



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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock<(Of <(T>)>)..::.Expect Method
 Mock<(Of <(T>)>) Class See Also Send Feedback

Overload List

Name

- Expect(Expression<(Of <(Action<(Of <(T>)>)>))) Expect<(Of <(TResult>)>)
- (Expression<(Of <(Func<(Of <(T, TResult>)>)>)))

Description

$$\label{eq:constraint} \begin{split} & \text{Implements } \underline{\text{Expect}(\text{Expression} < (Of < (Of < (T >) >) >) >))).} \\ & \text{Implements } \underline{\text{Expect} < (Of} < (T \text{Result} >) >) (E \text{Expression} < (Of < (F \text{unc} < (Of < (T, T \text{Result} >) >) >))).} \end{split}$$

See Also

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

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#

Parameters

expression

Type: Expression<(Of <(Action<(Of $<(\underline{T}>)>)>)>$) Lambda expression that specifies the expected method invocation.

Implements

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 <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

Mock<(Of <(T>)>)..::.Expect<(Of <(TResult>)>) Method (<(Of <(<(Of <(T, TResult>)>)>)>)))

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

Implements <u>Expect<(Of <(TResult>)>)(Expression<(Of <(Func<(Of <(T, TResult>)>)>)>))</u>).

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

Syntax

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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 <u>moqdisc@googlegroups.com</u>

■ □ C# Moq Mock<(Of <(T>)>)..::ExpectGet<(Of <(TProperty>)>) Method <u>Mock<(Of <(T>)>) Class</u> <u>See Also</u> <u>Send Feedback</u>

Implements <u>ExpectGet<(Of <(TProperty>)>)(Expression<(Of <(Func<(Of <(T, TProperty>)>)>)>))</u>.

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

Syntax

C#

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.



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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock<(Of <(T>)>)..::.ExpectSet Method
 Mock<(Of <(T>)>) Class See Also Send Feedback

Overload List

=**i**

Name

ExpectSet<(Of <(TProperty>)>) (Expression<(Of <(Func<(Of =Q <(T, TProperty>)>)>)))

ExpectSet<(Of <(TProperty>)>) (Expression<(Of <(Func<(Of <<u>(T, TProperty>)>)>)>)</u>, TProperty)

Description

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

Implements <a href="mailto:ExpectSet<(Of"><u>ExpectSet<(Of</u> <(TProperty>)>)(Expression<(Of <(Func<(Of <(T, TProperty>)>)>))), TProperty).

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

Send comments on this topic to moqdisc@googlegroups.com

-

□ C#

Moq

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

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

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

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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.

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

-

□ C#

Moq

Mock<(Of <(T>)>)..::.ExpectSet<(Of <(TProperty>)>) Method (<(Of <(<(Of <(T, TProperty>)>)>)>), TProperty) <u>Mock<(Of <(T>)>) Class</u> <u>See Also</u> <u>Send Feedback</u>

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

C#

)

Parameters

expression

Type: Expression<(Of <(Func<(Of <(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.

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

□ C#
 Moq
 Mock<(Of <(T>)>)..::.GetObject Method
 Mock<(Of <(T>)>) Class See Also Send Feedback
 Returns the mocked object value.

C#

protected override Object GetObject()



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

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock<(Of <(T>)>)..::.Verify Method
 Mock<(Of <(T>)>) Class See Also Send Feedback

Overload List

Name

- Verify()()()
- <u>Verify(Expression<(Of</u>
- (Action<(Of <(T>)>)>)))
 - <u>Verify<(Of <(TResult>)>)</u>
- (Expression<(Of <(Func<(Of <(T, TResult>)>)>)))

Description

Implements Verify()()(.) (Overrides Mock..::.Verify()()().) Implements Verify(Expression < (Of < (Action < (Of < (T >) >) >))). Implements Verify < (Of < (TResult >) >)(Expression < (Of < (Func < (Of < (T, TResult >) >) >))).

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

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq Mock<(Of <(T>)>)..::.Verify Method <u>Mock<(Of <(T>)>) Class See Also Send Feedback</u> Implements <u>Verify()()()</u>.

C#

public override void Verify()

Implements

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

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

■
□ C#
Moq
Mock<(Of <(T>)>)...:.Verify Method (<(Of <(<(Of <(T>)>)>)>)))
Mock<(Of <(T>)>) Class See Also Send Feedback
Implements Verify(Expression<(Of <(Action<(Of <(T>)>)>))).

C#

Parameters

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

Implements

IMock<(Of <(T>)>)..::.Verify(Expression<(Of <(Action<(Of <(T>)>)>)>)))

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

□ C# Moq Mock<(Of <(T>)>)..::.Verify<(Of <(TResult>)>) Method (<(Of <(<(Of <(T, TResult>)>)>)))) Mock<(Of <(T>)>)>))) Mock<(Of <(T>)>)Class See Also Send Feedback Implements Verify<(Of <(TPecult>)>)(Expression<(Of <(Func<(Of <(T)))))

Implements <u>Verify<(Of <(TResult>)>)(Expression<(Of <(Func<(Of <(T, TResult>)>)>)>))</u>.

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, TResult>)>)>)>) Expression to verify.

Type Parameters

TResult

Type of return value from the expression.

Implements

IMock<(Of <(T>)>)..::.Verify(Expression<(Of <(Action<(Of <(T>)>)>)>)))

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

■
□ C#
Moq
Mock<(Of <(T>)>)...:.VerifyAll Method
Mock<(Of <(T≥)>) Class See Also Send Feedback
Implements VerifyAll()()().

C#

public override void VerifyAll()

Implements

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



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

■ □ C# Moq Mock<(Of <(T>)>)..::.VerifyGet<(Of <(TProperty>)>) Method <u>Mock<(Of <(T>)>) Class</u> <u>See Also</u> <u>Send Feedback</u> Implements VerifyCet<(Of <(TProperty>)>)(European <(Of <(CProperty>)>))

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

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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

<u>IMock<(Of <(T>)>)..::.VerifyGet<(Of <(TProperty>)>)(Expression<(Of <(Func<(Of <(T, TProperty>)>)>))</u>)



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

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock<(Of <(T>)>)..::.VerifySet Method
 Mock<(Of <(T>)>) Class See Also Send Feedback

Overload List

=**i**

Name

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

> <u>VerifySet<(Of <(TProperty>)>)</u> (Expression<(Of <(Func<(Of <(T, TProperty>)>)>))), TProperty)

Description

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

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

Send comments on this topic to moqdisc@googlegroups.com

-

□ C#

Moq

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

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

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

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

Syntax

C#

Parameters

expression

Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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

<u>Mock<(Of <(T>)>) Class</u> <u>VerifySet Overload</u> <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

Mock<(Of <(T>)>)..::.VerifySet<(Of <(TProperty>)>) Method (<(Of <(<(Of <(T, TProperty>)>)>)>)), TProperty) <u>Mock<(Of <(T>)>) Class</u> <u>See Also</u> <u>Send Feedback</u>

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

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,
        TProperty value
)
```

Parameters

expression

```
Type: Expression<(Of <(Func<(Of <(<u>T</u>, 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

<u>IMock<(Of <(T>)>)..::.VerifySet<(Of <(TProperty>)>)(Expression<(Of <(Func<(Of <(T, TProperty>)>)>)), TProperty)</u>

See Also

<u>Mock<(Of <(T>)>) Class</u> <u>VerifySet Overload</u> <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 Mock<(Of <(T>)>) Properties
 Mock<(Of <(T>)>) Class See Also Send Feedback

The $Mock \le (Of \le (T \ge))$ type exposes the following members.

Properties

	Name	Description
2	<u>Behavior</u>	Implements <u>Behavior</u> . (Inherited from <u>Mock</u> .)
2	<u>CallBase</u>	Implements <u>CallBase</u> . (Inherited from <u>Mock</u> .)
	<u>DefaultValue</u>	Implements <u>DefaultValue</u> . (Inherited from <u>Mock</u> .)
*	ImplementedInterfaces	Exposes the list of extra interfaces implemented by the mock. (Inherited from <u>Mock</u> .)
	<u>Object</u>	Exposes the mocked object instance.

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 Mock<(Of <(T>)>)..::.Object Property
 Mock<(Of <(T>)>).Class See Also Send Feedback
 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 <(T>)>)..:.Object



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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq MockBehavior Enumeration <u>See Also</u> <u>Send Feedback</u>

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

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



Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 MockedEvent Class
 Members See Also Send Feedback

Represents a generic event that has been mocked and can be rised.

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

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockedEvent Members
 MockedEvent Class Methods Events See Also Send Feedback

The <u>MockedEvent</u> type exposes the following members.

Operators

Name

Description

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

Events

Name

Description

Raised Event raised whenever the mocked event is rised.



MockedEvent Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockedEvent Methods
 <u>MockedEvent Class</u> See Also Send Feedback

The <u>MockedEvent</u> type exposes the following members.

Operators

Name

Description

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



MockedEvent Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq MockedEvent..::.Implicit Operator <u>MockedEvent Class</u> <u>See Also</u> <u>Send Feedback</u>

Provides support for attaching a <u>MockedEvent</u> to a generic EventHandler event.

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



C#

Parameters

mockEvent Type: <u>Moq..::MockedEvent</u> Event to convert.



<u>MockedEvent Class</u> <u>Moq Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockedEvent Events
 MockedEvent Class See Also Send Feedback

The <u>MockedEvent</u> type exposes the following members.

Events

Name

Description

Raised Event raised whenever the mocked event is rised.



MockedEvent Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 MockedEvent..::.Raised Event
 MockedEvent Class See Also Send Feedback
 Event raised whenever the mocked event is rised.

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

Syntax

C#

public event EventHandler Raised



<u>MockedEvent Class</u> <u>Moq Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 MockedEvent<(Of <(TEventArgs>)>) Class
 Members See Also Send Feedback
 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#

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<(Of <(TEventArgs>)>)

See Also

<u>MockedEvent<(Of <(TEventArgs>)>)</u> <u>Members</u> <u>Moq Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

□ Include Protected Members

 \square Include Inherited Members

Moq

MockedEvent<(Of <(TEventArgs>)>) Members

<u>MockedEvent<(Of <(TEventArgs>)>) Class</u> <u>Methods</u> <u>Events</u> <u>See Also</u> <u>Send</u> <u>Feedback</u>

The <u>MockedEvent<(Of <(TEventArgs>)></u>) type exposes the following members.

Methods

Name

Description

Raise Raises the associated event with the given event argument data.

Operators

Events

Name

Description

Raised Event raised whenever the mocked event is rised.
 (Inherited from MockedEvent.)

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockedEvent<(Of <(TEventArgs>)>) Methods
 MockedEvent<(Of <(TEventArgs>)>) Class See Also Send Feedback

The <u>MockedEvent<(Of <(TEventArgs>)></u>) type exposes the following members.

Methods

Name

Description

Raise Raises the associated event with the given event argument data.

Operators

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq MockedEvent<(Of <(TEventArgs>)>)..::.Implicit Operator <u>MockedEvent<(Of <(TEventArgs>)>) Class See Also Send Feedback</u>

Provides support for attaching a <u>MockedEvent<(Of <(TEventArgs>)>)</u> to a generic EventHandler event.

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



C#

Parameters

mockEvent

Type: <u>Moq..::.MockedEvent</u><(Of <(<u>TEventArgs</u>>)>) Event to convert.

See Also

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq MockedEvent<(Of <(TEventArgs>)>)..::.Raise Method <u>MockedEvent<(Of <(TEventArgs>)>) Class</u> <u>See Also</u> <u>Send Feedback</u>

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#

Parameters

args

Type: <u>TEventArgs</u> Data to pass to the event.

See Also

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockedEvent<(Of <(TEventArgs>)>) Events
 MockedEvent<(Of <(TEventArgs>)>) Class See Also Send Feedback

The <u>MockedEvent<(Of <(TEventArgs>)></u>) type exposes the following members.

Events

Name

Description

Raised Event raised whenever the mocked event is rised.
 (Inherited from MockedEvent.)

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq MockException Class <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

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 dissapear 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

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockException Members
 MockException Class Constructors Methods See Also Send Feedback

The <u>MockException</u> type exposes the following members.

Constructors

Name

Description

MockException Supports the serialization infrastructure.

Methods

Name Description
 Supports the serialization infrastructure.
 GetObjectData (Overrides ExceptionGetObjectData(SerializationInfo, StreamingContext).)



MockException Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 MockException Constructor
 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#

protected MockException(
 SerializationInfo info,
 StreamingContext context
)

Parameters

info

Type: SerializationInfo Serialization information.

context

Type: StreamingContext Streaming context.



MockException Class Moq Namespace

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockException Methods
 <u>MockException Class</u> See Also Send Feedback

The <u>MockException</u> type exposes the following members.

Methods

Name Description
 Supports the serialization infrastructure.
 GetObjectData (Overrides ExceptionGetObjectData(SerializationInfo, StreamingContext).)



MockException Class Moq Namespace

□ C#
 Moq
 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#

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)



MockException Class Moq Namespace

■ □ C# Moq MockFactory Class <u>Members Example</u> <u>See Also</u> <u>Send Feedback</u>

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 <u>MockBehavior</u> (if different than the <u>Default</u> 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 <u>MockBehavior</u> (unless overriden by calling <u>Create<(Of <(T>)>)(MockBehavior</u>)) and posterior verification.

Examples

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

```
C# Copy
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:

C#

_ Copy

```
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#

Copy

```
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 <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockFactory Members
 MockFactory Class Constructors Methods Properties See Also Send

Feedback

The <u>MockFactory</u> type exposes the following members.

Constructors

Name

Description

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

Methods

	Name	Description
=0	<u>Create</u>	Overloaded.
ē	$\frac{\text{CreateMock} \leq (\text{Of} \leq (T \geq)))}{\leq (T \geq)}$	Implements creation of a new mock within the factory.
=	Verify	Verifies all verifiable expectations on all mocks created by this factory.
≡Q	<u>VerifyAll</u>	Verifies all verifiable expectations on all mocks created by this factory.
ġ Q	<u>VerifyMocks</u>	Invokes verifyAction for each mock in <u>Mocks</u> , and accumulates the resulting MockVerificationException that might be thrown from the action.

Properties

Name	Description
	Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to trueTruetruetrue (True in Visual Basic).
DefaultValue	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks.
	Gets the mocks that have been created by this factory and that will get verified together.



MockFactory Class Moq Namespace

■ □ C# Moq MockFactory Constructor <u>MockFactory Class</u> <u>See Also</u> <u>Send Feedback</u>

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#

Parameters

defaultBehavior

Type: Moq..::.MockBehavior

The behavior to use for mocks created using the $\underline{\text{Create}}(Of \leq (T \geq) \geq)()()()$ factory method if not overriden by using the $\underline{\text{Create}}(Of \leq (T \geq) \geq)$ (MockBehavior) overload.



MockFactory Class Moq Namespace

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockFactory Methods
 MockFactory Class See Also Send Feedback

The <u>MockFactory</u> type exposes the following members.

Methods

	Name	Description
=0	<u>Create</u>	Overloaded.
ē	$\frac{\text{CreateMock} \leq (\text{Of} \leq (T \geq)))}{\leq (T \geq)}$	Implements creation of a new mock within the factory.
=	Verify	Verifies all verifiable expectations on all mocks created by this factory.
≡Q	<u>VerifyAll</u>	Verifies all verifiable expectations on all mocks created by this factory.
ġ Q	<u>VerifyMocks</u>	Invokes verifyAction for each mock in <u>Mocks</u> , and accumulates the resulting MockVerificationException that might be thrown from the action.

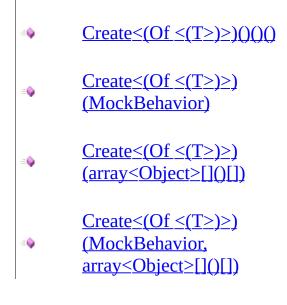


MockFactory Class Moq Namespace

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockFactory..::.Create Method
 MockFactory Class See Also Send Feedback

Overload List

Name



Description

Creates a new mock with the default <u>MockBehavior</u> specified at factory construction time.

Creates a new mock with the given behavior.

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

See Also

MockFactory Class MockFactory Members Moq Namespace

■ □ C# Moq MockFactory..::.Create<(Of <(T>)>) Method <u>MockFactory Class</u> Example See Also Send Feedback

Creates a new mock with the default <u>MockBehavior</u> 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

Т

Type to mock.

Return Value

A new $\underline{Mock} \leq (Of \leq (T \geq) \geq)$.

Examples

C#

Сору

var factory = new MockFactory(MockBehavior.Strict);

var foo = factory.Create<IFoo>();
// use mock on tests

factory.VerifyAll();



MockFactory Class Create Overload Moq Namespace

■ □ C# Moq MockFactory..::.Create<(Of <(T>)>) Method (MockBehavior) <u>MockFactory Class</u> <u>Example</u> <u>See Also</u> <u>Send Feedback</u> 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#

Parameters

behavior

Type: <u>Moq..::MockBehavior</u> Behavior to use for the mock, which overrides the default behavior

specified at factory construction time.

Type Parameters

Т

Type to mock.

Return Value

A new $\underline{Mock} \leq (Of \leq (T \geq) \geq)$.

Examples

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

C# Copy var factory = new MockFactory(MockBehavior.Strict);

var foo = factory.Create<IFoo>(MockBehavior.Loose);



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 <u>MockBehavior</u> 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#

Parameters

args

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

Type Parameters

Т

Type to mock.

Return Value

A new $\underline{Mock} \leq (Of \leq (T \geq) \geq)$.

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#

Сору

var factory = new MockFactory(MockBehavior.Default);

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

factory.Verify();



MockFactory Class Create Overload Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

-

□ C#

Moq

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

Creates a new mock with the given behavior 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#

Parameters

behavior

Type: <u>Moq..::MockBehavior</u>

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

Т

Type to mock.

Return Value

A new $\underline{Mock} \leq (Of \leq (T \geq) \geq)$.

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# Copy var factory = new MockFactory(MockBehavior.Default); var mock = factory.Create<MyBase>(MockBehavior.Strict, "Foo", 25, tr



MockFactory Class Create Overload Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq MockFactory..::.CreateMock<(Of <(T>)>) Method <u>MockFactory Class</u> <u>See Also</u> <u>Send Feedback</u> 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#

Parameters

behavior

Type: <u>Moq..::MockBehavior</u> The behavior for the new mock.

args

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

Type Parameters

Т

Type to mock.



MockFactory Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq MockFactory..::.Verify Method <u>MockFactory Class</u> <u>See Also</u> <u>Send Feedback</u>

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

Exception

Condition

Moq..::.MockException One or more mocks had expectations that were not satisfied.

See Also

<u>MockFactory Class</u> <u>Moq Namespace</u> <u>Mock<(Of <(T>)>)..::.Verify()()()</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq MockFactory..::.VerifyAll Method <u>MockFactory Class</u> <u>See Also</u> <u>Send Feedback</u>

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 VerifyAll()

Exceptions

Exception

Condition

Moq..::.MockException One or more mocks had expectations that were not satisfied.

See Also

<u>MockFactory Class</u> <u>Moq Namespace</u> <u>Mock<(Of <(T>)>)..::.Verify()()()</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq MockFactory..::.VerifyMocks Method <u>MockFactory Class</u> <u>See Also</u> <u>Send Feedback</u>

Invokes verifyAction for each mock in <u>Mocks</u>, 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#

Parameters

verifyAction

Type: Action<(Of <(<u>IMock</u>>)>) The action to execute against each mock.



MockFactory Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 MockFactory Properties
 <u>MockFactory Class See Also Send Feedback</u>

The <u>MockFactory</u> type exposes the following members.

Properties

Name	Description
	Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to trueTruetruetrue (True in Visual Basic).
DefaultValue	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks.
	Gets the mocks that have been created by this factory and that will get verified together.



MockFactory Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq MockFactory..::.CallBase Property <u>MockFactory Class</u> <u>See Also</u> <u>Send Feedback</u>

Whether the base member virtual implementation will be called for mocked classes if no expectation is met. Defaults to trueTruetruetrue (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; }



MockFactory Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq MockFactory..::.DefaultValue Property <u>MockFactory Class</u> <u>See Also</u> <u>Send Feedback</u>

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 { get; set; }



MockFactory Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq MockFactory..::.Mocks Property <u>MockFactory Class</u> <u>See Also</u> <u>Send Feedback</u>

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)



C#

protected IEnumerable<<u>IMock</u>> Mocks { get; }



MockFactory Class Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq Range Enumeration <u>See Also</u> <u>Send Feedback</u>

Kind of range to use in a filter specified through <u>IsInRange<(Of <(TValue>)>)</u> (<u>TValue, TValue, Range</u>).

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

Syntax

C#

public enum Range

Members

Member name

Description

InclusiveThe range includes the to and from values.ExclusiveThe range does not include the to and from values.



Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq Moq.Language Namespace <u>Send Feedback</u>

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

	Interface	Description
~0	<u>ICallback</u>	Defines the Callback verb and overloads.
~0	<u>ICallback<(Of</u> < <u>(TResult>)>)</u>	Defines the Callback verb and overloads for callbacks on expectations that return a value.
°-0	<u>ICallbackGetter<(Of</u> < <u>(TProperty>)>)</u>	Defines the Callback verb for property getter expectations.
~0	<u>ICallbackSetter<(Of</u> < <u>(TProperty>)>)</u>	Defines the Callback verb for property setter expectations.
~0	<u>INever</u>	Defines the Never verb.
~0	<u>IOccurrence</u>	Defines occurrence members to constraint expectations.
~0	<u>IRaise</u>	Defines the Raises verb.
~0	<u>IReturns</u>	Base interface for <u>IReturns<(Of <(TResult>)></u>).
°-0	<u>IReturns<(Of</u> < <u>(TResult>)</u> >)	Defines the Returns verb.
⊶ 0	<u>IReturnsGetter≤(Of</u> ≤ <u>(TProperty≥)≥)</u>	Defines the Returns verb for property get expectations.
~0	<u>IThrows</u>	Defines the Throws verb.
°~0	<u>IVerifies</u>	Defines the Verifiable verb.

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C#
Moq
ICallback Interface
Members See Also Send Feedback
Defines the Callback verb and overloads.

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

Syntax

C#

public interface ICallback

See Also

ICallback Members Moq.Language Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallback Members
 ICallback Interface Methods See Also Send Feedback

Methods

Name Description

<u>Callback</u> Overloaded.

See Also

ICallback Interface Moq.Language Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallback Methods
 ICallback Interface See Also Send Feedback

Methods

Name Description

<u>Callback</u> Overloaded.



ICallback Interface Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallback..::.Callback Method
 ICallback Interface See Also Send Feedback

Overload List

Name

- Callback(Action)
- $\stackrel{\text{Callback} < (Of < (T >) >)}{(Action < (Of < (T >) >))}$
- ▲ <u>Callback<(Of <(T1, T2>)>)</u> (Action<(Of <(T1, T2>)>))
- <u>Callback<(Of <(T1, T2,</u>
- <u>T3>)>)(Action<(Of <(T1, T2, T3>)>))</u>
 - <u>Callback<(Of <(T1, T2, T3, T4>)>)(Action<(Of <(T1, T2, T3, T4>))</u>

Description

Specifies a callback to invoke when the method is called.

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

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

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

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



ICallback Interface ICallback Members Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 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:

C#

Сору

```
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 <u>moqdisc@googlegroups.com</u>

■ □ C# Moq ICallback..::.Callback<(Of <(T>)>) Method (<(Of <(T>)>)) <u>ICallback Interface Example See Also Send Feedback</u>

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#

Parameters

callback

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

Type Parameters

Т

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#

Сору

```
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 <u>moqdisc@googlegroups.com</u>

■ □ C# Moq ICallback..::.Callback<(Of <(T1, T2>)>) Method (<(Of <(T1, T2>)>)) <u>ICallback Interface Example See Also Send Feedback</u>

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#

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#

Copy

ICallback Interface Callback Overload Moq.Language Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C# Moq

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#

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.

Т3

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#

Сору

ICallback Interface Callback Overload Moq.Language Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C# Moq 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#

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.

Т3

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#

Сору

ICallback Interface Callback Overload Moq.Language Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq ICallback<(Of <(TResult>)>) Interface <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

Defines the Callback verb and overloads for callbacks on expectations that return a value.

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

Syntax

C#

public interface ICallback<TResult>

Type Parameters

TResult

Type of the return value of the expectation.

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallback<(Of <(TResult>)>) Members
 ICallback<(Of <(TResult>)>) Interface Methods See Also Send Feedback

Methods

Name Description

<u>Callback</u> Overloaded.

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallback<(Of <(TResult>)>) Methods
 ICallback<(Of <(TResult>)>) Interface See Also Send Feedback

Methods

Name Description

<u>Callback</u> Overloaded.

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

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallback<(Of <(TResult>)>)..::.Callback Method
 ICallback<(Of <(TResult>)>) Interface See Also Send Feedback

Overload List

Name

- Callback(Action)
- $\stackrel{\text{Callback} < (Of < (T >) >)}{(Action < (Of < (T >) >))}$
- ▲ <u>Callback<(Of <(T1, T2>)>)</u> (Action<(Of <(T1, T2>)>))
- <u>Callback<(Of <(T1, T2,</u>
- <u>T3>)>)(Action<(Of <(T1, T2, T3>)>))</u>
 - <u>Callback<(Of <(T1, T2, T3, T4>)>)(Action<(Of <(T1, T2, T3, T4>))</u>

Description

Specifies a callback to invoke when the method is called.

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

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

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

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

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#

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#

Copy

```
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.

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

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#

Parameters

callback

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

Type Parameters

Т

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#

Сору

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

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C# Moq 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#

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#

Copy

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C# Moq 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#

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.

Т3

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#

Copy

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

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#

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.

Т3

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:

Copy

C#

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq ICallbackGetter<(Of <(TProperty>)>) Interface <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

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.

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

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallbackGetter<(Of <(TProperty>)>) Members
 ICallbackGetter<(Of <(TProperty>)>) Interface Methods See Also Send Feedback

The <u>ICallbackGetter<(Of <(TProperty>)>)</u> type exposes the following members.

Methods

Name

Description

• <u>Callback</u> Specifies a callback to invoke when the property is retrieved.

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

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallbackGetter<(Of <(TProperty>)>) Methods
 ICallbackGetter<(Of <(TProperty>)>) Interface See Also Send Feedback

The <u>ICallbackGetter<(Of <(TProperty>)>)</u> type exposes the following members.

Methods

Name

Description

• <u>Callback</u> Specifies a callback to invoke when the property is retrieved.

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

Send comments on this topic to moqdisc@googlegroups.com

■ C# Moq 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#

Parameters

callback Type: Action Callback method to invoke.

Examples

Invokes the given callback with the property value being set.

C#

Сору

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

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq ICallbackSetter<(Of <(TProperty>)>) Interface <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

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.

$$\label{eq:logithtarge} \begin{split} & \underline{\text{ICallbackSetter}}(Of <(\underline{\text{TProperty}}) \geq) \underline{\text{Members}} \\ & \underline{\text{Moq.Language Namespace}} \\ & \underline{\text{IMock}}(Of <(\underline{\text{T}}) \geq) \dots \\ & \underline{\text{Expression}}(Of <(\underline{\text{TProperty}}) \geq) \geq) \\ & \underline{\text{Superator}}(Of <(\underline{\text{TProperty}}) \geq) \geq) \\ & \underline{\text{IMock}}(Of <(\underline{\text{T}}) \geq) \dots \\ & \underline{\text{Expression}}(Of <(\underline{\text{TProperty}}) \geq) \geq) \\ & \underline{\text{IMock}}(Of <(\underline{\text{TProperty}}) \geq) \geq) \\ & \underline{\text{Superator}}(Of <(\underline{\text{TProperty}}) \geq) \geq (\underline{\text{TProperty}}) \geq 0 \\ & \underline{\text{Superator}}(Of <(\underline{\text{TProperty}}) \geq) \geq) \\ & \underline{\text{Superator}}(Of <(\underline{\text{TProperty}}) \geq) \geq 0 \\ & \underline{\text{Superator}}(Of <(\underline{\text{TProperty}}) \geq) \geq 0 \\ & \underline{\text{Superator}}(Of <(\underline{\text{TProperty}}) \geq) \geq 0 \\ & \underline{\text{Superator}}(Of <(\underline{\text{TProperty}}) \geq 0 \\ & \underline{\text{TProperty}}(Of <(\underline{\text{TProperty}}) \geq$$

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallbackSetter<(Of <(TProperty>)>) Members
 ICallbackSetter<(Of <(TProperty>)>) Interface Methods See Also Send Feedback

The <u>ICallbackSetter<(Of <(TProperty>)></u>) type exposes the following members.

Methods

Name

Description

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

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

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ICallbackSetter<(Of <(TProperty>)>) Methods
 ICallbackSetter<(Of <(TProperty>)>) Interface See Also Send Feedback

The <u>ICallbackSetter<(Of <(TProperty>)></u>) type exposes the following members.

Methods

Name

Description

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

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

```
■

□ C#

Moq

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#

Parameters

callback

Type: Action<(Of <(<u>TProperty</u>>)>) Callback method to invoke.

Examples

Invokes the given callback with the property value being set.

C#

Сору

mock.ExpectSet(x => x.Suspended)
 .Callback((bool state) => Console.WriteLine(state));

See Also

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

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C#
Moq
INever Interface
Members See Also Send Feedback
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

<u>INever Members</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 INever Members
 INever Interface Methods See Also Send Feedback

The **INever** type exposes the following members.

Methods

Name

Description

Never The expected invocation is never expected to happen.



INever Interface Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 INever Methods
 <u>INever Interface</u> See Also Send Feedback

The **INever** type exposes the following members.

Methods

Name

Description

Never The expected invocation is never expected to happen.



INever Interface Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq INever..::.Never Method <u>INever Interface Example See Also Send Feedback</u> 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 inmediately as the invocations are performed, like strict mocks do with unexpected invocations.

Examples

C#

Сору

```
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 <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IOccurrence Interface <u>Members See Also Send Feedback</u>

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 <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IOccurrence Members
 IOccurrence Interface Methods See Also Send Feedback

The <u>IOccurrence</u> type exposes the following members.

Methods

	Name	Description
- :	<u>AtMost</u>	The expected invocation can happen at most specified number of times.
=	<u>AtMostOnce</u>	The expected invocation can happen at most once.



IOccurrence Interface Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IOccurrence Methods
 <u>IOccurrence Interface</u> See Also Send Feedback

The <u>IOccurrence</u> type exposes the following members.

Methods

	Name	Description
- :	<u>AtMost</u>	The expected invocation can happen at most specified number of times.
=	<u>AtMostOnce</u>	The expected invocation can happen at most once.



IOccurrence Interface Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 IOccurrence..::.AtMost Method
 <u>IOccurrence Interface Example See Also Send Feedback</u>
 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#

Parameters

callCount

Type: Int32

[Missing <param name="callCount"/> documentation for "M:Moq.Language.IOccurrence.AtMost(System.Int32)"]

Examples

C#

Сору

```
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 <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IOccurrence..::.AtMostOnce Method <u>IOccurrence Interface Example See Also Send Feedback</u> 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#

Сору

```
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 <u>moqdisc@googlegroups.com</u>

□ C#
Moq
IRaise Interface
Members See Also Send Feedback
Defines the Raises verb.

Syntax

C#

public interface IRaise

IRaise Members Moq.Language Namespace

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IRaise Members
 IRaise Interface Methods See Also Send Feedback

Methods

Name Description

■ <u>Raises</u> Overloaded.

IRaise Interface Moq.Language Namespace

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IRaise Methods
 IRaise Interface See Also Send Feedback

Methods

Name Description

■ <u>Raises</u> Overloaded.



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

Name

≡ 	<u>Raises(MockedEvent, EventArgs)</u>
≡.	<u>Raises(MockedEvent, Func≤(Of</u> <u><(EventArgs≥)≥))</u>
≡∳	<u>Raises<(Of <(T≥)>)(MockedEvent,</u> <u>Func<(Of <(T, EventArgs≥)≥))</u>
≡∳	<u>Raises<(Of <(T1, T2≥)>)</u> (<u>MockedEvent, Func≤(Of ≤(T1, T2,</u> <u>EventArgs≥)>))</u>
= Q	<u>Raises≤(Of ≤(T1, T2, T3≥)≥)</u> (<u>MockedEvent, Func≤(Of ≤(T1, T2,</u> <u>T3, EventArgs≥)≥)</u>)
≡∳	<u>Raises<(Of <(T1, T2, T3, T4≥)>)</u> (<u>MockedEvent, Func<(Of <(T1, T2, T3, T4, EventArgs≥)>)</u>)

Description

Specifies the mocked event that will be raised when the expectation is met.

Specifies the mocked event that will be raised when the expectation is met.

Specifies the mocked event that will be raised when the expectation is met.

Specifies the mocked event that will be raised when the expectation is met.

Specifies the mocked event that will be raised when the expectation is met.

Specifies the mocked event that will be raised when the expectation is met.

<u>IRaise Interface</u> <u>IRaise Members</u> <u>Moq.Language Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IRaise..::.Raises Method (MockedEvent,) <u>IRaise Interface Example See Also Send Feedback</u> Specifies the mocked event that will be raised when the

Specifies the mocked event that will be raised when the expectation is met.

Syntax

C#

<u>IVerifies</u> Raises(<u>MockedEvent</u> eventHandler, EventArgs args)

Parameters

eventHandler

Type: <u>Moq..::.MockedEvent</u> The mocked event, retrieved from <u>CreateEventHandler()()()</u> or <u>CreateEventHandler<(Of <(TEventArgs>)>)()()()</u>.

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:

C# Copy

IRaise Interface Raises Overload Moq.Language Namespace

■ □ C# Moq IRaise..::.Raises Method (MockedEvent, <(Of <(>)>)) <u>IRaise Interface</u> <u>See Also</u> <u>Send Feedback</u>

Specifies the mocked event that will be raised when the expectation is met.

Syntax

C#

<u>IVerifies</u> Raises(<u>MockedEvent</u> eventHandler, Func<EventArgs> func)

Parameters

eventHandler

Type: <u>Moq..::.MockedEvent</u> The mocked event, retrieved from <u>CreateEventHandler()()()</u> or <u>CreateEventHandler<(Of <(TEventArgs>)>)()()()</u>.

func

Type: Func<(Of <(EventArgs>)>) A function that will build the EventArgs to pass when raising the event.

IRaise Interface Raises Overload Moq.Language Namespace IRaise..::Raises(MockedEvent, EventArgs)

■ □ C# Moq IRaise..::.Raises<(Of <(T>)>) Method (MockedEvent, <(Of <(T, >)>)) <u>IRaise Interface</u> <u>See Also</u> <u>Send Feedback</u>

Specifies the mocked event that will be raised when the expectation is met.

Syntax

C#

<u>IVerifies</u> Raises<T>(<u>MockedEvent</u> eventHandler, Func<T, EventArgs> func)

Parameters

eventHandler

Type: <u>Moq..::.MockedEvent</u> The mocked event, retrieved from <u>CreateEventHandler()()()</u> or <u>CreateEventHandler<(Of <(TEventArgs>)>)()()()</u>.

func

Type: Func<(Of <(T, EventArgs>)>) A function that will build the EventArgs to pass when raising the event.

Type Parameters

Т

Type of the argument received by the expected invocation.

IRaise Interface Raises Overload Moq.Language Namespace IRaise..::Raises(MockedEvent, EventArgs)

■ □ 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.

Syntax

C#

```
<u>IVerifies</u> Raises<T1, T2>(
<u>MockedEvent</u> eventHandler,
Func<T1, T2, EventArgs> func
)
```

Parameters

eventHandler

Type: <u>Moq..::.MockedEvent</u> The mocked event, retrieved from <u>CreateEventHandler()()()</u> or <u>CreateEventHandler<(Of <(TEventArgs>)>)()()()</u>.

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.

IRaise Interface Raises Overload Moq.Language Namespace IRaise..::Raises(MockedEvent, EventArgs)

⊟ □ 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.

Syntax

C#

<u>IVerifies</u> Raises<T1, T2, T3>(<u>MockedEvent</u> eventHandler, Func<T1, T2, T3, EventArgs> func)

Parameters

eventHandler

Type: <u>Moq..::.MockedEvent</u> The mocked event, retrieved from <u>CreateEventHandler()()()</u> or <u>CreateEventHandler<(Of <(TEventArgs>)>)()()()</u>.

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.

Т3

Type of the third argument received by the expected invocation.

IRaise Interface Raises Overload Moq.Language Namespace IRaise..::Raises(MockedEvent, EventArgs)

□ 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.

Syntax

C#

Parameters

eventHandler

Type: <u>Moq..::.MockedEvent</u> The mocked event, retrieved from <u>CreateEventHandler()()()</u> or <u>CreateEventHandler<(Of <(TEventArgs>)>)()()()</u>.

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.

Т3

Type of the third argument received by the expected invocation.

T4

Type of the fourth argument received by the expected invocation.

IRaise Interface Raises Overload Moq.Language Namespace IRaise..::Raises(MockedEvent, EventArgs)

■
□ C#
Moq
IReturns Interface
See Also Send Feedback
Base interface for IReturns<(Of <(TResult>)>).

Syntax

C#

public interface IReturns

Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IReturns<(Of <(TResult>)>) Interface <u>Members</u> <u>See Also</u> <u>Send Feedback</u> 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.

<u>IReturns<(Of <(TResult>)>) Members</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IReturns<(Of <(TResult>)>) Members
 IReturns<(Of <(TResult>)>) Interface Methods See Also Send Feedback

Methods

Name Description

■ <u>Returns</u> Overloaded.

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IReturns<(Of <(TResult>)>) Methods
 IReturns<(Of <(TResult>)>) Interface See Also Send Feedback

Methods

Name Description

■ <u>Returns</u> Overloaded.

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Moq.Language Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IReturns<(Of <(TResult>)>)..::.Returns Method
 IReturns<(Of <(TResult>)>) Interface See Also Send Feedback

Overload List

Name Returns(Func<(Of = <(TResult>)>)) <u>Returns<(Of <(T>)></u>) (Func<(Of <(T, :0 TResult>)>)) Returns<(Of <(T1, T2>)>) <u>(Func<(Of <(T1, T2, </u> ŵ TResult>)>)) Returns<(Of <(T1, T2, <u>T3>)>)(Func<(Of <(T1,</u> :0 <u>T2, T3, TResult>)>))</u> Returns<(Of <(T1, T2, T3, <u>T4>)>)(Func<(Of <(T1, </u> Ŵ <u>T2, T3, T4, TResult>)>))</u> **Returns**(**TResult**) 0

Description

Specifies a function that will calculate the value to return from the method.

Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.

Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.

Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.

Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.

Specifies the value to return.

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

Send comments on this topic to moqdisc@googlegroups.com

□ C# Moq IReturns<(Of <(TResult>)>)..::.Returns Method (<(Of <(TResult>)>)) IReturns<(Of <(TResult>)>) Interface Example See Also Send Feedback 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 <(<u>TResult</u>>)>) The function that will calculate the return value.

Examples

Return a calculated value when the method is called:

C#

Copy

```
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.

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Returns Overload</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

IReturns<(Of <(TResult>)>)..::.Returns<(Of <(T>)>) Method (<(Of <(T, TResult>)>))

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Example</u> <u>See Also</u> <u>Send Feedback</u>

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<T>(
 Func<T, TResult> valueFunction
)

Parameters

valueFunction

Type: Func<(Of <(T, <u>TResult</u>>)>) The function that will calculate the return value.

Type Parameters

Т

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:

C#

Сору

```
mock.Expect(x => x.Execute(It.IsAny<string>()))
    .Returns((string command) => returnValues[command]);
```

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Returns Overload</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

IReturns<(Of <(TResult>)>)..::.Returns<(Of <(T1, T2>)>) Method (<(Of <(T1, T2, TResult>)>))

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Example</u> <u>See Also</u> <u>Send Feedback</u>

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>(
 Func<T1, T2, TResult> valueFunction
)

Parameters

valueFunction

Type: Func<(Of <(T1, T2, <u>TResult</u>>)>) 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#

Copy

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Returns Overload</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

IReturns<(Of <(TResult>)>)..::.Returns<(Of <(T1, T2, T3>)>) Method (<(Of <(T1, T2, T3, TResult>)>))

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Example</u> <u>See Also</u> <u>Send Feedback</u>

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, <u>TResult</u>>)>) 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.

Т3

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:

C#

```
Сору
```

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Returns Overload</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq

IReturns<(Of <(TResult>)>)..::.Returns<(Of <(T1, T2, T3, T4>)>) Method (<(Of <(T1, T2, T3, T4, TResult>)>))

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Example</u> <u>See Also</u> <u>Send Feedback</u>

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, <u>TResult</u>>)>) 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.

Т3

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:

C#

Сору

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Returns Overload</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C# Moq IReturns<(Of <(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#

IReturnsResult Returns(TResult value)

Parameters

value

Type: <u>TResult</u>

The value to return, or nullNothingnullptra null reference (Nothing in Visual Basic).

Examples

Return a true value from the method call:

C#

Сору

```
mock.Expect(x => x.Execute("ping"))
   .Returns(true);
```

<u>IReturns<(Of <(TResult>)>)</u> Interface <u>Returns Overload</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IReturnsGetter<(Of <(TProperty>)>) Interface <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

Defines the Returns verb for property get expectations.

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

Syntax

C#

public interface IReturnsGetter<TProperty> : IReturns

Type Parameters

TProperty

Type of the property.

<u>IReturnsGetter<(Of <(TProperty>)>) Members</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq

IReturnsGetter<(Of <(TProperty>)>) Members

<u>IReturnsGetter<(Of <(TProperty>)>)</u> Interface Methods See Also Send <u>Feedback</u>

Methods

Name Description

■ <u>Returns</u> Overloaded.

<u>IReturnsGetter<(Of <(TProperty>)>)</u> Interface <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IReturnsGetter<(Of <(TProperty>)>) Methods
 IReturnsGetter<(Of <(TProperty>)>) Interface See Also Send Feedback

Methods

Name Description

■ <u>Returns</u> Overloaded.

<u>IReturnsGetter<(Of <(TProperty>)>)</u> Interface <u>Moq.Language Namespace</u>

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

Name

- Returns(Func<(Of <(TProperty>)>))
- Returns(TProperty)

Description

Specifies a function that will calculate the value to return for the property. Specifies the value to return.

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: <u>Moq.Language</u> Assembly: Moq (in Moq.dll) Version: 2.6.1014.1 (2.6.0.0)

Syntax

C#

IReturnsResult Returns(
 Func<TProperty> valueFunction
)

Parameters

valueFunction

Type: Func<(Of <(<u>TProperty</u>>)>) The function that will calculate the return value.

Examples

Return a calculated value when the property is retrieved:

C#

Copy

```
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.

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#

Parameters

value

Type: <u>TProperty</u>

The value to return, or nullNothingnullptra null reference (Nothing in Visual Basic).

Examples

Return a true value from the property getter call:

C#

Сору

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

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

IThrows Members Moq.Language Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IThrows Members
 IThrows Interface Methods See Also Send Feedback

Methods

Name Description

■ <u>Throws</u> Overloaded.

IThrows Interface Moq.Language Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IThrows Methods
 IThrows Interface See Also Send Feedback

Methods

Name Description

■ <u>Throws</u> Overloaded.



<u>IThrows Interface</u> <u>Moq.Language Namespace</u>

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

Name

- Throws<(Of <(TException>)>)()()()
- Throws(Exception)

Description

Specifies the type of exception to throw when the method is invoked.

Specifies the exception to throw when the method is invoked.



<u>IThrows Interface</u> <u>IThrows Members</u> <u>Moq.Language Namespace</u>

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#

Сору

```
mock.Expect(x => x.Execute(""))
.Throws<ArgumentException>();
```

See Also

<u>IThrows Interface</u> <u>Throws Overload</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C#
 Moq
 IThrows..::.Throws Method ()
 IThrows Interface Example See Also Send Feedback
 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.

Examples

This example shows how to throw an exception when the method is invoked with an empty string argument:

C#

Сору

```
mock.Expect(x => x.Execute(""))
    .Throws(new ArgumentException());
```

See Also

<u>IThrows Interface</u> <u>Throws Overload</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

□ C#
Moq
IVerifies Interface
Members See Also Send Feedback
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

<u>IVerifies Members</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IVerifies Members
 IVerifies Interface Methods See Also Send Feedback

The <u>IVerifies</u> type exposes the following members.

Methods

=Q

NameDescriptionVerifiableMarks the expectation as verifiable, meaning that a call to
Verify()()) will check if this particular expectation was met.



<u>IVerifies Interface</u> <u>Moq.Language Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IVerifies Methods
 <u>IVerifies Interface</u> See Also Send Feedback

The <u>IVerifies</u> type exposes the following members.

Methods

=Q

NameDescriptionVerifiableMarks the expectation as verifiable, meaning that a call to
Verify()()) will check if this particular expectation was met.



<u>IVerifies Interface</u> <u>Moq.Language Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IVerifies..::.Verifiable Method <u>IVerifies Interface Example See Also Send Feedback</u>

Marks the expectation as verifiable, meaning that a call to <u>Verify()()()</u> 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#

Сору

```
mock.Expect(x => x.Execute("ping"))
   .Returns(true)
   .Verifiable();
```

See Also

<u>IVerifies Interface</u> <u>Moq.Language Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq Moq.Protected Namespace <u>Send Feedback</u>

Importing this namespace will enable the Protected() method on mocks, enabling expectations on protected members by specifying the member name as a string.

Classes

	Class	Description
4 3	<u>ItExpr</u>	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.
43	<u>ProtectedExtension</u>	Enables the Protected() method on $\underline{Mock}\leq(Of \leq (T \geq) \geq)$, 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.

Interfaces

	Interface	Description
~0	<u>IProtectedMock</u>	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.

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq IProtectedMock Interface <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

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



IProtectedMock Members Moq.Protected Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IProtectedMock Members
 IProtectedMock Interface Methods See Also Send Feedback

The <u>IProtectedMock</u> type exposes the following members.

Methods

Name

Description

• <u>Expect</u>

=Q

Overloaded.

■ <u>ExpectGet<(Of</u> <u><(TProperty>)></u>)

- ExpectSet<(Of
 - <u><(TProperty>)>)</u>

Sets an expectation on a property getter with the given propertyName.

Sets an expectation on a property setter with the given propertyName.



IProtectedMock Interface Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 IProtectedMock Methods
 IProtectedMock Interface See Also Send Feedback

The <u>IProtectedMock</u> type exposes the following members.

Methods

Name

Description

• <u>Expect</u>

=Q

Overloaded.

■ <u>ExpectGet<(Of</u> <u><(TProperty>)></u>)

- ExpectSet<(Of
 - <u><(TProperty>)>)</u>

Sets an expectation on a property getter with the given propertyName.

Sets an expectation on a property setter with the given propertyName.



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

Overload List

Name

Expect<(Of <(TResult>)>)(String, array<Object>[]()[])

Expect(String, array<Object>[]()[])

Description

Sets an expectation on a property or a non void method with the given methodOrPropertyName, optionally specifying arguments for the method call.

Sets an expectation on the void method with the given voidMethodName, optionally specifying arguments for the method call.

See Also

IProtectedMock Interface IProtectedMock Members Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

□ C# Moq IProtectedMock..::.Expect<(Of <(TResult>)>) Method (, array<>[]()[]) IProtectedMock Interface See Also Send Feedback

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 <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IProtectedMock..::.Expect Method (, array<>[]()[]) <u>IProtectedMock Interface</u> <u>See Also</u> <u>Send Feedback</u>

Sets an expectation on the void method with the given voidMethodName, 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 Expect(
 string voidMethodName,
 params Object[] args
)

Parameters

voidMethodName Type: String Name of the void method to be invoke.

args

Type: array< Object >[]()[] Optional arguments for the invocation.

See Also

IProtectedMock Interface Expect Overload Moq.Protected Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IProtectedMock..::.ExpectGet<(Of <(TProperty>)>) Method <u>IProtectedMock Interface</u> <u>See Also</u> <u>Send Feedback</u>

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#

Parameters

propertyName Type: String Name of the property.

Type Parameters

TProperty

Type of the property.



IProtectedMock Interface Moq.Protected Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq IProtectedMock..::.ExpectSet<(Of <(TProperty>)>) Method <u>IProtectedMock Interface</u> <u>See Also</u> <u>Send Feedback</u>

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#

Parameters

propertyName Type: String Name of the property.

Type Parameters

TProperty

Type of the property.



IProtectedMock Interface Moq.Protected Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq ItExpr Class <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

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: <u>Moq.Protected</u> 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 <u>It</u> 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



ItExpr Members Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ItExpr Members
 ItExpr Class Methods See Also Send Feedback

The <u>ItExpr</u> type exposes the following members.

Methods

	Name	Description
≓≬S	<u> Is<(Of <(TValue>)>)</u>	Matches any value that satisfies the given predicate.
≡ ≬ S	<pre>IsAny<(Of <(TValue>)>)</pre>	Matches any value of the given TValue type.
≓≬S	<u>IsInRange<(Of</u> < <u>(TValue>)></u>)	Matches any value that is in the range specified.
= \$	<u>IsRegex</u>	Overloaded.



ItExpr Class Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ItExpr Methods
 ItExpr Class See Also Send Feedback

The <u>ItExpr</u> type exposes the following members.

Methods

	Name	Description
≓≬S	<u> Is<(Of <(TValue>)>)</u>	Matches any value that satisfies the given predicate.
≡ ≬ S	<pre>IsAny<(Of <(TValue>)>)</pre>	Matches any value of the given TValue type.
≓≬S	<u>IsInRange<(Of</u> < <u>(TValue>)></u>)	Matches any value that is in the range specified.
= \$	<u>IsRegex</u>	Overloaded.



ItExpr Class Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 ItExpr..::.Is<(Of <(TValue>)>) Method
 ItExpr Class Example See Also Send Feedback
 Matches any value that satisfies the given predicate.

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

Syntax

C#

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

C#

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

```
C# Copy
```

```
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:

Copy

```
mock.Protected()
    .Expect("GetUser", ItExpr.Is<int>(i => i < 0))
    .Throws(new ArgumentException());</pre>
```



ItExpr Class Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

□ C#
 Moq
 ItExpr..::.IsAny<(Of <(TValue>)>) Method
 ItExpr Class Example See Also Send Feedback
 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#

Сору

// Throws an exception for a call to Remove with any string value.
mock.Protected()

.Expect("Remove", ItExpr.IsAny<string>())
.Throws(new InvalidOperationException());



ItExpr Class Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq ItExpr..::.IsInRange<(Of <(TValue>)>) 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#

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: <u>Moq..::Range</u> The kind of range. See <u>Range</u>.

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#

Copy

```
mock.Protected()
    .Expect("HasInventory",
        ItExpr.IsAny<string>(),
        ItExpr.IsInRange(0, 100, Range.Inclusive))
    .Returns(false);
```



ItExpr Class Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ItExpr..::.IsRegex Method
 ItExpr Class See Also Send Feedback

Overload List

Name

Description

■ IsRegex(String) IsRegex(String, Regex(Dptions) Matches a string argument if it matches the given regular expression pattern.

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



<u>ItExpr Class</u> <u>ItExpr Members</u> <u>Moq.Protected Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq ItExpr.::.IsRegex Method () <u>ItExpr Class</u> Example See Also Send Feedback

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#

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#	Сору
----	------

```
mock.Protected()
    .Expect("Check", ItExpr.IsRegex("[a-z]+"))
    .Returns(1);
```

See Also

<u>ItExpr Class</u> <u>IsRegex Overload</u> <u>Moq.Protected Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq ItExpr.::.IsRegex Method (,) <u>ItExpr Class</u> Example See Also Send Feedback

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#

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:

```
mock.Protected()
    .Expect("Check", ItExpr.IsRegex("[a-z]+", RegexOptions.IgnoreCas
    .Returns(1);
```

See Also

<u>ItExpr Class</u> <u>IsRegex Overload</u> <u>Moq.Protected Namespace</u>

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq ProtectedExtension Class <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

Enables the Protected() method on $\underline{Mock} \leq (\underline{Of} \leq (\underline{T} \geq) \geq)$, 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.

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

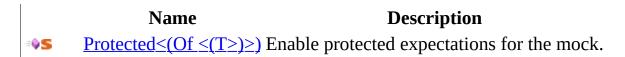
ProtectedExtension Members Moq.Protected Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ProtectedExtension Members
 ProtectedExtension Class Methods See Also Send Feedback

The <u>ProtectedExtension</u> type exposes the following members.

Methods





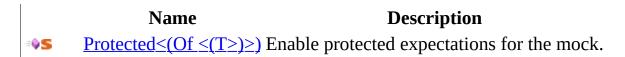
ProtectedExtension Class Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 ProtectedExtension Methods
 ProtectedExtension Class See Also Send Feedback

The <u>ProtectedExtension</u> type exposes the following members.

Methods





ProtectedExtension Class Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com

■ □ C# Moq ProtectedExtension..::.Protected<(Of <(T>)>) Method <u>ProtectedExtension Class</u> <u>See Also</u> <u>Send Feedback</u> 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#

Parameters

mock

Type: <u>Moq..::.Mock</u><(Of <(T>)>) The mock to set the protected expectations on.

Type Parameters

Т

Mocked object type. Typically omitted as it can be inferred from the mock instance.



ProtectedExtension Class Moq.Protected Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq Moq.Stub Namespace <u>Send Feedback</u>

[Missing <summary> documentation for N:Moq.Stub]

Classes

Class	Description
STUDE VIANCIANC	Adds Stub extension method to a mock so that you can stub properties.

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq StubExtensions Class <u>Members</u> <u>See Also</u> <u>Send Feedback</u>

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 <u>moqdisc@googlegroups.com</u>

C#
 Include Protected Members
 Include Inherited Members
 Moq
 StubExtensions Members
 StubExtensions Class Methods See Also Send Feedback

The <u>StubExtensions</u> type exposes the following members.

Methods

	Name	Description
≡≬S	<u>Stub</u>	Overloaded.
≓∳S	<u>StubAll≤(Of</u> ≤(T≥)≥)	Stubs all properties on the mock, setting the default value to the one generated as specified by the <u>DefaultValue</u> property.



StubExtensions Class Moq.Stub Namespace

Send comments on this topic to moqdisc@googlegroups.com

C#
 Include Protected Members
 Include Inherited Members
 Moq
 StubExtensions Methods
 StubExtensions Class See Also Send Feedback

The <u>StubExtensions</u> type exposes the following members.

Methods

	Name	Description
≡≬S	<u>Stub</u>	Overloaded.
≓∳S	<u>StubAll≤(Of</u> ≤(T≥)≥)	Stubs all properties on the mock, setting the default value to the one generated as specified by the <u>DefaultValue</u> property.



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

Overload List

Name	Description
Stub<(Of <(T, TProperty>)>)(Mock<(Of	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.
	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.

See Also

<u>StubExtensions Class</u> <u>StubExtensions Members</u> <u>Moq.Stub Namespace</u>

Send comments on this topic to moqdisc@googlegroups.com

-

□ C#

Moq

StubExtensions..::.Stub<(Of <(T, TProperty>)>) Method (Mock<(Of <(T>)>), <(Of <(<(Of <(T, TProperty>)>)>))) StubExtensions Class Example See Also Send Feedback

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)

Syntax

C#

Parameters

mock

Type: <u>Moq..::.Mock</u><(Of <(T>)>) The instance to stub.

property

```
Type: Expression<(Of <(Func<(Of <(T, TProperty>)>)>)>)
Property expression to stub.
```

Type Parameters

Т

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# Copy

```
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#

```
Сору
```

```
IHaveValue v = mock.Object;
```

```
v.Value = 5;
Assert.Equal(5, v.Value);
```



StubExtensions Class Stub Overload Moq.Stub Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

-

□ C#

Moq StubExtensions..::.Stub<(Of <(T, TProperty>)>) Method (Mock<(Of <(T>)>), <(Of <(<(Of <(T, TProperty>)>)>)), TProperty) <u>StubExtensions Class Example See Also Send Feedback</u>

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#

Parameters

mock

```
Type: <u>Moq..::.Mock</u><(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

Т

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:

Сору

```
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:

C#

C#

Copy

```
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);
```



StubExtensions Class Stub Overload Moq.Stub Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>

■ □ C# Moq StubExtensions..::.StubAll<(Of <(T>)>) Method <u>StubExtensions Class</u> <u>See Also</u> <u>Send Feedback</u>

Stubs all properties on the mock, setting the default value to the one generated as specified by the <u>DefaultValue</u> property.

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

Syntax

C#

Parameters

mock

Type: <u>Moq..::.Mock</u><(Of <(T>)>) The mock to stub.

Type Parameters

Т

Mocked type, typically omitted as it can be inferred from the mock argument.

Remarks

If the mock <u>DefaultValue</u> is set to <u>Mock</u>, the mocked default values will also be stubbed recursively.



StubExtensions Class Moq.Stub Namespace

Send comments on this topic to <u>moqdisc@googlegroups.com</u>