□ C#
Moq
Moq Namespace
Send Feedback

Root namespace of Moq (pronounced "Mock-you" or just "Mock"), the only mocking library for .NET and Silverlight 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
9 3	<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.
4 3	<u>Mock</u>	Base class for mocks and static helper class with methods that apply to mocked objects, such as $Get < (Of <<'(T>)>>)(T)$ to retrieve a $Mock < (Of <<('T>)>>)>$ from an object instance.
9 \$	<u>Mock<(Of</u> <(<'T>)>)>	Provides a mock implementation of T.
43	MockException	Exception thrown by mocks when setups are not matched, the mock is not properly setup, etc.
4 3	MockExtensions	Provides additional methods on mocks.
* \$	MockFactory	Obsolete. Utility factory class to use to construct multiple mocks when consistent verification is desired for all of them.
? (\$	MockRepository	Utility repository class to use to construct multiple mocks when consistent verification is desired for all of them.
? \$	<u>Mocks</u>	Allows querying the universe of mocks for those that behave according to the LINQ query specification.
43	<u>MockSequence</u>	Helper class to setup a full trace between many mocks
4 \$	<u>MockSequenceHelpe</u>	r define nice api
43	SequenceExtensions	Helper for sequencing return values in the same method.

Structures

Structure

Description



Times

Defines the number of invocations allowed by a mocked method.

Enumerations

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

-

□ C#

Moq

DefaultValue Enumeration

See Also Send Feedback

Determines the way default values are generated calculated for loose mocks.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public enum DefaultValue

Members

Member name Description

Empty

Mock

Moq Namespace

-

□ C#

Moq

It Class

Members See Also Send Feedback

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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public static class It

Remarks

This class allows the setup 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

System..:..Object
Moq..:..It

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

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

The Lt type exposes the following members.

Methods

	Name	Description
= \$ S	<u>Is<(Of</u> <<'(TValue>)>>)	
∉ ≬S	<u>IsAny<(Of</u> <<' <u>(TValue>)>>)</u>	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.
= \$ S	IsRegex(String)	Matches a string argument if it matches the given regular expression pattern.
≡ ≬S	<pre>IsRegex(String, RegexOptions)</pre>	Matches a string argument if it matches the given regular expression pattern.

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

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

The Lt type exposes the following members.

Methods

	Name	Description
= \$ S	<u>Is<(Of</u> <<'(TValue>)>>)	
∉ ≬S	<u>IsAny<(Of</u> <<' <u>(TValue>)>>)</u>	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.
= \$ S	IsRegex(String)	Matches a string argument if it matches the given regular expression pattern.
≡ ≬S	<pre>IsRegex(String, RegexOptions)</pre>	Matches a string argument if it matches the given regular expression pattern.

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

```
☐ C#
Moq
It..::..Is<(Of <(<'TValue>)>)> Method
It Class See Also Send Feedback
```

[Missing <summary> documentation for "M:Moq.It.Is``1(System.Linq.Expressions.Expression{System.Func`2})"]

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

match

```
Type: System.Linq.Expressions..:..Expression<(Of <(<'Func<(Of <(<'T, TResult>)>)>)>)>
```

Type Parameters

TValue

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

_

□ C#

Moq

It......Is Any < (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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public static TValue IsAny<TValue>()

Type Parameters

TValue

Type of the value.

Remarks

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

Examples

C#

// Throws an exception for a call to Remove with any string value. $mock.Setup(x \Rightarrow x.Remove(It.IsAny<string>())).Throws(new InvalidOper)$

<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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

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.Setup(x => x.HasInventory(
It.IsAny<string>(),
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
- <u>It Class See Also Send Feedback</u>

Overload List

	Name	Description
= QS	IsRegex(String)	Matches a string argument if it matches the given regular expression pattern.
= QS	IsRegex(String, RegexOptions)	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 (String)

It Class Example See Also Send Feedback

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

regex

Type: <u>System.....String</u>

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# Copy

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

It Class IsRegex Overload Moq Namespace

-

□ C#

Moq

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

It Class Example See Also Send Feedback

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

regex

Type: System..:..String

The pattern to use to match the string argument value.

options

Type: <u>System.Text.RegularExpressions..:..RegexOptions</u>

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.Setup(x => x.Check(It.IsRegex("[a-z]+", RegexOptions.IgnoreCase)))

It Class IsRegex Overload Moq Namespace

-

□ C#

Moq

Mock Class

Members See Also Send Feedback

Base class for mocks and static helper class with methods that apply to mocked objects, such as $\underline{\text{Get} < (\text{Of} <<'(\text{T}\geq)\geq>)(\text{T})}$ to retrieve a $\underline{\text{Mock} < (\text{Of} <(<'\text{T}\geq)\geq)\geq}$ from an object instance.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public abstract class Mock : IHideObjectMembers

Inheritance Hierarchy

System..:..Object Moq..::..Mock

 $\underline{Moq.....Mock} \le (\underline{Of} \le (\le '\underline{T} \ge) \ge) \ge$

Mock Members Moq Namespace

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

The Mock type exposes the following members.

Constructors

Name

Description

·

Mock Initializes a new instance of the Mock class.

Methods

		Name	Description
=(• <u>A</u>	us<(Of <<'(TInterface>)>>)	Adds an interface implementation to the mock, allowing setups to be specified for it.
=1	•s <u>G</u>	<u>set<(Of <<'(T>)>>)</u>	Retrieves the mock object for the given object instance.
=1	§ S <u>O</u>	<u>of<(Of <<'(T>)>>)()()()()</u>	Creates an mock object of the indicated type.
		<u>of<(Of <<'(T>)>>)</u>	
=(•	Expression<(Of <<'(Func<(Of (<'T, TResult>)>)>>)>))	
8	<u>O</u>	<u>nGetObject</u>	Returns the mocked object value.
=1		<u>etReturnsDefault<(Of</u> <' <u>(TReturn>)>>)</u>	
=1	<u>V</u>	<u>'erify</u>	Verifies that all verifiable expectations have been met.
=(<u>V</u>	<u>'erifyAll</u>	Verifies all expectations regardless of whether they have been flagged as verifiable.

Properties

Name	Description
<u>Behavior</u>	Behavior of the mock, according to the value set in the constructor.
	Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalsefalse (False in Visual Basic).
<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks.
<u>Object</u>	Gets the mocked object instance.

Mock Class Moq Namespace

-

□ C#

Moq

Mock Constructor

Mock Class See Also Send Feedback

Initializes a new instance of the Mock class.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

protected Mock()

Mock Class Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Mock Methods

Mock Class See Also Send Feedback

The Mock type exposes the following members.

Methods

		Name	Description
=(• <u>A</u>	us<(Of <<'(TInterface>)>>)	Adds an interface implementation to the mock, allowing setups to be specified for it.
=1	•s <u>G</u>	<u>set<(Of <<'(T>)>>)</u>	Retrieves the mock object for the given object instance.
=1	§ S <u>O</u>	<u>of<(Of <<'(T>)>>)()()()()</u>	Creates an mock object of the indicated type.
		<u>of<(Of <<'(T>)>>)</u>	
=(•	Expression<(Of <<'(Func<(Of (<'T, TResult>)>)>>)>))	
8	<u>O</u>	<u>nGetObject</u>	Returns the mocked object value.
=1		<u>etReturnsDefault<(Of</u> <' <u>(TReturn>)>>)</u>	
=1	<u>V</u>	<u>'erify</u>	Verifies that all verifiable expectations have been met.
=(<u>V</u>	<u>'erifyAll</u>	Verifies all expectations regardless of whether they have been flagged as verifiable.

Mock Class Moq Namespace

-

□ C#

Moq

Mock..:..As<(Of <(<'TInterface>)>)> Method

Mock Class Example See Also Send Feedback

Adds an interface implementation to the mock, allowing setups to be specified for it.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public virtual $\underline{\mathsf{Mock}} < \mathsf{TInterface} > \mathsf{As} < \mathsf{TInterface} > ()$ where $\mathsf{TInterface} : \mathsf{class}$

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.Setup(x => x.Execute("ping"));

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

Exceptions

	Exception	Condition
		The mock type has already been
	System:InvalidOperationException	generated by accessing the Object
		property.
	System ArgumentEvantion	The TInterface specified is not an
3	System:ArgumentException	interface.

Mock Class Moq Namespace

-

□ 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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

Parameters

mocked

Type: T

The instance of the mocked object.

Type Parameters

T

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

Return Value

The mock associated with the mocked object.

Examples

The following example shows how to add a new setup to an object instance which is not the original $\underline{Mock} < (\underline{Of} < (\underline{<'T} >) >) >$ but rather the object associated with it:

C# □ Copy

Exceptions

Exception

Condition

System..:..ArgumentException by Moq.

The received mocked instance was not created by Mog.

Mock Class Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Mock..:..Of Method

Mock Class See Also Send Feedback

Overload List

Name

- $\underline{Of} \leq (\underline{Of} \leq \leq'(\underline{T} \geq) \geq)()()()()$ **=\$**S
- <u>Of<(Of <<'(T>)>>)(Expression<(Of <<''(Func<(Of <(<'T, TResult>)>)>>))</u>

Description

Creates an mock object of the indicated type.

Mock Class Mock Members Moq Namespace

-

□ C#

Moq

Mock..:..Of<(Of <(<'T>)>)> Method

Mock Class See Also Send Feedback

Creates an mock object of the indicated type.

Namespace: Moq

\blacksquare Syntax

C#

public static T Of<T>()
where T : class

■Type Parameters

T

The type of the mocked object.

Return Value

The mocked object created.

Mock Class Of Overload Moq Namespace

```
☐ C#

Moq

Mock..:..Of<(Of <(<'T>)>)> Method (Expression<(Of <(<'Func>)>)>)

Mock Class See Also Send Feedback

[Missing <summary> documentation for
"M:Moq.Mock.Of`1(System.Linq.Expressions.Expression{System.Func`2})"]
```

Namespace: Moq

Syntax

Parameters

```
predicate
```

Type Parameters

T

Mock Class Of Overload Moq Namespace

-

□ C#

Moq

Mock..:..OnGetObject Method

Mock Class See Also Send Feedback

Returns the mocked object value.

Namespace: Moq

$\blacksquare Syntax$

C#

protected abstract Object OnGetObject()

Mock Class Moq Namespace

```
-
```

□ C#

Moq

Mock..:..SetReturnsDefault<(Of <(<'TReturn>)>)> Method

Mock Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Mock.SetReturnsDefault``1(``0)"]

Namespace: Moq

\blacksquare Syntax

C#

Parameters

value

Type: TReturn

Type Parameters

TReturn

Mock Class Moq Namespace

-

□ C#

Moq

Mock..:..Verify Method

Mock Class Example See Also Send Feedback

Verifies that all verifiable expectations have been met.

Namespace: Moq

$\blacksquare Syntax$

C#

public 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 setup was invoked:

C# Copy

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

Exceptions

Exception

Condition

Moq..:..MockException

Not all verifiable expectations were met.

Mock Class Moq Namespace

_

□ C#

Moq

Mock..:..VerifyAll Method

Mock Class Example See Also Send Feedback

Verifies all expectations regardless of whether they have been flagged as verifiable.

Namespace: Moq

\blacksquare Syntax

C#

public 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>();
this.Setup(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.
this.VerifyAll();
```

Exceptions

Exception

Condition

Moq..:..MockException

At least one expectation was not met.

Mock Class Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Mock Properties

Mock Class See Also Send Feedback

The Mock type exposes the following members.

Properties

Name	Description
<u>Behavior</u>	Behavior of the mock, according to the value set in the constructor.
	Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalsefalse (False in Visual Basic).
<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks.
<u>Object</u>	Gets the mocked object instance.

Mock Class Moq Namespace

_

□ C#

Moq

Mock..:..Behavior Property

Mock Class See Also Send Feedback

Behavior of the mock, according to the value set in the constructor.

Namespace: Moq

Syntax

C#

```
public virtual MockBehavior Behavior { get; internal set; }
```

Mock Class Moq Namespace

-

□ C#

Moq

Mock..:..CallBase Property

Mock Class See Also Send Feedback

Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalsefalse (False in Visual Basic).

Namespace: Moq

\blacksquare Syntax

C#

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

Mock Class Moq Namespace

-

□ C#

Moq

Mock..:..DefaultValue Property

Mock Class See Also Send Feedback

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public virtual DefaultValue DefaultValue { get; set; }

Mock Class Moq Namespace

-

□ C#

Moq

Mock..:..Object Property

Mock Class See Also Send Feedback

Gets the mocked object instance.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

```
public Object Object { get; }
```

Mock Class Moq Namespace

-

□ C#

Moq

Mock<(Of <(<'T>)>)> Class

Members Example See Also Send Feedback

Provides a mock implementation of T.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public class Mock<T> : $\underline{\mathsf{Mock}}$ where T : class

Type Parameters

T

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

Remarks

Any interface type can be used for mocking, but for classes, only abstract and virtual members can be mocked.

The behavior of the mock with regards to the setups and the actual calls is determined by the optional $\underline{\text{MockBehavior}}$ that can be passed to the $\underline{\text{Mock}} \leq (\leq T \geq) \geq (\underline{\text{MockBehavior}})$ constructor.

Examples

The following example shows establishing setups with specific values for method invocations:

C# Copy

```
// Arrange
var order = new Order(TALISKER, 50);
var mock = new Mock<IWarehouse>();
mock.Setup(x => x.HasInventory(TALISKER, 50)).Returns(true);
// Act
order.Fill(mock.Object);
// Assert
Assert.True(order.IsFilled);
```

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

C# Copy

```
// Arrange
var order = new Order(TALISKER, 50);
var mock = new Mock<IWarehouse>();
// shows how to expect a value within a range
mock.Setup(x => x.HasInventory(
            It.IsAny<string>(),
            It.IsInRange(0, 100, Range.Inclusive)))
     .Returns(false);
// shows how to throw for unexpected calls.
mock.Setup(x => x.Remove(
            It.IsAny<string>(),
            It.IsAny<int>()))
     .Throws(new InvalidOperationException());
// Act
order.Fill(mock.Object);
// Assert
```

Assert.False(order.IsFilled);

Inheritance Hierarchy

System..:..Object

Moq..:..Mock Moq..:..Mock<(Of <(<'T>)>)>

Mock<(Of <(<'T>)>)> Members Moq Namespace

□ C#
□ Include Protected Members
□ Include Inherited Members

Moq

Mock<(Of <(<'T>)>)> Members

Mock<(Of <(<'T>)>)> Class Constructors Methods Properties See Also Send Feedback

The $\underline{\text{Mock}} \leq (\underline{\text{Of}} \leq (\underline{\text{C'T}}) \geq \underline{\text{C'T}} \geq \underline{\text{Mock}} \geq \underline{\text{Mock}} \leq \underline{\text{Mock}} \leq$

Constructors

	Name	Description
=0	<u>Mock<(Of <(<'T>)>)>()()()</u>	Initializes an instance of the mock with <u>default behavior</u> .
∄ ∳	<u>Mock<(Of <(<'T>)>)></u> (array <object>[]()[][])</object>	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)
=	Mock<(Of <(<'T>)>)> (MockBehavior)	Initializes an instance of the mock with the specified <u>behavior</u> .
≓ ∳	Mock<(Of <(<'T>)>)> (MockBehavior, array <object>[] ()[][])</object>	Initializes an instance of the mock with a specific behavior with the given constructor arguments for the class.

\blacksquare Methods

	Name	Description
=	<u>As<(Of <<'(TInterface>)>>)</u>	Adds an interface implementation to the mock, allowing setups to be specified for it. (Inherited from Mock.)
<u> </u>	<u>OnGetObject</u>	Returns the mocked object value. (Overrides <a "stubbing").="" (this="" <a="" also="" and="" as="" be="" behavior",="" by="" cause="" default="" each="" for="" generated="" href="DefaultValue" is="" it="" its="" known="" later="" meaning="" one="" property="" requested.="" returned="" saved="" setting="" specified="" that="" the="" to="" value="" when="" will="">DefaultValue property for the mock.

```
SetupGet<(Of <<'(TProperty>)>>)
SetupProperty<(Of
<<'(TProperty>)>>)(Expression<(Of
<<'(Func<(Of <(<'T,
TResult>)>)>>)>>))
SetupProperty<(Of
<<'(TProperty>)>>)(Expression<(Of
<<'(Func<(Of <(<'T,
<u>TResult>)>)>>), Boolean)</u>
                                      Specifies a setup on the mocked
                                      type for a call to to a property
\underline{SetupSet(Action < (Of <<'(T>)>>))}
                                      setter.
                                      Specifies a setup on the mocked
SetupSet<(Of <<'(TProperty>)>>)
                                     type for a call to to a property
(Action < (Of << '(T>)>>))
                                      setter.
                                      Verifies that all verifiable
                                      expectations have been met.
<u>Verify()()()</u>
                                      (Inherited from Mock.)
                                      Verifies that a specific invocation
                                      matching the given expression
<u>Verify(Expression<(Of</u>
                                     was performed on the mock. Use
<<'(Action<(Of <<'(T>)>>)>)>))
                                     in conjuntion with the default
                                      Loose.
                                      Verifies that a specific invocation
<u>Verify(Expression<(Of</u>
                                      matching the given expression
<<'(Action<(Of <<'(T>)>>)>),
                                      was performed on the mock. Use
Times)
                                     in conjuntion with the default
                                      Loose.
                                      Verifies that a specific invocation
                                     matching the given expression
Verify(Expression<(Of</pre>
                                     was performed on the mock,
<<'(Action<(Of <<'(T>)>>)>)>),
                                      specifying a failure error message.
String)
                                      Use in conjuntion with the default
                                      Loose.
                                      Verifies that a specific invocation
                                      matching the given expression
Verify(Expression<(Of</pre>
```

 $\leq \leq (Action \leq (Of \leq \leq (T \geq) \geq) \geq),$ Times, String) was performed on the mock, specifying a failure error message. Use in conjuntion with the default Loose.

<u>Verify<(Of <<'(TResult>)>>)</u>

(Expression<(Of <<'(Func<(Of <<('T, TResult>)>)>>)>>))

Verify<(Of <<'(TResult>)>>)

(Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>), Boolean)

Verify<(Of <<'(TResult>)>>)

(Expression<(Of <<'(Func<(Of <<(<'T, TResult>)>)>>), Boolean, T)

<u>VerifyAll</u>

Verifies all expectations regardless of whether they have been flagged as verifiable. (Inherited from Mock.)

<u>VerifyGet<(Of <<'(TProperty>)>>)</u>
(Expression<(Of <<'(Func<(Of

(Expression<(Of <<'(Func<(Of

<(<'T, TResult>)>)>>)>))

VerifyGet<(Of <<'(TProperty>)>>)

(Expression≤(Of ≤≤'(Func≤(Of

<(<'T, TResult>)>)>>), Boolean)

 $\underline{VerifyGet} \leq (\underline{Of} \leq \leq '(\underline{TProperty} \geq) \geq \geq)$

(Expression<(Of <<'(Func<(Of <<(<'T, TResult>)>)>>), Boolean, T)

 \vee <u>VerifySet(Action<(Of <<'(T>)>>))</u>

VerifySet(Action<(Of <<'(T>)>>),
Times)

 $\frac{\text{VerifySet(Action} < (\text{Of} < <'(\text{T}>) >>),}{\text{String})}$

Verifies that a property was set on the mock.

Verifies that a property was set on the mock.

Verifies that a property was set on the mock, specifying a failure message.

Verifies that a property was set on

- $\frac{\text{VerifySet(Action} < (\text{Of} < < '(\text{T} >) > >),}{\text{Times, String)}} \text{ the mock, specifying a failure message.}$
- **♦** When

Extension Methods

	Name	Description
₩	InSequence<(Of <<'(TMock>)>>)	Perform an expectation in the trace. (Defined by MockSequenceHelper.)
₩	<u>Protected<(Of <<'(T>)>>)</u>	Enable protected setups for the mock. (Defined by ProtectedExtension .)
•	SetupSequence<(Of <<'(TMock, TResult>)>>)	(Defined by SequenceExtensions .)
•	<pre>SetupSet<(Of <<'(T, TProperty>)>>)</pre>	(Defined by MockExtensions.)
₽ į	VerifySet<(Of <<'(T, TProperty>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>)))	Overloaded. (Defined by MockExtensions.)
Q į	VerifySet<(Of <<'(T, TProperty>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>), Boolean)	Overloaded. (Defined by MockExtensions.)
∳ įį	VerifySet<(Of <<'(T, TProperty>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>), Boolean)	Overloaded. (Defined by MockExtensions.)
•	VerifySet<(Of <<'(T, TProperty>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>), Boolean, T)	Overloaded. (Defined by MockExtensions.)

\blacksquare Properties

Name	Description
<u>Behavior</u>	Behavior of the mock, according to the value set in the constructor. (Inherited from Mock.)
<u>CallBase</u>	Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalsefalse (False in Visual Basic). (Inherited from Mock.)
<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks. (Inherited from Mock.)
<u>Object</u>	Exposes the mocked object instance.

Mock<(Of <(<'T>)>)> Class Moq Namespace

_

□ C#

- ☐ Include Protected Members
- □ Include Inherited Members

Moq

Mock < (Of < (<'T>)>)> Constructor

 $\underline{Mock} \le (\underline{Of} \le (\underline{\le'T} \ge) \ge) \ge Class$ See Also Send Feedback

Overload List

	Name	Description
≡	Mock<(Of <(<'T>)>)>()()()	Initializes an instance of the mock with <u>default behavior</u> .
≅∳	<u>Mock<(Of <(<'T>)>)></u> (<u>array<object>[]()[][]</object></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)
≡	Mock<(Of <(<'T>)>)> (MockBehavior)	Initializes an instance of the mock with the specified <u>behavior</u> .
∃	Mock<(Of <(<'T>)>)> (MockBehavior, array <object>[] ()[][])</object>	Initializes an instance of the mock with a specific behavior with the given constructor arguments for the class.

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Members Moq Namespace

_

□ C#

Moq

Mock<(Of <(<'T>)>)> Constructor

<u>Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback</u>

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public Mock()

Examples

C#

var mock = new Mock<IFormatProvider>();

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Overload Moq Namespace

-

□ C#

Moq

Mock<(Of <(<'T>)>)> Constructor (array<Object>[]()[][])

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: 4.0.10827.0 (4.0.0.0)

Syntax

Parameters

args

Type: array<<u>System..:..Object</u>>[]()[][]
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#

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

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Overload Moq Namespace

_

□ C#

Moq

Mock<(Of <(<'T>)>)> Constructor (MockBehavior)

<u>Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback</u>

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

```
)
```

Parameters

behavior

Type: Moq..:..MockBehavior Behavior of the mock.

Examples

C#

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

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Overload Moq Namespace

-

□ C#

Moq

Mock<(Of <(<'T>)>)> Constructor (MockBehavior, array<Object>[]()[][])

Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback

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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

behavior

Type: Moq....MockBehavior

Behavior of the mock.

args

Type: array<<u>System..:..Object</u>>[]()[][]

Optional constructor arguments if the mocked type is a class.

Remarks

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

Examples

C#

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

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Overload Moq Namespace

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq

Mock<(Of <(<'T>)>)> Methods

Mock<(Of <(<'T>)>)> Class See Also Send Feedback
```

The $\underline{Mock} \le (\underline{Of} \le (\underline{\le'T} \ge) \ge)$ type exposes the following members.

\blacksquare Methods

	Name	Description
=	<u>As<(Of <<'(TInterface>)>>)</u>	Adds an interface implementation to the mock, allowing setups to be specified for it. (Inherited from Mock.)
<u> </u>	<u>OnGetObject</u>	Returns the mocked object value. (Overrides <a "stubbing").="" (this="" <a="" also="" and="" as="" be="" behavior",="" by="" cause="" default="" each="" for="" generated="" href="DefaultValue" is="" it="" its="" known="" later="" meaning="" one="" property="" requested.="" returned="" saved="" setting="" specified="" that="" the="" to="" value="" when="" will="">DefaultValue property for the mock.

```
SetupGet<(Of <<'(TProperty>)>>)
SetupProperty<(Of
<<'(TProperty>)>>)(Expression<(Of
<<'(Func<(Of <(<'T,
TResult>)>)>>)>>))
SetupProperty<(Of
<<'(TProperty>)>>)(Expression<(Of
<<'(Func<(Of <(<'T,
<u>TResult>)>)>>), Boolean)</u>
                                      Specifies a setup on the mocked
                                      type for a call to to a property
\underline{SetupSet(Action < (Of <<'(T>)>>))}
                                      setter.
                                      Specifies a setup on the mocked
SetupSet<(Of <<'(TProperty>)>>)
                                     type for a call to to a property
(Action < (Of << '(T>)>>))
                                      setter.
                                      Verifies that all verifiable
                                      expectations have been met.
<u>Verify()()()</u>
                                      (Inherited from Mock.)
                                      Verifies that a specific invocation
                                      matching the given expression
<u>Verify(Expression<(Of</u>
                                     was performed on the mock. Use
<<'(Action<(Of <<'(T>)>>)>)>))
                                     in conjuntion with the default
                                      Loose.
                                      Verifies that a specific invocation
<u>Verify(Expression<(Of</u>
                                      matching the given expression
<<'(Action<(Of <<'(T>)>>)>),
                                      was performed on the mock. Use
Times)
                                     in conjuntion with the default
                                      Loose.
                                      Verifies that a specific invocation
                                     matching the given expression
Verify(Expression<(Of</pre>
                                     was performed on the mock,
<<'(Action<(Of <<'(T>)>>)>)>),
                                      specifying a failure error message.
String)
                                      Use in conjuntion with the default
                                      Loose.
                                      Verifies that a specific invocation
                                      matching the given expression
Verify(Expression<(Of</pre>
```

 $\leq \leq (Action \leq (Of \leq \leq (T \geq) \geq) \geq),$ Times, String) was performed on the mock, specifying a failure error message. Use in conjuntion with the default Loose.

<u>Verify<(Of <<'(TResult>)>>)</u>

(Expression<(Of <<'(Func<(Of <<('T, TResult>)>)>>)>>))

Verify<(Of <<'(TResult>)>>)

(Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>), Boolean)

Verify<(Of <<'(TResult>)>>)

(Expression<(Of <<'(Func<(Of <<(<'T, TResult>)>)>>), Boolean, T)

<u>VerifyAll</u>

Verifies all expectations regardless of whether they have been flagged as verifiable. (Inherited from Mock.)

<u>VerifyGet<(Of <<'(TProperty>)>>)</u>
(Expression<(Of <<'(Func<(Of

(Expression<(Of <<'(Func<(Of

<(<'T, TResult>)>)>>)>))

VerifyGet<(Of <<'(TProperty>)>>)

(Expression≤(Of ≤≤'(Func≤(Of

<(<'T, TResult>)>)>>), Boolean)

 $\underline{VerifyGet} \leq (\underline{Of} \leq \leq '(\underline{TProperty} \geq) \geq \geq)$

(Expression<(Of <<'(Func<(Of <<(<'T, TResult>)>)>>), Boolean, T)

 \vee <u>VerifySet(Action<(Of <<'(T>)>>))</u>

VerifySet(Action<(Of <<'(T>)>>),
Times)

 $\frac{\text{VerifySet(Action} < (\text{Of} < <'(\text{T}>) >>),}{\text{String})}$

Verifies that a property was set on the mock.

Verifies that a property was set on the mock.

Verifies that a property was set on the mock, specifying a failure message.

Verifies that a property was set on

- $\frac{\text{VerifySet(Action} < (\text{Of} < < '(\text{T} >) > >),}{\text{Times, String)}} \text{ the mock, specifying a failure message.}$
- **♦** When

Extension Methods

	Name	Description
₩	InSequence<(Of <<'(TMock>)>>)	Perform an expectation in the trace. (Defined by MockSequenceHelper.)
₩	<u>Protected<(Of <<'(T>)>>)</u>	Enable protected setups for the mock. (Defined by ProtectedExtension .)
•	SetupSequence<(Of <<'(TMock, TResult>)>>)	(Defined by SequenceExtensions .)
•	<pre>SetupSet<(Of <<'(T, TProperty>)>>)</pre>	(Defined by MockExtensions.)
₽ į	VerifySet<(Of <<'(T, TProperty>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>)))	Overloaded. (Defined by MockExtensions.)
Q į	VerifySet<(Of <<'(T, TProperty>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>), Boolean)	Overloaded. (Defined by MockExtensions.)
∳ įį	VerifySet<(Of <<'(T, TProperty>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>), Boolean)	Overloaded. (Defined by MockExtensions.)
•	VerifySet<(Of <<'(T, TProperty>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>), Boolean, T)	Overloaded. (Defined by MockExtensions.)

Mock<(Of <(<'T>)>)> Class Moq Namespace

-

□ C#

Moq

Mock<(Of <(<'T>)>)>..::.OnGetObject Method

<u>Mock<(Of <(<'T>)>)> Class See Also Send Feedback</u>

Returns the mocked object value.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

protected override Object OnGetObject()

Mock<(Of <(<'T>)>)> Class Moq Namespace

□ C#

☐ Include Protected Members

□ Include Inherited Members

Moq

Mock < (Of < (<'T>)>)>.....Raise Method

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

Overload List

Name	Description
<<'(1>1>>1	Raises the event referenced in eventExpression using the given args argument.
Raise(Action<(Of	Raises the event referenced in eventExpression using the given args argument for a non-
<u>array<object>[]()[][]</object></u>)	EventHandler typed event.

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Members Moq Namespace

-

□ C#

Moq

 $Mock < (Of < (<'T>)>)> \dots Raise Method (Action < (Of < (<'T>)>)>, EventArgs)$

Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback

Raises the event referenced in eventExpression using the given args argument.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

args

Type: <u>System..:..EventArgs</u>

C#

Examples

The following example shows how to raise a **PropertyChanged** event:

C#
var mock = new Mock<IViewModel>();
mock.Raise(x => x.PropertyChanged -= null, new PropertyChangedEventA

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 presenter = new OrdersPresenter(mockView.Object);

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

// Raise the event with a specific arguments data
mockView.Raise(v => v.SelectionChanged += null, new OrderEventArgs {

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

Exceptions

Exception

Condition

The args argument is invalid for the target event System.::..ArgumentException invocation, or the eventExpression is not an event attach or detach expression.

Mock<(Of <(<'T>)>)> Class Raise Overload Moq Namespace

```
□ C#

Moq

Mock<(Of <(<'T>)>)>..::..Raise Method (Action<(Of <(<'T>)>)>,
array<Object>[]()[][])

Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback
```

Raises the event referenced in eventExpression using the given args argument for a non-EventHandler typed event.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

Parameters

```
eventExpression
     Type: System..:..Action<(Of <(<'T>)>)>
args
     Type: array<System..:..Object>[]()[][]
```

Examples

The following example shows how to raise a custom event that does not adhere to the standard EventHandler:

C# □ Copy

```
var mock = new Mock<IViewModel>();
mock.Raise(x => x.MyEvent -= null, "Name", bool, 25);
```

Exceptions

Exception

Condition

The args arguments are invalid for the target System.::.ArgumentException event invocation, or the eventExpression is not an event attach or detach expression.

Mock<(Of <(<'T>)>)> Class Raise Overload Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Mock<(Of <(<'T>)>)>.....Setup Method

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

Overload List

Name

- $\underbrace{ Setup(Expression < (Of <<'(Action < (Of <<'(T>)>>)>>))}_{})}_{}$
 - Setup<(Of <<'(TResult>)>>)

Description

Specifies a setup on the mocked type for a call to to a void method.

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Members Moq Namespace

_

□ C#

Moq

Mock<(Of <(<'T>)>)>.....Setup Method (Expression<(Of <(<'Action<(Of <(<'T>)>)>>)>))

Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback

Specifies a setup on the mocked type for a call to to a void method.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

expression

Lambda expression that specifies the expected method invocation.

Remarks

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

Examples

C#

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

Mock<(Of <(<'T>)>)> Class
Setup Overload
Moq Namespace

□ C#

Moq

Mock<(Of <(<'T>)>)>..::.Setup<(Of <(<'TResult>)>)> Method
(Expression<(Of <(<'Func>)>)>)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Mock`1.Setup``1(System.Linq.Expressions.Expression{System.Func`2})"]

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression\\
```

Type Parameters

TResult

Mock<(Of <(<'T>)>)> Class
Setup Overload
Moq Namespace

```
□ C#

Moq

Mock<(Of <(<'T>)>)>..::..SetupAllProperties Method

Mock<(Of <(<'T>)>)> Class See Also Send Feedback
```

Specifies that the all properties on the mock should have "property behavior", meaning that setting its value will cause it to be saved and later returned when the property is requested. (this is also known as "stubbing"). The default value for each property will be the one generated as specified by the <u>DefaultValue</u> property for the mock.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public Mock<T> SetupAllProperties()

Remarks

If the mock <u>DefaultValue</u> is set to <u>Mock</u>, the mocked default values will also get all properties setup recursively.

Mock<(Of <(<'T>)>)> Class Moq Namespace

□ C#

Moq

 $Mock < (Of < (<'T>)>)>SetupGet < (Of < (<'TProperty>)>)> Method \\ \underline{Mock} < (Of < (<'T>)>)> Class See Also Send Feedback$

[Missing <summary> documentation for "M:Moq.Mock`1.SetupGet``1(System.Linq.Expressions.Expression{System.Func`2})"]

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

expression

Type Parameters

TProperty

Mock<(Of <(<'T>)>)> Class Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Mock<(Of <(<'T>)>)>.....SetupProperty Method

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

Overload List

	Name	Description
≡	SetupProperty<(Of <<'(TProperty>)>>)(Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)))	
≡	<u>SetupProperty<(Of <<'(TProperty>)>>)(Expression<(Of <<''(Func<(Of <(<'T, TResult>)>)>>), Boolean)</u>	

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Members Moq Namespace

□ C#

_

Moq

Mock < (Of < (<'T>)>)>SetupProperty < (Of < (<'TProperty>)>)> Method (Expression < (Of < (<'Func>)>)>)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Mock`1.SetupProperty``1(System.Linq.Expressions.Expression{System.Func`2})"]

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
property
```

```
Type: <u>System.Linq.Expressions..:..Expression</u><(Of <(<'<u>Func</u><(<u>Of</u> <(<'<u>T</u>, <u>TResult</u>>)>)>)>)>
```

Type Parameters

TProperty

Mock<(Of <(<'T>)>)> Class SetupProperty Overload Moq Namespace

□ C#

Moq

Mock<(Of <(<'T>)>)>.....SetupProperty<(Of <(<'TProperty>)>)> Method
(Expression<(Of <(<'Func>)>)>, Boolean)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Mock`1.SetupProperty``1(System.Linq.Expressions.Expression{System.Func`2},System.Boo

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
property
```

```
Type: <u>System.Linq.Expressions..:..Expression</u><(Of <(<'<u>Func</u><(<u>Of</u> <(<'<u>T, TResult</u>>)>)>)>)>
```

initialValue

Type: <u>System..:..Boolean</u>

Type Parameters

TProperty

Mock<(Of <(<'T>)>)> Class SetupProperty Overload Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq

Mock<(Of <(<'T>)>)>..::..SetupSet Method

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

Overload List

Name	Description
<u>SetupSet<(Of</u> <u><<'(TProperty>)>>)(Action<(Of</u> <u><<'(T>)>>))</u>	Specifies a setup on the mocked type for a call to to a property setter.
= W	Specifies a setup on the mocked type for a call to to a property setter.

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Members Moq Namespace

-

□ C#

Moq

Mock<(Of <(<'T>)>)>..:..SetupSet<(Of <(<'TProperty>)>)> Method (Action<(Of <(<'T>)>)>)

Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback

Specifies a setup on the mocked type for a call to to a property setter.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

setter Expression

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

The Lambda expression that sets a property to a value.

Type Parameters

TProperty

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

Remarks

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

This overloads allows the use of a callback already typed for the property type.

Examples

C#

mock.SetupSet(x => x.Suspended = true);

Mock<(Of <(<'T>)>)> Class SetupSet Overload Moq Namespace

⊟ □ C#

Moq

 $Mock < (Of < (<'T>)>)>SetupSet\ Method\ (Action < (Of < (<'T>)>)>)$

<u>Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback</u>

Specifies a setup on the mocked type for a call to to a property setter.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

setter Expression

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

Lambda expression that sets a property to a value.

Remarks

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

Examples

C#

mock.SetupSet(x => x.Suspended = true);

Mock<(Of <(<'T>)>)> Class SetupSet Overload Moq Namespace

□ **C**#

☐ Include Protected Members

□ Include Inherited Members

Moq

 $Mock < (Of < (< \top >) >) > Verify Method$

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

Overload List

	Name	Description
= (<u>Verify()()()</u> ()	Verifies that all verifiable expectations have been met. (Inherited from Mock.)
≟	<u>Verify(Expression<(Of</u> <<'(Action<(Of <<'(T>)>>)>))	Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjuntion with the default Loose.
= Q	Verify<(Of <<'(TResult>)>>) (Expression<(Of <<'(Func<(Of <('T, TResult>)>)>>)>>)))	
≟	Verify(Expression<(Of <<'(Action<(Of <<'(T>)>>)>)>), Times)	Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjuntion with the default Loose.
≘	Verify(Expression<(Of <<'(Action<(Of <<'(T>)>>)>)>), String)	Verifies that a specific invocation matching the given expression was performed on the mock, specifying a failure error message. Use in conjuntion with the default Loose.
∄	Verify<(Of <<'(TResult>)>>) (Expression<(Of <<'(Func<(Of <((-T, TResult>)>)>>)>>), Boolean)	
	Verify<(Of <<'(TResult>)>>) (Expression<(Of <<'(Func<(Of <(('T, TResult>)>)>>)>), Boolean)	
⊒	Verify(Expression<(Of <<'(Action<(Of <<'(T>)>>)>)>), Times, String)	Verifies that a specific invocation matching the given expression was performed on the mock, specifying a failure error message. Use in conjuntion with the default Loose.

Verify<(Of <<'(TResult>)>>)
(Expression<(Of <<'(Func<(Of <<('T, TResult>)>)>>)>>),
Boolean, T)

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Members Moq Namespace

⊟ □ C#

Moq

Mock<(Of <(<'T>)>)>.....Verify Method (Expression<(Of <(<'Action<(Of <(<'T>)>)>>)>))

Mock<(Of <(<'T>)>)> Class 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: 4.0.10827.0 (4.0.0.0)

Syntax

Parameters

```
expression\\
```

```
Type: <u>System.Linq.Expressions..:..Expression</u><(Of <(<'<u>Action</u><(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

Condition

Moq..:..MockException The invocation was not performed on the mock.

Mock<(Of <(<'T>)>)> Class
Verify Overload
Moq Namespace

□ C#

Moq

Mock<(Of <(<'T>)>)>.....Verify<(Of <(<'TResult>)>)> Method
(Expression<(Of <(<'Func>)>)>)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Mock`1.Verify``1(System.Linq.Expressions.Expression{System.Func`2})"]

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

expression

```
Type: System.Linq.Expressions..:..Expression<(Of <(<'Func<(Of <(<'T, TResult>)>)>)>)>
```

Type Parameters

TResult

Mock<(Of <(<'T>)>)> Class
Verify Overload
Moq Namespace

□ C#

_

Moq

Mock<(Of <(<'T>)>)>.....Verify Method (Expression<(Of <(<'Action<(Of <(<'T>)>)>)>)>, Times)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression
```

```
Type: <u>System.Linq.Expressions..::.Expression</u><(Of <(<'<u>Action</u><(Of <(<'<u>T</u>>)>)>)> Expression to verify.
```

times

Type: Moq..:..Times

The number of times a method is allowed to be called.

Exceptions

Exception

Condition

Moq..:..MockException The invocation was not call the times specified by times.

Mock<(Of <(<'T>)>)> Class
Verify Overload
Moq Namespace

-

□ C#

Moq

Mock<(Of <(<'T>)>)>.....Verify Method (Expression<(Of <(<'Action<(Of <(<'T>)>)>>)>), String)

Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback

Verifies that a specific invocation matching the given expression was performed on the mock, specifying a failure error message. Use in conjuntion with the default <u>Loose</u>.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression
```

```
Type: <u>System.Linq.Expressions..::.Expression</u><(Of <(<'<u>Action</u><(Of <(<'<u>T</u>>)>)>)> Expression to verify.
```

failMessage

Type: <u>System....String</u>

Message to show if verification fails.

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

Condition

Moq..:..MockException The invocation was not performed on the mock.

Mock<(Of <(<'T>)>)> Class
Verify Overload
Moq Namespace

□ C#

_

Moq

Mock<(Of <(<'T>)>)>..:..Verify<(Of <(<'TResult>)>)> Method (Expression<(Of <(<'Func>)>)>, Boolean)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for

"M:Moq.Mock`1.Verify``1(System.Linq.Expressions.Expression{System.Func`2},System.Boolean)"]

Namespace: Moq

Assemblies: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression
```

times

Type: <u>System..:..Boolean</u>

Type Parameters

TResult

Mock<(Of <(<'T>)>)> Class
Verify Overload
Moq Namespace

⊟ □ C#

Moq

Mock<(Of <(<'T>)>)>.....Verify Method (Expression<(Of <(<'Action<(Of <(<'T>)>)>)>)>, Times, String)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

Verifies that a specific invocation matching the given expression was performed on the mock, specifying a failure error message. Use in conjuntion with the default <u>Loose</u>.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression
```

```
Type: <u>System.Linq.Expressions..::.Expression</u><(Of <(<'<u>Action</u><(Of <(<'<u>T</u>>)>)>)>) Expression to verify.
```

times

Type: Moq..:..Times

The number of times a method is allowed to be called.

failMessage

Type: <u>System..:..String</u>

Message to show if verification fails.

Exceptions

Exception

Condition

Moq..:..MockException The invocation was not call the times specified by times.

Mock<(Of <(<'T>)>)> Class
Verify Overload
Moq Namespace

□ C#

Moq

Mock<(Of <(<'T>)>)>.....Verify<(Of <(<'TResult>)>)> Method
(Expression<(Of <(<'Func>)>)>, Boolean, T)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Mock`1.Verify``1(System.Linq.Expressions.Expressions(System.Func`2),System.Boolean,`0)"

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression
```

```
Type: <u>System.Linq.Expressions..::.Expression</u><(Of <(<'<u>Func</u><(<u>Of</u> <(<'<u>T, TResult</u>>)>)>)>
```

times

Type: System.::..Boolean

failMessage

Type: T

Type Parameters

TResult

Mock<(Of <(<'T>)>)> Class
Verify Overload
Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq

Mock<(Of <(<'T>)>)>.....VerifyGet Method

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

Overload List

	Name	Description
=	<pre>VerifyGet<(Of <<'(TProperty>)>>)(Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)))</pre>	
=	VerifyGet<(Of <<'(TProperty>)>>)(Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>), Boolean)	
	<u>VerifyGet<(Of <<'(TProperty>)>>)(Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>), Boolean)</u>	
=	VerifyGet<(Of <<'(TProperty>)>>)(Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>), Boolean, T)	

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Members Moq Namespace

⊟ □ C#

Moq

Mock<(Of <(<'T>)>)>.....VerifyGet<(Of <(<'TProperty>)>)> Method (Expression<(Of <(<'Func>)>)>)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Mock`1.VerifyGet``1(System.Linq.Expressions.Expression{System.Func`2})"]

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression\\
```

Type Parameters

TProperty

Mock<(Of <(<'T>)>)> Class VerifyGet Overload Moq Namespace

□ C#

_

Moq

Mock < (Of < (<'T>)>)>......VerifyGet < (Of < (<'TProperty>)>)> Method (Expression < (Of < (<'Func>)>)>, Boolean)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for

"M:Moq.Mock`1.VerifyGet``1 (System.Linq.Expressions.Expressions(System.Func`2), System.Boolean)

Namespace: Moq

Assemblies: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression
```

```
Type: <u>System.Linq.Expressions..:..Expression</u><(Of <(<'<u>Func</u><(<u>Of</u> <(<'<u>T, TResult</u>>)>)>)>)>
```

failMessage

Type: <u>System..:..Boolean</u>

Type Parameters

TProperty

Mock<(Of <(<'T>)>)> Class VerifyGet Overload Moq Namespace

□ C#

Moq

Mock<(Of <(<'T>)>)>.....VerifyGet<(Of <(<'TProperty>)>)> Method
(Expression<(Of <(<'Func>)>)>, Boolean, T)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Mock`1.VerifyGet``1(System.Linq.Expressions.Expressions(System.Func`2),System.Boolean,

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression
```

```
Type: <u>System.Linq.Expressions..::.Expression</u><(Of <(<'<u>Func</u><(<u>Of</u> <(<'<u>T, TResult</u>>)>)>)>)>
```

times

Type: System.::..Boolean

failMessage

Type: T

Type Parameters

TProperty

Mock<(Of <(<'T>)>)> Class VerifyGet Overload Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq

Mock < (Of < (<'T>)>)>......VerifySet Method

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

Overload List

	Name	Description
≡	<u>VerifySet(Action</u> <(Of <<'(T≥)≥≥))	Verifies that a property was set on the mock.
≡	<pre>VerifySet(Action<(Of <<'(T>)>>), Times)</pre>	Verifies that a property was set on the mock.
≡	<u>VerifySet(Action<(Of</u> <<' <u>(T>)>>)</u> , String)	Verifies that a property was set on the mock, specifying a failure message.
≡	<pre>VerifySet(Action<(Of <<'(T>)>>), Times, String)</pre>	Verifies that a property was set on the mock, specifying a failure message.

Mock<(Of <(<'T>)>)> Class Mock<(Of <(<'T>)>)> Members Moq Namespace

-

□ C#

Moq

Mock<(Of <(<'T>)>)>..:..VerifySet Method (Action<(Of <(<'T>)>)>)

<u>Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback</u>

Verifies that a property was set on the mock.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

Parameters

Examples

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

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 = true);
```

Exceptions

Exception

Condition

Moq..:..MockException The invocation was not performed on the mock.

Mock<(Of <(<'T>)>)> Class VerifySet Overload Moq Namespace

-

□ C#

Moq

Mock < (Of < (<'T>)>)> VerifySet Method (Action < (Of < (<'T>)>)>, Times)

<u>Mock<(Of <(<'T>)>)> Class See Also Send Feedback</u>

Verifies that a property was set on the mock.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
setter Expression\\
```

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

times

Type: Moq..:..Times

The number of times a method is allowed to be called.

Exceptions

Exception

Condition

Moq..:..MockException The invocation was not call the times specified by times.

Mock<(Of <(<'T>)>)> Class VerifySet Overload Moq Namespace

-

□ C#

Moq

Mock < (Of < (<'T>)>)> VerifySet Method (Action < (Of < (<'T>)>)>, String)

Mock<(Of <(<'T>)>)> Class Example See Also Send Feedback

Verifies that a property was set on the mock, specifying a failure message.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
setter Expression\\
```

Type: $\underline{\text{System.::..}Action} < (\text{Of } < (< '\underline{\text{T}}>)>)> \\ \text{Expression to verify.}$

failMessage

Type: <u>System..:..String</u>

Message to show if verification fails.

Examples

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

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 = true, "Warehouse sh
```

Exceptions

Exception

Condition

Moq..:..MockException The invocation was not performed on the mock.

Mock<(Of <(<'T>)>)> Class VerifySet Overload Moq Namespace

-

□ C#

Moq

Mock<(Of <(<'T>)>)>.....VerifySet Method (Action<(Of <(<'T>)>)>, Times, String)

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

Verifies that a property was set on the mock, specifying a failure message.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
setter Expression\\
```

Type: $\underline{\text{System.::..}Action} < (\text{Of } < (<'\underline{\text{T}}>)>)> \\ \text{Expression to verify.}$

times

Type: Moq..:..Times

The number of times a method is allowed to be called.

failMessage

Type: System..:..String

Message to show if verification fails.

Exceptions

Exception

Condition

Moq..:..MockException The invocation was not call the times specified by times.

Mock<(Of <(<'T>)>)> Class VerifySet Overload Moq Namespace

```
-
```

□ C#

Moq

Mock < (Of < (< T>)>)>.....When Method

Mock<(Of <(<'T>)>)> Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Mock`1.When(System.Func`1)"]

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

condition

Type: <u>System..:..Func<(Of <(<'TResult>)>)></u>

Mock<(Of <(<'T>)>)> Class Moq Namespace

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq

Mock<(Of <(<'T>)>)> Properties

Mock<(Of <(<'T>)>)> Class See Also Send Feedback
```

The $\underline{Mock} \le (\underline{Of} \le (\underline{\le'T} \ge) \ge)$ type exposes the following members.

\blacksquare Properties

Name	Description
<u>Behavior</u>	Behavior of the mock, according to the value set in the constructor. (Inherited from Mock.)
<u>CallBase</u>	Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalsefalse (False in Visual Basic). (Inherited from Mock.)
<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks. (Inherited from Mock.)
<u>Object</u>	Exposes the mocked object instance.

Mock<(Of <(<'T>)>)> Class Moq Namespace

```
-
```

□ 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: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public virtual T Object { get; }

Mock<(Of <(<'T>)>)> Class Moq Namespace

-

□ C#

Moq

MockBehavior Enumeration

See Also Send Feedback

Options to customize the behavior of the mock.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public enum MockBehavior

Members

Member name Description

Strict

Loose

Default

Moq Namespace

_

□ C#

Moq

MockException Class

Members See Also Send Feedback

Exception thrown by mocks when setups are not matched, the mock is not properly setup, etc.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare 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

System..:..Object

System.::..Exception
Moq..::..MockException

MockException Members Moq Namespace

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

The MockException type exposes the following members.

Constructors

Name

Description

·

MockException Supports the serialization infrastructure.

Methods

	Name	Description
=(Get∩hiectData	Supports the serialization infrastructure.
		(Overrides
	<u>GelObjectData</u>	(Overrides Exception:GetObjectData(SerializationInfo ,
		StreamingContext).)

MockException Class Moq Namespace

_

□ C#

Moq

MockException Constructor

MockException Class See Also Send Feedback

Supports the serialization infrastructure.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

info

Type: <u>System.Runtime.Serialization...</u>:..<u>SerializationInfo</u> Serialization information.

context

Type: <u>System.Runtime.Serialization..:..StreamingContext</u> Streaming context.

MockException Class Moq Namespace

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

The MockException type exposes the following members.

Methods

	Name	Description
=(Get∩hiectData	Supports the serialization infrastructure.
		(Overrides
	<u>GelObjectData</u>	(Overrides Exception:GetObjectData(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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

info

Type: <u>System.Runtime.Serialization...</u>:..<u>SerializationInfo</u> Serialization information.

context

Type: <u>System.Runtime.Serialization..:..StreamingContext</u> Streaming context.

Implements

MockException Class Moq Namespace

-

□ C#

Moq

MockExtensions Class

Members See Also Send Feedback

Provides additional methods on mocks.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public static class MockExtensions

Inheritance Hierarchy

System..:..Object

Moq..:..MockExtensions

MockExtensions Members Moq Namespace

□ C#
☐ Include Protected Members
□ Include Inherited Members
Moq
MockExtensions Members
MockExtensions Class Methods See Also Send Feedback

The MockExtensions type exposes the following members.

\blacksquare Methods

	Name	Description
=0	SetupSet<(Of <<'(T, TProperty>)>>)	Obsolete.
€0	<u>VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>))</u>	Obsolete.
=∅	VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>), Boolean)	Obsolete.
= Q	VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>), Boolean, T)	Obsolete.

MockExtensions Class Moq Namespace

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

The MockExtensions type exposes the following members.

\blacksquare Methods

	Name	Description
=0	SetupSet<(Of <<'(T, TProperty>)>>)	Obsolete.
€0	<u>VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>))</u>	Obsolete.
=∅	VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>), Boolean)	Obsolete.
= Q	VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>), Boolean, T)	Obsolete.

MockExtensions Class Moq Namespace

-

□ C#

Moq

MockExtensions..:..SetupSet<(Of <(<'T, TProperty>)>)> Method

MockExtensions Class See Also Send Feedback

[Missing <summary> documentation for

"M:Moq.MockExtensions.SetupSet``2(Moq.Mock{``0},System.Linq.Expressions.Expression{System.F

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Type Parameters

T TProperty

MockExtensions Class Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq

 $Mock Extensions.....Verify Set\ Method$

MockExtensions Class See Also Send Feedback

Overload List

	Name	Description
∉∳S	<u>VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>))</u>	Obsolete.
ĕ∳S	<u>VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>), Boolean)</u>	Obsolete.
	<u>VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>), Boolean)</u>	
∉∳S	<u>VerifySet<(Of <<'(T, TProperty>)>>)(Mock<(Of <<'(T>)>>), Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>), Boolean, T)</u>	Obsolete.

MockExtensions Class
MockExtensions Members
Moq Namespace

⊟ □ C#

Moq

MockExtensions..:..VerifySet<(Of <(<'T, TProperty>)>)> Method (Mock<(Of <(<'T>)>)>, Expression<(Of <(<'Func>)>)>)

MockExtensions Class See Also Send Feedback

[Missing <summary> documentation for

 $"M: Moq. Mock Extensions. Verify Set ``2 (Moq. Mock \{``0\}, System. Linq. Expressions. Expression \{System. Linq. Expression \{System. Linq. Expression \} \} = \{System. Linq. Expression \{System. Linq. Expression \} \} = \{System. Linq. Expression \} = \{System. Linq$

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Type Parameters

T TProperty

MockExtensions Class VerifySet Overload Moq Namespace

⊟ □ C#

Moq

MockExtensions..:..VerifySet<(Of <(<'T, TProperty>)>)> Method (Mock<(Of <(<'T>)>)>, Expression<(Of <(<'Func>)>)>, Boolean)

MockExtensions Class See Also Send Feedback

[Missing <summary> documentation for

 $"M: Moq. Mock Extensions. Verify Set ``2 (Moq. Mock \{``0\}, System. Linq. Expressions. Expression \{System. Linq. Expression \{System. Linq. Expression \} \} = \{System. Linq. Expression \{System. Linq. Expression \} \} = \{System. Linq. Expression \} = \{System. Linq$

Namespace: Moq

Assemblies: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Type Parameters

T TProperty

MockExtensions Class VerifySet Overload Moq Namespace

⊟ □ C#

Moq

MockExtensions..:..VerifySet<(Of <(<'T, TProperty>)>)> Method (Mock<(Of <(<'T>)>)>, Expression<(Of <(<'Func>)>)>, Boolean, T)

MockExtensions Class See Also Send Feedback

[Missing <summary> documentation for

 $"M: Moq. Mock Extensions. Verify Set ``2 (Moq. Mock \{``0\}, System. Linq. Expressions. Expression \{System. Linq. Expression \{System. Linq. Expression \} \} = \{System. Linq. Expression \{System. Linq. Expression \} \} = \{System. Linq. Expression \} = \{System. Linq$

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

```
[ObsoleteAttribute("Replaced by VerifySet(Action, Times, string)")]
public static void VerifySet<T, TProperty>(
         this Mock<T> mock,
         Expression<Func> expression,
         bool times,
         T failMessage
where T : class
Parameters
mock
    Type: <u>Moq..:..Mock</u><(Of <(<'T>)>)>
expression
    Type: System.Ling.Expressions..:..Expression<(Of <(<'Func<(Of <(<'T,
    <u>TResult>)>)></u>>)>)>
times
    Type: <u>System..:..Boolean</u>
fail Message \\
    Type: T
```

Type Parameters

T TProperty

MockExtensions Class VerifySet Overload Moq Namespace

-

□ C#

Moq

MockFactory Class

Members Example See Also Send Feedback

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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

Remarks

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

This factory class helps in that scenario by providing a simplified creation of multiple mocks with a default $\underline{\text{MockBehavior}}$ (unless overriden by calling $\underline{\text{Create}} < (\underline{\text{Of}} < '(\underline{\text{T}} >) >>)(\underline{\text{MockBehavior}})$) 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 setup
// as we'll be validating all of them anyway.
foo.Setup(f => f.Do());
bar.Setup(b => b.Redo());

// exercise the mocks here

factory.VerifyAll();
// At this point all setups 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 setup will also throw a MockExceptic
```

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

C# Copy

```
var factory = new MockFactory(MockBehavior.Loose);
var foo = factory.Create<IFoo>();
var bar = factory.Create<IBar>();

// this setup will be verified when we verify the factory
foo.Setup(f => f.Do()).Verifiable();

// this setup will NOT be verified
foo.Setup(f => f.Calculate());

// this setup will be verified when we verify the factory
bar.Setup(b => b.Redo()).Verifiable();
```

```
// exercise the mocks here
// note that because the mocks are Loose, members
// called in the interfaces for which no matching
// setups exist will NOT throw exceptions,
// and will rather return default values.

factory.Verify();
// At this point verifiable setups 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>();
// specify setups
// exercise the mocks here
factory.Verify();
```

Inheritance Hierarchy

System..:..Object

Moq..:..MockFactory

Moq..:..MockRepository

MockFactory Members
Moq Namespace
Moq..:..MockBehavior

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

The MockFactory type exposes the following members.

Constructors

Name

Description

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

\blacksquare Methods

	Name	Description
=0	<u>Create<(Of <<'(T>)>>)()()()</u>	Creates a new mock with the default MockBehavior specified at factory construction time.
≓	<u>Create<(Of <<'(T>)>>)</u> (array <object>[]()[][])</object>	Creates a new mock with the default MockBehavior specified at factory construction time and with the the given constructor arguments for the class.
≡	Create<(Of <<'(T>)>>) (MockBehavior)	Creates a new mock with the given behavior.
= Q	<pre>Create<(Of <<'(T>)>>) (MockBehavior, array<object>[] ()[][])</object></pre>	Creates a new mock with the given behavior and with the the given constructor arguments for the class.
ē	CreateMock<(Of <<'(T>)>>)	Implements creation of a new mock within the factory.
≡	Verify	Verifies all verifiable expectations on all mocks created by this factory.
=	<u>VerifyAll</u>	Verifies all verifiable expectations on all mocks created by this factory.
ij 	<u>VerifyMocks</u>	Invokes verifyAction for each mock in Mocks, and accumulates the resulting MockVerificationException that might be thrown from the action.

Properties

Name	Description
<u>CallBase</u>	Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalse (False in Visual Basic).
<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks.
TOTOCKS	Gets the mocks that have been created by this factory and that will get verified together.

MockFactory Class Moq Namespace

_

□ C#

Moq

MockFactory Constructor

MockFactory Class See Also Send Feedback

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

Parameters

defaultBehavior

Type: Moq..:..MockBehavior

The behavior to use for mocks created using the $\frac{\text{Create} < (Of <<'(T>)>>)()()}{()()}$ factory method if not overriden by using the $\frac{\text{Create} < (Of <<'(T>)>>)}{()()}$

(MockBehavior) overload.

MockFactory Class Moq Namespace

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

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

\blacksquare Methods

	Name	Description
=0	<u>Create<(Of <<'(T>)>>)()()()</u>	Creates a new mock with the default MockBehavior specified at factory construction time.
≓	<u>Create<(Of <<'(T>)>>)</u> (array <object>[]()[][])</object>	Creates a new mock with the default MockBehavior specified at factory construction time and with the the given constructor arguments for the class.
≡	Create<(Of <<'(T>)>>) (MockBehavior)	Creates a new mock with the given behavior.
= Q	<pre>Create<(Of <<'(T>)>>) (MockBehavior, array<object>[] ()[][])</object></pre>	Creates a new mock with the given behavior and with the the given constructor arguments for the class.
ē	CreateMock<(Of <<'(T>)>>)	Implements creation of a new mock within the factory.
≡	Verify	Verifies all verifiable expectations on all mocks created by this factory.
=	<u>VerifyAll</u>	Verifies all verifiable expectations on all mocks created by this factory.
ij 	<u>VerifyMocks</u>	Invokes verifyAction for each mock in Mocks, 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
= Q	<u>Create<(Of <<'(T>)>>)()()()</u>	Creates a new mock with the default MockBehavior specified at factory construction time.
≘	<u>Create<(Of <<'(T>)>>)</u> (array <object>[]()[][])</object>	Creates a new mock with the default MockBehavior specified at factory construction time and with the the given constructor arguments for the class.
=	Create<(Of <<'(T>)>>) (MockBehavior)	Creates a new mock with the given behavior.
=0	<pre>Create<(Of <<'(T>)>>) (MockBehavior, array<object>[] ()[][])</object></pre>	Creates a new mock with the given behavior and with the the given constructor arguments for the class.

MockFactory Class
MockFactory Members
Moq Namespace

-

□ C#

Moq

MockFactory..:..Create<(Of <(<'T>)>)> Method

MockFactory Class 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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

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

■Type Parameters

T

Type to mock.

Return Value

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

Examples

C#

```
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 (array<Object>[]()[][])

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 given constructor arguments for the class.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

args

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

■Type Parameters

T

Type to mock.

Return Value

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

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#

```
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

_

□ C#

Moq

MockFactory..:..Create<(Of <(<'T>)>)> Method (MockBehavior)

MockFactory Class Example See Also Send Feedback

Creates a new mock with the given behavior.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

behavior

Type: Moq..:..MockBehavior

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

■Type Parameters

T

Type to mock.

Return Value

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

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

```
□ C#
```

_

Moq

MockFactory..::..Create<(Of <(<'T>)>)> Method (MockBehavior, array<Object>[]()[][])

MockFactory Class Example See Also Send Feedback

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

behavior

Type: Moq..:..MockBehavior

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

args

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

■Type Parameters

T

Type to mock.

Return Value

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

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

-

□ C#

Moq

MockFactory.....CreateMock < (Of < (< 'T >) >) > Method

MockFactory Class See Also Send Feedback

Implements creation of a new mock within the factory.

Namespace: Moq

Syntax

C#

Parameters

behavior

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

args

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

Type Parameters

T

Type to mock.

MockFactory Class Moq Namespace

-

□ C#

Moq

MockFactory..:..Verify Method

MockFactory Class See Also Send Feedback

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

Namespace: Moq

\blacksquare Syntax

C#

public virtual void Verify()

Exceptions

Exception

Condition

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

MockFactory Class
Moq Namespace
Mock..:..Verify()()()()

-

□ C#

Moq

MockFactory..:..VerifyAll Method

MockFactory Class See Also Send Feedback

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

Namespace: Moq

\blacksquare Syntax

C#

public virtual void VerifyAll()

Exceptions

Exception

Condition

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

MockFactory Class
Moq Namespace
Mock..:..Verify()()()()

-

□ C#

Moq

MockFactory..:..VerifyMocks Method

MockFactory Class See Also Send Feedback

Invokes verifyAction for each mock in <u>Mocks</u>, and accumulates the resulting MockVerificationException that might be thrown from the action.

Namespace: Moq

Syntax

C#

```
)
```

Parameters

verifyAction

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

MockFactory Class Moq Namespace

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

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

Properties

Name	Description
<u>CallBase</u>	Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalse (False in Visual Basic).
<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks.
TOTOCKS	Gets the mocks that have been created by this factory and that will get verified together.

MockFactory Class Moq Namespace

-

□ C#

Moq

MockFactory..:..CallBase Property

MockFactory Class See Also Send Feedback

Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalsefalse (False in Visual Basic).

Namespace: Moq

$\blacksquare Syntax$

```
C#
```

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

MockFactory Class Moq Namespace

_

□ C#

Moq

MockFactory..:..DefaultValue Property

MockFactory Class See Also Send Feedback

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

Namespace: Moq

\blacksquare Syntax

C#

```
public DefaultValue DefaultValue { get; set; }
```

MockFactory Class Moq Namespace

-

□ C#

Moq

MockFactory..:..Mocks Property

MockFactory Class See Also Send Feedback

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

Namespace: Moq

$\blacksquare Syntax$

C#

protected internal <u>IEnumerable</u><<u>Mock</u>> Mocks { get; }

MockFactory Class Moq Namespace

-

□ C#

Moq

MockRepository Class

Members Example See Also Send Feedback

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

Namespace: Moq

$\blacksquare Syntax$

C#

public class MockRepository : MockFactory

Remarks

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

This repository class helps in that scenario by providing a simplified creation of multiple mocks with a default $\underline{\text{MockBehavior}}$ (unless overriden by calling $\underline{\text{Create}}(\underline{\text{Of}} <<'(\underline{\text{T}}>)>>)(\underline{\text{MockBehavior}})$) and posterior verification.

Examples

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

C# □ Copy

```
var repository = new MockRepository(MockBehavior.Strict);

var foo = repository.Create<IFoo>();

var bar = repository.Create<IBar>();

// no need to call Verifiable() on the setup

// as we'll be validating all of them anyway.

foo.Setup(f => f.Do());

bar.Setup(b => b.Redo());

// exercise the mocks here

repository.VerifyAll();

// At this point all setups 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 setup will also throw a MockExceptic
```

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

C# Copy

```
var repository = new MockRepository(MockBehavior.Loose);
var foo = repository.Create<IFoo>();
var bar = repository.Create<IBar>();

// this setup will be verified when we verify the repository foo.Setup(f => f.Do()).Verifiable();

// this setup will NOT be verified foo.Setup(f => f.Calculate());

// this setup will be verified when we verify the repository bar.Setup(b => b.Redo()).Verifiable();
```

```
// exercise the mocks here
// note that because the mocks are Loose, members
// called in the interfaces for which no matching
// setups exist will NOT throw exceptions,
// and will rather return default values.
repository.Verify();
// At this point verifiable setups are already checked
// and an optional MockException might be thrown.
```

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

C# Copy

```
var repository = new MockRepository(MockBehavior.Strict);

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

// specify setups

// exercise the mocks here
repository.Verify();
```

Inheritance Hierarchy

System..:..Object

Moq..:..MockFactory
Moq..:..MockRepository

MockRepository Members
Moq Namespace
Moq..:..MockBehavior

□ C#
□ Include Protected Members
□ Include Inherited Members
Moq
MockRepository Members
MockRepository Class Constructors Methods Properties See Also Send
<u>Feedback</u>

The MockRepository type exposes the following members.

Constructors

Name

Description

Initializes the repository with the given MockRepository defaultBehavior for newly created mocks from the repository.

\blacksquare Methods

	Name	Description
≓	<u>Create</u> <(<u>Of</u> <<' <u>(</u> T>)>>)()()()()	Creates a new mock with the default MockBehavior specified at factory construction time. (Inherited from MockFactory.)
≟	<u>Create<(Of <<'(T>)>>)</u> (array <object>[]()[][])</object>	Creates a new mock with the default MockBehavior specified at factory construction time and with the the given constructor arguments for the class. (Inherited from MockFactory.)
=	Create<(Of <<'(T>)>>) (MockBehavior)	Creates a new mock with the given behavior. (Inherited from MockFactory.)
≟	Create<(Of <<'(T>)>>) (MockBehavior, array <object>[] ()[][])</object>	Creates a new mock with the given behavior and with the the given constructor arguments for the class. (Inherited from MockFactory.)
Ģ [©]	CreateMock<(Of <<'(T>)>>)	Implements creation of a new mock within the factory. (Inherited from MockFactory.)
≟	<u>Of<(Of <<''(T>)>>)()()()()</u>	Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.
≟	<u>Of<(Of <<'(T>)>>)</u> (Expression<(Of <<'(Func<(Of <<('T, TResult>)>)>>)>>)))	
≟	<u>OneOf<(Of <<'(T>)>>)()()()()</u>	Creates an mock object of the indicated type.
≓	<u>OneOf<(Of <<'(T>)>>)</u> (Expression<(Of <<'(Func<(Of <<(<u>'T, TResult>)>)>>)>>)</u>))	V 1

Verify

VerifyAll

VerifyMocks

Verifies all verifiable expectations on all mocks created by this factory. (Inherited from MockFactory.)

Verifies all verifiable expectations on all mocks created by this factory. (Inherited from MockFactory.)

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

(Inherited from MockFactory.)

Properties

Name	Description
<u>CallBase</u>	Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalsefalse (False in Visual Basic). (Inherited from MockFactory.)
<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks. (Inherited from MockFactory.)
<u>Mocks</u>	Gets the mocks that have been created by this factory and that will get verified together. (Inherited from MockFactory.)

MockRepository Class Moq Namespace

-

□ C#

Moq

MockRepository Constructor

MockRepository Class See Also Send Feedback

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

defaultBehavior

Type: Moq..:..MockBehavior

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

MockRepository Class Moq Namespace

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

The MockRepository type exposes the following members.

\blacksquare Methods

	Name	Description
≓	<u>Create</u> <(<u>Of</u> <<' <u>(</u> T>)>>)()()()()	Creates a new mock with the default MockBehavior specified at factory construction time. (Inherited from MockFactory.)
≟	<u>Create<(Of <<'(T>)>>)</u> (array <object>[]()[][])</object>	Creates a new mock with the default MockBehavior specified at factory construction time and with the the given constructor arguments for the class. (Inherited from MockFactory.)
=	Create<(Of <<'(T>)>>) (MockBehavior)	Creates a new mock with the given behavior. (Inherited from MockFactory.)
≟	Create<(Of <<'(T>)>>) (MockBehavior, array <object>[] ()[][])</object>	Creates a new mock with the given behavior and with the the given constructor arguments for the class. (Inherited from MockFactory.)
Ģ [©]	CreateMock<(Of <<'(T>)>>)	Implements creation of a new mock within the factory. (Inherited from MockFactory.)
≟	<u>Of<(Of <<''(T>)>>)()()()()</u>	Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.
≟	<u>Of<(Of <<'(T>)>>)</u> (Expression<(Of <<'(Func<(Of <<('T, TResult>)>)>>)>>)))	
≟	<u>OneOf<(Of <<'(T>)>>)()()()()</u>	Creates an mock object of the indicated type.
≓	<u>OneOf<(Of <<'(T>)>>)</u> (Expression<(Of <<'(Func<(Of <<(<u>'T, TResult>)>)>>)>>)</u>))	V 1

Verify

VerifyAll

VerifyMocks

Verifies all verifiable expectations on all mocks created by this factory. (Inherited from MockFactory.)

Verifies all verifiable expectations on all mocks created by this factory. (Inherited from MockFactory.)

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

(Inherited from MockFactory.)

MockRepository Class Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
MockRepository..:..Create Method

MockRepository Class See Also Send Feedback

Overload List

	Name	Description
≓	<u>Create<(Of <<'(T>)>>)()()()</u>	Creates a new mock with the default MockBehavior specified at factory construction time. (Inherited from MockFactory.)
≓	<u>Create<(Of <<'(T>)>>)</u> (<u>array<object>[]()[][]</object></u>)	Creates a new mock with the default MockBehavior specified at factory construction time and with the the given constructor arguments for the class. (Inherited from MockFactory.)
=0	Create<(Of <<'(T>)>>) (MockBehavior)	Creates a new mock with the given behavior. (Inherited from MockFactory.)
≡	Create<(Of <<'(T>)>>) (MockBehavior, array <object>[] ()[][])</object>	Creates a new mock with the given behavior and with the the given constructor arguments for the class. (Inherited from MockFactory.)

MockRepository Class
MockRepository Members
Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
MockRepository..:..Of Method

MockRepository Class See Also Send Feedback

Overload List

Name

Description

• Of < (Of < < '(T>)>>)()()()

Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.

 $\underline{Of} \leq (\underline{Of} \leq \leq '(\underline{T} \geq) \geq \geq)$

(Expression < (Of

<<'(Func<(Of <(<'T,

<u>TResult>)>)>>)>>))</u>

MockRepository Class
MockRepository Members
Moq Namespace

-

□ C#

Moq

MockRepository..:..Of<(Of <(<'T>)>)> Method

MockRepository Class See Also Send Feedback

Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public <u>IQueryable</u><T> Of<T>()
where T : class

Type Parameters

T

The type of the mocked object to query.

MockRepository Class Of Overload Moq Namespace

```
□ C#

Moq

MockRepository..::..Of<(Of <(<'T>)>)> Method (Expression<(Of <(<'Func>)>)>)
```

[Missing <summary> documentation for "M:Moq.MockRepository.Of``1(System.Linq.Expressions.Expression{System.Func`2})"]

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

MockRepository Class See Also Send Feedback

Syntax

C#

Parameters

```
specification
```

```
Type: System.Linq.Expressions..:..Expression < (Of < (< 'Func< (Of < (< 'Func< (Of < (< 'T, TResult>)>)>)>
```

Type Parameters

T

MockRepository Class Of Overload Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
MockRepository..:..OneOf Method

MockRepository Class See Also Send Feedback

Overload List

Name

- OneOf<(Of <<'(T>)>>)()()()
- <u>OneOf<(Of <<'(T>)>>)(Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>))</u>

Description

Creates an mock object of the indicated type.

MockRepository Class
MockRepository Members
Moq Namespace

-

□ C#

Moq

MockRepository..::..OneOf<(Of <(<'T>)>)> Method

MockRepository Class See Also Send Feedback

Creates an mock object of the indicated type.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public T OneOf<T>()
where T : class

■Type Parameters

T

The type of the mocked object.

Return Value

The mocked object created.

MockRepository Class OneOf Overload Moq Namespace

□ C#

_

Moq

 $\label{lem:mockRepository.....OneOf<(Of <(<'T>)>)> Method (Expression<(Of <(<'Func>)>)>)$

MockRepository Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.MockRepository.OneOf``1(System.Linq.Expressions.Expressions(System.Func`2))"]

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

Parameters

```
specification
```

Type: System.Linq.Expressions..:..Expression < (Of < (< 'Func< (Of < (< 'Func< (Of < (< 'T, TResult>)>)>)>

Type Parameters

T

MockRepository Class OneOf Overload Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
MockRepository Properties

MockRepository Class See Also Send Feedback

The MockRepository type exposes the following members.

Properties

Name	Description
<u>CallBase</u>	Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalsefalse (False in Visual Basic). (Inherited from MockFactory.)
<u>DefaultValue</u>	Specifies the behavior to use when returning default values for unexpected invocations on loose mocks. (Inherited from MockFactory.)
<u>Mocks</u>	Gets the mocks that have been created by this factory and that will get verified together. (Inherited from MockFactory.)

MockRepository Class Moq Namespace

-

□ C#

Moq

Mocks Class

Members See Also Send Feedback

Allows querying the universe of mocks for those that behave according to the LINQ query specification.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public static class Mocks

Inheritance Hierarchy

System..:..Object Moq..:..Mocks

Mocks Members Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Mocks Members

Mocks Class Methods See Also Send Feedback

Methods

	Name	Description
∉∳S	<u>Of<(Of <<'(T>)>>)()()()()</u>	Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.
∉∳S	Of<(Of <<'(T>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)))	
≅∳S	<u>OneOf<(Of <<'(T>)>>)()()()</u>	Obsolete. Creates an mock object of the indicated type.
∉∳S	OneOf<(Of <<'(T>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>))	Obsolete.

Mocks Class Moq Namespace

_ □ C# ☐ Include Protected Members □ Include Inherited Members Moq Mocks Methods

Mocks Class See Also Send Feedback

Methods

	Name	Description
∉∳S	<u>Of<(Of <<'(T>)>>)()()()()</u>	Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.
∉∳S	Of<(Of <<'(T>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)))	
≅∳S	<u>OneOf<(Of <<'(T>)>>)()()()</u>	Obsolete. Creates an mock object of the indicated type.
∉∳S	OneOf<(Of <<'(T>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>))	Obsolete.

Mocks Class Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Mocks..:..Of Method

Mocks Class See Also Send Feedback

Overload List

	Name	Description
∉∳S	<u>Of<(Of <<'(T>)>>)()()()</u>	Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.
≅ 0 \$	Of<(Of <<'(T>)>>) (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)))	

Mocks Class Mocks Members Moq Namespace

-

□ C#

Moq

Mocks..:..Of<(Of <(<'T>)>)> Method

Mocks Class See Also Send Feedback

Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public static <u>IQueryable</u><T> Of<T>()
where T : class

Type Parameters

T

The type of the mocked object to query.

Mocks Class Of Overload Moq Namespace

```
□ C#

Moq

Mocks..:..Of<(Of <(<'T>)>)> Method (Expression<(Of <(<'Func>)>)>)

Mocks Class See Also Send Feedback

[Missing <summary> documentation for
"M:Moq.Mocks.Of``1(System.Linq.Expressions.Expression{System.Func`2})"]
```

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
specification
```

Type Parameters

T

Mocks Class Of Overload Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Mocks..:..OneOf Method

Mocks Class See Also Send Feedback

Overload List

	Name	Description
= \$ S	<u>OneOf<(Of <<'(T>)>>)()()()</u>	Obsolete. Creates an mock object of the indicated type.
ĕ\$ S	<u>OneOf<(Of <<'(T>)>>)(Expression<(Of <<''(Func<(Of <(<'T, TResult>)>)>>)>))</u>	Obsolete.

Mocks Class Mocks Members Moq Namespace

-

□ C#

Moq

Mocks..:..OneOf < (Of < (<'T>)>)> Method

Mocks Class See Also Send Feedback

Creates an mock object of the indicated type.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

[ObsoleteAttribute("Moved to Mock.Of<T>, as it's a single one, so not true)]
public static T OneOf<T>()
where T : class

■Type Parameters

T

The type of the mocked object.

Return Value

The mocked object created.

Mocks Class OneOf Overload Moq Namespace

```
□ C#

Moq

Mocks..:..OneOf<(Of <(<'T>)>)> Method (Expression<(Of <(<'Func>)>)>)

Mocks Class See Also Send Feedback

[Missing <summary> documentation for
"M:Moq.Mocks.OneOf`1(System.Linq.Expressions.Expression{System.Func`2})"]
```

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
specification
```

Type Parameters

T

Mocks Class OneOf Overload Moq Namespace

-

□ C#

Moq

MockSequence Class

Members See Also Send Feedback

Helper class to setup a full trace between many mocks

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public class MockSequence

Inheritance Hierarchy

System..:..Object

Moq..:..MockSequence

MockSequence Members Moq Namespace

-	
□ C#	
□ Inc	lude Protected Members
□ Inc	lude Inherited Members
Moq	
Mock	Sequence Members
Mock	Sequence Class Constructors Properties See Also Send Feedback

The MockSequence type exposes the following members.

Constructors

Name Description

MockSequence Initialize a trace setup

Properties

Description Name

<u>Cyclic</u> Allow sequence to be repeated

MockSequence Class Moq Namespace

-

□ C#

Moq

MockSequence Constructor

MockSequence Class See Also Send Feedback

Initialize a trace setup

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public MockSequence()

MockSequence Class Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
MockSequence Properties

MockSequence Class See Also Send Feedback

The MockSequence type exposes the following members.

Properties

Description Name

<u>Cyclic</u> Allow sequence to be repeated

MockSequence Class Moq Namespace

-

□ C#

Moq

MockSequence..:..Cyclic Property

MockSequence Class See Also Send Feedback

Allow sequence to be repeated

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

```
public bool Cyclic { get; set; }
```

MockSequence Class Moq Namespace

-

□ C#

Moq

MockSequenceHelper Class

Members See Also Send Feedback

define nice api

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public static class MockSequenceHelper

Inheritance Hierarchy

System..:..Object

Moq..:..MockSequenceHelper

MockSequenceHelper Members Moq Namespace

□ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
MockSequenceHelper Members
MockSequenceHelper Class Methods See Also Send Feedback

The MockSequenceHelper type exposes the following members.

Methods

Name

Description

≣**©**S

<u>InSequence<(Of <<'(TMock>)>>)</u> Perform an expectation in the trace.

MockSequenceHelper Class Moq Namespace

∃ ▶
□ C#
□ Include Protected Members
□ Include Inherited Members
Moq
MockSequenceHelper Methods
MockSequenceHelper Class See Also Send Feedback

The MockSequenceHelper type exposes the following members.

Methods

Name

Description

∄\$

<u>InSequence<(Of <<'(TMock>)>>)</u> Perform an expectation in the trace.

MockSequenceHelper Class Moq Namespace

_

□ C#

Moq

MockSequenceHelper..::..InSequence<(Of <(<'TMock>)>)> Method <u>MockSequenceHelper Class See Also Send Feedback</u>

Perform an expectation in the trace.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Type Parameters

TMock

MockSequenceHelper Class Moq Namespace

-

□ C#

Moq

Range Enumeration

See Also Send Feedback

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

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public enum Range

Members

Member name Description

Inclusive

Exclusive

Moq Namespace

-

□ C#

Moq

SequenceExtensions Class

Members See Also Send Feedback

Helper for sequencing return values in the same method.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public static class SequenceExtensions

Inheritance Hierarchy

System..:..Object

Moq..:..SequenceExtensions

SequenceExtensions Members Moq Namespace

□ C#	
☐ Include Protected Members	
☐ Include Inherited Members	
Moq	
SequenceExtensions Members	
SequenceExtensions Class Methods See Also Send Feedba	ıck

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

Methods

Description

Name
SetupSequence<(Of <<'(TMock, TResult>)>>)

SequenceExtensions Class Moq Namespace

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

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

Methods

Description

Name
SetupSequence<(Of <<'(TMock, TResult>)>>)

SequenceExtensions Class Moq Namespace

_

□ C#

Moq

SequenceExtensions..:..SetupSequence<(Of <(<'TMock, TResult>)>)> Method SequenceExtensions Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.SequenceExtensions.SetupSequence``2(Moq.Mock{``0},System.Linq.Expressions.Expression

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Type Parameters

TMock TResult

SequenceExtensions Class Moq Namespace

-

□ C#

Moq

Times Structure

Members See Also Send Feedback

Defines the number of invocations allowed by a mocked method.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public struct Times

Times Members Moq Namespace

□ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Times Members
<u>Times Structure Methods Operators See Also Send Feedback</u>

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

Methods

	Name	Description
∉ ∳S	AtLeast	Specifies that a mocked method should be invoked callCount times as minimum.
∈ ≬S	<u>AtLeastOnce</u>	Specifies that a mocked method should be invoked one time as minimum.
∈ ≬S	AtMost	Specifies that a mocked method should be invoked callCount time as maximun.
∉ ∳S	<u>AtMostOnce</u>	Specifies that a mocked method should be invoked one time as maximun.
∉ ∳S	<u>Between</u>	Specifies that a mocked method should be invoked between callCountFrom and callCountTo times.
=	<u>Equals</u>	Determines whether the specified <u>Object</u> is equal to this instance. (Overrides <u>ValueType:Equals(Object</u>).)
∉ ∳S	Exactly	Specifies that a mocked method should be invoked exactly callCount times.
≡	GetHashCode	Returns a hash code for this instance. (Overrides ValueType:GetHashCode()()()().)
=\$S	<u>Never</u>	Specifies that a mocked method should not be invoked.
∉ ∳S	Once	Specifies that a mocked method should be invoked exactly one time.

\blacksquare Operators

Name	Description
≝₅s <u>Equality</u>	Determines whether two specified <u>Times</u> objects have the same value.
	Determines whether two specified <u>Times</u> objects have different values.

Times Structure Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Times Methods
Times Structure See Also Send Feedback

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

Methods

	Name	Description
∉ ∳S	AtLeast	Specifies that a mocked method should be invoked callCount times as minimum.
∈ ≬S	<u>AtLeastOnce</u>	Specifies that a mocked method should be invoked one time as minimum.
∈ ≬S	AtMost	Specifies that a mocked method should be invoked callCount time as maximun.
∉ ∳S	<u>AtMostOnce</u>	Specifies that a mocked method should be invoked one time as maximun.
∉ ∳S	<u>Between</u>	Specifies that a mocked method should be invoked between callCountFrom and callCountTo times.
=	<u>Equals</u>	Determines whether the specified <u>Object</u> is equal to this instance. (Overrides <u>ValueType:Equals(Object</u>).)
∉ ∳S	Exactly	Specifies that a mocked method should be invoked exactly callCount times.
≡	GetHashCode	Returns a hash code for this instance. (Overrides ValueType:GetHashCode()()()().)
=\$S	<u>Never</u>	Specifies that a mocked method should not be invoked.
∉ ∳S	Once	Specifies that a mocked method should be invoked exactly one time.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..AtLeast Method

Times Structure See Also Send Feedback

Specifies that a mocked method should be invoked callCount times as minimum.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

Parameters

callCount

Type: <u>System..:..Int32</u>

The minimun number of times.

Return Value

An object defining the allowed number of invocations.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..AtLeastOnce Method

Times Structure See Also Send Feedback

Specifies that a mocked method should be invoked one time as minimum.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

public static <u>Times</u> AtLeastOnce()

Return Value

An object defining the allowed number of invocations.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..AtMost Method

Times Structure See Also Send Feedback

Specifies that a mocked method should be invoked callCount time as maximun.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

Parameters

callCount

Type: <u>System..:..Int32</u>

The maximun number of times.

Return Value

An object defining the allowed number of invocations.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..AtMostOnce Method

Times Structure See Also Send Feedback

Specifies that a mocked method should be invoked one time as maximun.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

public static <u>Times</u> AtMostOnce()

Return Value

An object defining the allowed number of invocations.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..Between Method

Times Structure See Also Send Feedback

Specifies that a mocked method should be invoked between callCountFrom and callCountTo times.

Namespace: Moq

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

callCountFrom

Type: <u>System..:..Int32</u>

The minimun number of times.

callCountTo

Type: System..:..Int32

The maximun number of times.

rangeKind

Type: Moq....Range

The kind of range. See Range.

Return Value

An object defining the allowed number of invocations.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..Equals Method

<u>Times Structure</u> <u>See Also Send Feedback</u>

Determines whether the specified **Object** is equal to this instance.

Namespace: Moq

Syntax

Parameters

obj

Type: <u>System..:..Object</u>

The **Object** to compare with this instance.

Return Value

true if the specified **Object** is equal to this instance; otherwise, false.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..Exactly Method

<u>Times Structure</u> <u>See Also Send Feedback</u>

Specifies that a mocked method should be invoked exactly callCount times.

Namespace: Moq

Syntax

C#

Parameters

callCount

Type: <u>System..:..Int32</u>

The times that a method or property can be called.

Return Value

An object defining the allowed number of invocations.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..GetHashCode Method

<u>Times Structure</u> <u>See Also Send Feedback</u>

Returns a hash code for this instance.

Namespace: Moq

Syntax

C#

public override int GetHashCode()

Return Value

A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..Never Method

<u>Times Structure</u> <u>See Also Send Feedback</u>

Specifies that a mocked method should not be invoked.

Namespace: Moq

Syntax

C#

public static <u>Times</u> Never()

Return Value

An object defining the allowed number of invocations.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..Once Method

<u>Times Structure</u> <u>See Also Send Feedback</u>

Specifies that a mocked method should be invoked exactly one time.

Namespace: Moq

Syntax

C#

public static <u>Times</u> Once()

Return Value

An object defining the allowed number of invocations.

Times Structure Moq Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
Times Operators
Times Structure See Also Send Feedback

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

\blacksquare Operators

Name	Description
≝₅s <u>Equality</u>	Determines whether two specified <u>Times</u> objects have the same value.
	Determines whether two specified <u>Times</u> objects have different values.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..Equality Operator

<u>Times Structure</u> <u>See Also Send Feedback</u>

Determines whether two specified <u>Times</u> objects have the same value.

Namespace: Moq

Syntax

Parameters

```
left
    Type: Moq..:..Times
    The first Times.

right
    Type: Moq..:..Times
    The second Times.
```

Return Value

true if the value of left is the same as the value of right; otherwise, false.

Times Structure Moq Namespace

-

□ C#

Moq

Times..:..Inequality Operator

<u>Times Structure</u> <u>See Also Send Feedback</u>

Determines whether two specified **Times** objects have different values.

Namespace: Moq

Syntax

```
C# public static \frac{bool}{left} operator !=( \frac{Times}{left} right )
```

Parameters

```
left
    Type: Moq..:..Times
    The first Times.

right
    Type: Moq..:..Times
    The second Times.
```

Return Value

true if the value of left is different from the value of right; otherwise, false.

Times Structure Moq Namespace

□ C#
Moq
Moq.Language Namespace
Send Feedback

This namespace defines the interfaces that contain the methods available in Moq fluent API, such as Setup, 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
o ~ O	<u>ICallback</u>	Defines the Callback verb and overloads.
⊶ 0	<pre>ICallback<(Of <(<'TMock, TResult>)>)></pre>	Defines the Callback verb and overloads for callbacks on setups that return a value.
⊶0	ICallbackGetter<(Of <(<'TMock, TProperty>)>)>	Defines the Callback verb for property getter setups.
~ 0	<pre>ICallbackSetter<(Of <(<'TProperty>)>)></pre>	Defines the Callback verb for property setter setups.
o - O	<u>IRaise<(Of <(<'T>)>)></u>	Defines the Raises verb.
~ 0	IReturns<(Of <(<'TMock, TResult>)>)>	Defines the Returns verb.
o-O	IReturnsGetter<(Of <(<'TMock, TProperty>)>)>	Defines the Returns verb for property get setups.
⊶ 0	ISetupConditionResult≤(Of ≤(≤'T≥)≥)≥	Implements the fluent API.
⊶ 0	<pre>ISetupSequentialResult<(Of <(<'TResult>)>)></pre>	Language for ReturnSequence
o ~ O	<u>IThrows</u>	Defines the Throws verb.
o - 0	<u>IVerifies</u>	Defines the Verifiable verb.

_

□ C#

Moq

ICallback Interface

Members See Also Send Feedback

Defines the Callback verb and overloads.

Namespace: Moq.Language

$\blacksquare Syntax$

C#

public interface ICallback : IHideObjectMembers

ICallback Members
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
ICallback Members

ICallback Interface Methods See Also Send Feedback

\blacksquare Methods

	Name	Description
≡	Callback(Action)	Specifies a callback to invoke when the method is called.
≅ ©	<u>Callback<(Of <<'(T>)>>)(Action<(Of <<'(T>)>>))</u>	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅ ©	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≡	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>))	Specifies a callback to invoke when the method is called that receives the original

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>))

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15,

- ▼ T16>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>>))
- Callback<(Of <<'(T1, T2>)>>)(Action<(Of <(('T1, T2>)>)>)
- <u>Callback<(Of <<'(T1, T2, T3>)>>)(Action<(Of <(('T1, T2, T3>)>)>)</u>
- Callback<(Of <<'(T1, T2, T3, T4>)>>)
 (Action<(Of <(<'T1, T2, T3, T4>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5, T6>)>)>)
 - <u>Callback<(Of <<'(T1, T2, T3, T4, T5, T6,</u>
- T7>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T6, T7, T8>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>))

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 Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
ICallback Methods

ICallback Interface See Also Send Feedback

\blacksquare Methods

	Name	Description
≡	Callback(Action)	Specifies a callback to invoke when the method is called.
≅ ©	<u>Callback<(Of <<'(T>)>>)(Action<(Of <<'(T>)>>))</u>	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅ ©	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≡	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>))	Specifies a callback to invoke when the method is called that receives the original

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>))

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15,

- ▼ T16>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>>))
- Callback<(Of <<'(T1, T2>)>>)(Action<(Of <(<'T1, T2>)>)>)
- <u>Callback<(Of <<'(T1, T2, T3>)>>)(Action<(Of <(('T1, T2, T3>)>)>)</u>
- Callback<(Of <<'(T1, T2, T3, T4>)>>)
 (Action<(Of <(<'T1, T2, T3, T4>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5, T6>)>)>)
 - <u>Callback<(Of <<'(T1, T2, T3, T4, T5, T6,</u>
- T7>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T6, T7, T8>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>))

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 Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq

ICallback..:..Callback Method

ICallback Interface See Also Send Feedback

Overload List

	Name	Description
≡	Callback(Action)	Specifies a callback to invoke when the method is called.
=	<u>Callback<(Of <<'(T>)>>)(Action<(Of <<'(T>)>>))</u>	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
=	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
=	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≟	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
=	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>))	Specifies a callback to invoke when the method is called that receives the original

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>))

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15,

- ▼ T16>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>>))
- Callback<(Of <<'(T1, T2>)>>)(Action<(Of <(<'T1, T2>)>)>)
- <u>Callback<(Of <<'(T1, T2, T3>)>>)(Action<(Of <(('T1, T2, T3>)>)>)</u>
- Callback<(Of <<'(T1, T2, T3, T4>)>>)
 (Action<(Of <(<'T1, T2, T3, T4>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5, T6>)>)>)
 - <u>Callback<(Of <<'(T1, T2, T3, T4, T5, T6,</u>
- T7>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T6, T7, T8>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>))

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

-

□ C#

Moq

ICallback..:..Callback Method (Action)

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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

```
ICallbackResult Callback(
         <u>Action</u> action
)
```

Parameters

action

Type: System..:..Action
The callback method to invoke.

Examples

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

C# □ Copy

```
var called = false;
mock.Setup(x => x.Execute())
    .Callback(() => called = true);
```

ICallback Interface
Callback Overload
Moq.Language Namespace

-

□ C#

Moq

ICallback..:..Callback<(Of <(<'T>)>)> Method (Action<(Of <(<'T>)>)>)

ICallback Interface Example See Also Send Feedback

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

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T>)>)> The callback method to invoke.

Type Parameters

T

The 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# Copy

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

ICallback Interface
Callback Overload
Moq.Language Namespace

⊟ □ C#

Moq

ICallback..::..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>)>) 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

```
Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>)>
```

The callback method to invoke.

Type Parameters

T1	
Т2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
Т3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
Т6	The type of the sixth argument of the invoked method.
T7	
Т8	The type of the seventh argument of the invoked method.
Т9	The type of the eighth argument of the invoked method.
T10	The type of the nineth argument of the invoked method.
110	The type of the tenth argument of the invoked method.

Return Value

A reference to ICallbackResult interface.

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

⊟ □ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>)>)

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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>)>

The callback method to invoke.

Type Parameters

T1	
	The type of the first argument of the invoked method.
T2	The type of the second argument of the invoked method.
T3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
T6	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	The type of the eleventh argument of the invoked method.

Return Value

A reference to ICallbackResult interface.

Examples

C#

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

It.IsAny<string>(),
It.IsAny<string>(),

It.IsAny<string>(),
It.IsAny<string>(),
It.IsAny<string>(),
It.IsAny<string>(),
It.IsAny<string>(),
It.IsAny<string>(),

.Callback((string arg1, string arg2, string arg3, string arg4, s

ICallback Interface
Callback Overload
Moq.Language Namespace

-

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>)>)

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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>)>

The callback method to invoke.

Type Parameters

T1	
	The type of the first argument of the invoked method.
T2	The type of the second argument of the invoked method.
T3	
T4	The type of the third argument of the invoked method.
	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
T6	51
T7	The type of the sixth argument of the invoked method.
	The type of the seventh argument of the invoked method.
T8	The type of the eighth argument of the invoked method.
T9	The type of the eighth digunient of the hivoked method.
т10	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	
T12	The type of the eleventh argument of the invoked method.
-	The type of the twelfth argument of the invoked method.

Return Value

A reference to ICallbackResult interface.

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

_

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>)>)

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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>)>

The callback method to invoke.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
Т3	The type of the third argument of the invoked method.
T4	U
T5	The type of the fourth argument of the invoked method.
T6	The type of the fifth argument of the invoked method.
	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
T8	The type of the eighth argument of the invoked method.
Т9	
T10	The type of the nineth argument of the invoked method.
T11	The type of the tenth argument of the invoked method.
	The type of the eleventh argument of the invoked method.
T12	The type of the twelfth argument of the invoked method.
T13	The type of the thirteenth argument of the invoked method
	y F

Return Value

A reference to ICallbackResult interface.

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

_

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>)>)

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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>)>

The callback method to invoke.

■Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
T3	The type of the third argument of the invoked method.
T4	
T5	The type of the fourth argument of the invoked method.
Т6	The type of the fifth argument of the invoked method.
	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
T8	
Т9	The type of the eighth argument of the invoked method.
T10	The type of the nineth argument of the invoked method.
	The type of the tenth argument of the invoked method.
T11	The type of the eleventh argument of the invoked method.
T12	
T13	The type of the twelfth argument of the invoked method.
T14	The type of the thirteenth argument of the invoked method.
114	The type of the fourteenth argument of the invoked method

Return Value

A reference to ICallbackResult interface.

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

```
mock.Setup(x => x.Execute(
                     It.IsAny<string>(),
                     It.IsAny<string>()))
    .Callback((string arg1, string arg2, string arg3, string arg4, s
```

ICallback Interface
Callback Overload
Moq.Language Namespace

_

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>)>)

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

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>)> The callback method to invoke.

■Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
Т3	The type of the third argument of the invoked method.
T4	
T5	The type of the fourth argument of the invoked method.
T6	The type of the fifth argument of the invoked method.
	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	
T10	The type of the nineth argument of the invoked method.
T11	The type of the tenth argument of the invoked method.
	The type of the eleventh argument of the invoked method.
T12	The type of the twelfth argument of the invoked method.
T13	
T14	The type of the thirteenth argument of the invoked method.
T15	The type of the fourteenth argument of the invoked method.
110	The type of the fifteenth argument of the invoked method.

Return Value

A reference to ICallbackResult interface.

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

```
mock.Setup(x => x.Execute(
                     It.IsAny<string>(),
                     It.IsAny<string>()))
    .Callback((string arg1, string arg2, string arg3, string arg4, s
```

ICallback Interface
Callback Overload
Moq.Language Namespace

_

□ C#

Moq

ICallback..::..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>)>)

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

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>)>
The callback method to invoke.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
T3 T4	The type of the third argument of the invoked method.
т 4 Т5	The type of the fourth argument of the invoked method.
15 T6	The type of the fifth argument of the invoked method.
16 T7	The type of the sixth argument of the invoked method.
	The type of the seventh argument of the invoked method.
T8	The type of the eighth argument of the invoked method.
T9	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	The type of the eleventh argument of the invoked method.
T12	The type of the twelfth argument of the invoked method.
T13	The type of the thirteenth argument of the invoked method.
T14	The type of the fourteenth argument of the invoked method.
T15	The type of the fifteenth argument of the invoked method.
T16	The type of the sixteenth argument of the invoked method.

Return Value

A reference to ICallbackResult interface.

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

```
mock.Setup(x => x.Execute(
                     It.IsAny<string>(),
                     It.IsAny<string>()))
    .Callback((string arg1, string arg2, string arg3, string arg4, s
```

ICallback Interface
Callback Overload
Moq.Language Namespace

```
-
```

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2>)>)> Method (Action)

ICallback Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback.Callback``2(System.Action`2)"]

Namespace: Moq.Language

Syntax

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2>)>)></u>

Type Parameters

T1 T2

ICallback Interface
Callback Overload
Moq.Language Namespace

```
-
```

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3>)>)> Method (Action)

ICallback Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback.Callback``3(System.Action`3)"]

Namespace: Moq.Language

\blacksquare Syntax

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3>)>)></u>

Type Parameters

T1

T2

T3

ICallback Interface
Callback Overload
Moq.Language Namespace

-

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4>)>)> Method (Action)

ICallback Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback.Callback``4(System.Action`4)"]

Namespace: Moq.Language

Syntax

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3, T4>)>)></u>

Type Parameters

T1

T2

T3

T4

ICallback Interface
Callback Overload
Moq.Language Namespace

-

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5>)>)> Method (Action)

ICallback Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback.Callback``5(System.Action`5)"]

Namespace: Moq.Language

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(<u>Of</u> <(<u><</u>'<u>T1, T2, T3, T4, T5</u>>)>)>

Type Parameters

T1

T2

Т3

T4

T5

ICallback Interface
Callback Overload
Moq.Language Namespace

-

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6>)>)> Method (Action)

ICallback Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback.Callback``6(System.Action`6)"]

Namespace: Moq.Language

Syntax

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3, T4, T5, T6>)>)></u>

Type Parameters

T1

T2

T3

T4

T5

T6

ICallback Interface
Callback Overload
Moq.Language Namespace

-

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)> Method (Action)

ICallback Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback.Callback``7(System.Action`7)"]

Namespace: Moq.Language

Syntax

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)></u>

Type Parameters

T1

T2

Т3

T4

T5

T6

T7

ICallback Interface
Callback Overload
Moq.Language Namespace

-

□ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8>)>)> Method (Action)

ICallback Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback.Callback``8(System.Action`8)"]

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8>)>)></u>

Type Parameters

T1

T2

T3

T4

T5

T6

T7

T8

ICallback Interface
Callback Overload
Moq.Language Namespace

⊟ □ C#

Moq

ICallback..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9>)>)>)

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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9>)>)> The callback method to invoke.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
Т3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	
Т6	The type of the fifth argument of the invoked method.
T7	The type of the sixth argument of the invoked method.
	The type of the seventh argument of the invoked method.
T8	The type of the eighth argument of the invoked method.
T9	
	The type of the nineth argument of the invoked method.

Return Value

A reference to ICallbackResult interface.

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

-

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)> Interface

Members See Also Send Feedback

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

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

 $\hbox{public interface ICallback$<$TMock, TResult>: IHideObjectMembers where TMock: class}$

Type Parameters

TMock

Mocked type.

TResult

Type of the return value of the setup.

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

□ C#
□ Include Protected Members
□ Include Inherited Members

Moq
ICallback<(Of <(<'TMock, TResult>)>)> Members

ICallback<(Of <(<'TMock, TResult>)>)> Interface Methods See Also Send Feedback

\blacksquare Methods

	Name	Description
≡	Callback(Action)	Specifies a callback to invoke when the method is called.
≅ ©	<u>Callback<(Of <<'(T>)>>)(Action<(Of <<'(T>)>>))</u>	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅ ©	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≡	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>))	Specifies a callback to invoke when the method is called that receives the original

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>))

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15,

- ▼ T16>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>>))
- Callback<(Of <<'(T1, T2>)>>)(Action<(Of <(('T1, T2>)>)>)
- <u>Callback<(Of <<'(T1, T2, T3>)>>)(Action<(Of <(('T1, T2, T3>)>)>)</u>
- Callback<(Of <<'(T1, T2, T3, T4>)>>)
 (Action<(Of <(<'T1, T2, T3, T4>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5, T6>)>)>)
 - <u>Callback<(Of <<'(T1, T2, T3, T4, T5, T6,</u>
- T7>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T6, T7, T8>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>))

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 <(<'TMock, TResult>)>)> Interface
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ICallback<(Of <(<'TMock, TResult>)>)> Methods

ICallback<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

\blacksquare Methods

	Name	Description
≡	Callback(Action)	Specifies a callback to invoke when the method is called.
≅ ©	<u>Callback<(Of <<'(T>)>>)(Action<(Of <<'(T>)>>))</u>	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)))	Specifies a callback to invoke when the method is called that receives the original arguments.
≅ ©	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≡	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>))	Specifies a callback to invoke when the method is called that receives the original

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>))

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15,

- ▼ T16>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>>))
- Callback<(Of <<'(T1, T2>)>>)(Action<(Of <(('T1, T2>)>)>)
- <u>Callback<(Of <<'(T1, T2, T3>)>>)(Action<(Of <(('T1, T2, T3>)>)>)</u>
- Callback<(Of <<'(T1, T2, T3, T4>)>>)
 (Action<(Of <(<'T1, T2, T3, T4>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5, T6>)>)>)
 - <u>Callback<(Of <<'(T1, T2, T3, T4, T5, T6,</u>
- T7>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T6, T7, T8>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>))

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 <(<'TMock, TResult>)>)> Interface
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

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

Overload List

	Name	Description
≡	Callback(Action)	Specifies a callback to invoke when the method is called.
=	<u>Callback<(Of <<'(T>)>>)(Action<(Of <<'(T>)>>))</u>	Specifies a callback to invoke when the method is called that receives the original arguments.
≅	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
=	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
=	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
≟	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>))	Specifies a callback to invoke when the method is called that receives the original arguments.
=	Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>))	Specifies a callback to invoke when the method is called that receives the original

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>) (Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>))

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15,

- ▼ T16>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>>))
- Callback<(Of <<'(T1, T2>)>>)(Action<(Of <(('T1, T2>)>)>)
- <u>Callback<(Of <<'(T1, T2, T3>)>>)(Action<(Of <(('T1, T2, T3>)>)>)</u>
- Callback<(Of <<'(T1, T2, T3, T4>)>>)
 (Action<(Of <(<'T1, T2, T3, T4>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6>)>>)
 (Action<(Of <(<'T1, T2, T3, T4, T5, T6>)>)>)
 - <u>Callback<(Of <<'(T1, T2, T3, T4, T5, T6,</u>
- T7>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8>)>>)(Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T6, T7, T8>)>)>)
- Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Action<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>))

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

-

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..:..Callback Method (Action)

ICallback<(Of <(<'TMock, 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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

```
IReturnsThrows<TMock, TResult> Callback(
         <u>Action</u> action
)
```

Parameters

action

Type: System..:..Action
The callback method to invoke.

Examples

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

C#

```
var called = false;
mock.Setup(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.

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

_

□ C#

Moq

ICallback<(Of<(<'TMock, TResult>)>)>.....Callback<(Of<(<'T>)>)> Method (Action<(Of<(<'T>)>)>)

<u>ICallback<(Of <(<'TMock, TResult>)>)> 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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T>)>)> Callback method to invoke.

Type Parameters

T

The 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# □ Copy

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

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

_

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>.::..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>)>)

<u>ICallback<(Of <(<'TMock, TResult>)>)> 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>)>

The callback method to invoke.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
Т3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
Т6	
T7	The type of the sixth argument of the invoked method.
Т8	The type of the seventh argument of the invoked method
T9	The type of the eighth argument of the invoked method.
	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.

Return Value

A reference to IReturnsThrows interface.

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

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

_

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>.::..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>)>)

<u>ICallback<(Of <(<'TMock, TResult>)>)> 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>)>

The callback method to invoke.

Type Parameters

T1	
T a	The type of the first argument of the invoked method.
T2	The type of the second argument of the invoked method.
T3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
Т6	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	The type of the nineth argument of the invoked method.
T10	.
T11	The type of the tenth argument of the invoked method.
	The type of the eleventh argument of the invoked method.

Return Value

A reference to IReturnsThrows interface.

Examples

Invokes the given callback with the concrete invocation arguments values.

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

C#

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

_

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>.::..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>)>)

<u>ICallback<(Of <(<'TMock, TResult>)>)> 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>)>

The callback method to invoke.

Type Parameters

T1	
TO	The type of the first argument of the invoked method.
T2	The type of the second argument of the invoked method.
T3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
Т6	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	The type of the eleventh argument of the invoked method.
T12	
	The type of the twelfth argument of the invoked method.

Return Value

A reference to IReturnsThrows interface.

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

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

_

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>.::..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>)>)

<u>ICallback<(Of <(<'TMock, TResult>)>)> 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>)>

The callback method to invoke.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
T3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	
T6	The type of the fifth argument of the invoked method.
T7	The type of the sixth argument of the invoked method.
Т8	The type of the seventh argument of the invoked method.
	The type of the eighth argument of the invoked method.
T9	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	
T12	The type of the eleventh argument of the invoked method.
T13	The type of the twelfth argument of the invoked method.
	The type of the thirteenth argument of the invoked method

Return Value

A reference to IReturnsThrows interface.

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

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

-

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..::..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>)>)
ICallback<(Of <(<'TMock, 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>)>

The callback method to invoke.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
Т3	The type of the third argument of the invoked method.
T4	
T5	The type of the fourth argument of the invoked method.
TC	The type of the fifth argument of the invoked method.
T6	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
T8	
Т9	The type of the eighth argument of the invoked method.
Т10	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	The type of the eleventh argument of the invoked method.
T12	.
T13	The type of the twelfth argument of the invoked method.
	The type of the thirteenth argument of the invoked method.
T14	The type of the fourteenth argument of the invoked method

Return Value

A reference to IReturnsThrows interface.

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

```
mock.Setup(x => x.Execute(
                     It.IsAny<string>(),
                     It.IsAny<string>()))
    .Callback((arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9,
```

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

_

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..::..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>)>) ICallback<(Of <(<'TMock, 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>)> The callback method to invoke.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
Т3	The type of the third argument of the invoked method.
T4	
T5	The type of the fourth argument of the invoked method.
T6	The type of the fifth argument of the invoked method.
	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	
T10	The type of the nineth argument of the invoked method.
T11	The type of the tenth argument of the invoked method.
	The type of the eleventh argument of the invoked method.
T12	The type of the twelfth argument of the invoked method.
T13	
T14	The type of the thirteenth argument of the invoked method.
T15	The type of the fourteenth argument of the invoked method.
110	The type of the fifteenth argument of the invoked method.

Return Value

A reference to IReturnsThrows interface.

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

```
mock.Setup(x => x.Execute(
                     It.IsAny<string>(),
                     It.IsAny<string>()))
    .Callback((arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9,
```

See Also

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

Send comments on this topic to moqdisc@googlegroups.com

_

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>)>)

<u>ICallback<(Of <(<'TMock, TResult>)>)> 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>)>
The callback method to invoke.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
Т3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
Т6	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	The type of the eleventh argument of the invoked method.
T12	
T13	The type of the twelfth argument of the invoked method.
T14	The type of the thirteenth argument of the invoked method.
T15	The type of the fourteenth argument of the invoked method.
T16	The type of the fifteenth argument of the invoked method.
	The type of the sixteenth argument of the invoked method.

Return Value

A reference to IReturnsThrows interface.

Examples

Invokes the given callback with the concrete invocation arguments values.

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

C#

```
mock.Setup(x => x.Execute(
                     It.IsAny<string>(),
                     It.IsAny<string>()))
    .Callback((arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9,
```

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

-

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..::..Callback<(Of <(<'T1, T2>)>)> Method (Action)

ICallback<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback`2.Callback`2(System.Action`2)"]

Namespace: Moq.Language

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2>)>)></u>

T1 T2

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

```
□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..:..Callback<(Of <(<'T1, T2, T3>)>)> Method (Action)

ICallback<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback
```

[Missing <summary> documentation for "M:Moq.Language.ICallback`2.Callback`3(System.Action`3)"]

Namespace: Moq.Language

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3>)>)></u>

T1

T2

T3

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

⊟ □ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..:..Callback<(Of <(<'T1, T2, T3, T4>)>)> Method (Action)

ICallback<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback`2.Callback`4(System.Action`4)"]

Namespace: Moq.Language

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3, T4>)>)></u>

T1

T2

T3

T4

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

-

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..:..Callback<(Of <(<'T1, T2, T3, T4, T5>)>)> Method (Action)

ICallback<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback`2.Callback`5(System.Action`5)"]

Namespace: Moq.Language

C#

Parameters

action

Type: <u>System..:..Action</u><(<u>Of</u> <(<u><'T1, T2, T3, T4, T5</u>>)>)>

T1

T2

Т3

T4

T5

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

⊟ □ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6>)>)> Method (Action)

ICallback<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback`2.Callback`6(System.Action`6)"]

Namespace: Moq.Language

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3, T4, T5, T6>)>)></u>

T1

T2

T3

T4

T5

T6

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

⊟ □ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)> Method (Action)

ICallback<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback`2.Callback`7(System.Action`7)"]

Namespace: Moq.Language

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)></u>

T1

T2

T3

T4

T5

T6

T7

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

⊟ □ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8>)>)> Method (Action)

ICallback<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.ICallback`2.Callback`8(System.Action`8)"]

Namespace: Moq.Language

C#

Parameters

action

Type: <u>System..:..Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8>)>)></u>

T1

T2

T3

T4

T5

T6

T7

T8

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

_

□ C#

Moq

ICallback<(Of <(<'TMock, TResult>)>)>..:..Callback<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9>)>)> Method (Action<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9>)>)>)

<u>ICallback<(Of <(<'TMock, TResult>)>)> 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

C#

Parameters

action

Type: <u>System..:..Action</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9>)>)> The callback method to invoke.

T1 The type of the first argument of the invoked method. T2 The type of the second argument of the invoked method. T3 The type of the third argument of the invoked method. T4 The type of the fourth argument of the invoked method. T5 The type of the fifth argument of the invoked method. T6 The type of the sixth argument of the invoked method. T7 The type of the seventh argument of the invoked method. T8 The type of the eighth argument of the invoked method. T9 The type of the nineth argument of the invoked method.

Return Value

A reference to IReturnsThrows interface.

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<(Of <(<'TMock, TResult>)>)> Interface
Callback Overload
Moq.Language Namespace

-

□ C#

Moq

ICallbackGetter<(Of <(<'TMock, TProperty>)>)> Interface Members See Also Send Feedback

Defines the Callback verb for property getter setups.

Namespace: Moq.Language

C#

public interface ICallbackGetter<TMock, TProperty> : IHideObjectMemk
where TMock : class

Type Parameters

TMock
Mocked type.
TProperty
Type of the property.

ICallbackGetter<(Of <(<'TMock, TProperty>)>)> Members
Moq.Language Namespace
MockSetupGet``1(Expression<(Of <<'(Func<(Of <<'(UTP, UMP>)>>)>))

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ICallbackGetter<(Of <(<'TMock, TProperty>)>)> Members

ICallbackGetter<(Of <(<'TMock, TProperty>)>)>
Interface Methods See Also Send Feedback
```

The $\underline{ICallbackGetter} < (\underline{Of} < (\underline{<'TMock, TProperty}) >) > type exposes the following members.$

Methods

Name

Description

• Callback Specifies a callback to invoke when the property is retrieved.

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

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ICallbackGetter<(Of <(<'TMock, TProperty>)>)> Methods

ICallbackGetter<(Of <(<'TMock, TProperty>)>)> Interface See Also Send
Feedback
```

The $\underline{ICallbackGetter} < (\underline{Of} < (\underline{<'TMock, TProperty}) >) > type exposes the following members.$

Methods

Name

Description

• Callback Specifies a callback to invoke when the property is retrieved.

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

_

□ C#

Moq

ICallbackGetter<(Of <(<'TMock, TProperty>)>)>.....Callback Method

ICallbackGetter<(Of <(<'TMock, 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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

Parameters

action

Type: <u>System..:..Action</u> Callback method to invoke.

Examples

Invokes the given callback with the property value being set.

C#

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

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

-

□ C#

Moq

ICallbackSetter<(Of <(<'TProperty>)>)> Interface

Members See Also Send Feedback

Defines the Callback verb for property setter setups.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

 $\verb"public interface ICallbackSetter<TProperty> : IHideObjectMembers"$

Type Parameters

TProperty
Type of the property.

ICallbackSetter<(Of <(<'TProperty>)>)> Members
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ICallbackSetter<(Of <(<'TProperty>)>)> Members

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

The $\underline{ICallbackSetter} \le (\underline{Of} \le (\le \underline{TProperty} \ge) \ge)$ 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.

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

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ICallbackSetter<(Of <(<'TProperty>)>)> Methods

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

The $\underline{ICallbackSetter} \le (\underline{Of} \le (\le \underline{TProperty} \ge) \ge)$ 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.

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

-

□ 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

action

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

Examples

Invokes the given callback with the property value being set.

C#

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

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

-

□ C#

Moq

IRaise<(Of <(<'T>)>)> Interface

Members See Also Send Feedback

Defines the Raises verb.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public interface IRaise<T> : IHideObjectMembers

Type Parameters

T

IRaise<(Of <(<'T>)>)> Members Moq.Language Namespace

-

□ C#

☐ Include Protected Members

□ Include Inherited Members

Moq

IRaise<(Of <(<'T>)>)> Members

IRaise<(Of <(<'T>)>)> Interface Methods See Also Send Feedback

\blacksquare Methods

	Name	Description
≟ ∳	Raises(Action \leq (Of $\leq\leq'$ (T \geq) \geq), EventArgs)	Specifies the event that will be raised when the setup is met.
=0	<u>Raises(Action<(Of <<'(T>)>>), Func<(Of <<('TResult>)>)>)</u>	
	Raises(Action $<$ (Of $<<$ '(T $>$) $>>$), array $<$ Object $>$ []()[]	Specifies the custom event that will be raised when the setup is matched.
	Raises<(Of <<'(T1>)>>)(Action<(Of <<'(T>)>>), Func<(Of <(<'T, TResult>)>)>)	
	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≓	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≘ ∳	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≓	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>)(Action<(Of	Specifies the

event that will be raised when the setup is matched.

T10, T11, T12, T13, T14, T15>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, EventArgs>)>>))

Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9,

Specifies the event that will be raised when the setup is matched.

Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>>)(Action<(Of

Specifies the event that will be raised when the setup is matched.

- Raises<(Of <<'(T1, T2>)>>)(Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, TResult>)>)>)
- Raises<(Of <<'(T1, T2, T3>)>>)(Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, TResult>)>))
 Raises<(Of <<'(T1, T2, T3, T4>)>>)(Action<(Of
- - Raises < (Of < < '(T1, T2, T3, T4, T5 >) > >) (Action < (Of
- - Raises<(Of <<'(T1, T2, T3, T4, T5, T6>)>>)
- (Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, T5, T6, TResult>)>)>)
 - Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7>)>>)
- (Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, TResult>)>)>)
 - Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8>)>>)
- (Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, TResult>)>)>)

Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, EventArgs>)>>)) Specifies the event that will be raised when the setup is matched.

IRaise<(Of <(<'T>)>)> Interface
Moq.Language Namespace

□ C#

- ☐ Include Protected Members
- □ Include Inherited Members

Moq

 $IRaise < (Of < (< \top T>)>) > Methods$

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

\blacksquare Methods

	Name	Description
≟ ∳	Raises(Action \leq (Of $\leq\leq'$ (T \geq) \geq), EventArgs)	Specifies the event that will be raised when the setup is met.
=0	<u>Raises(Action<(Of <<'(T>)>>), Func<(Of <<('TResult>)>)>)</u>	
	Raises(Action $<$ (Of $<<$ '(T $>$) $>>$), array $<$ Object $>$ []()[]	Specifies the custom event that will be raised when the setup is matched.
	Raises<(Of <<'(T1>)>>)(Action<(Of <<'(T>)>>), Func<(Of <(<'T, TResult>)>)>)	
	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≓	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≘ ∳	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≓	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>)(Action<(Of	Specifies the

event that will be raised when the setup is matched.

T10, T11, T12, T13, T14, T15>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, EventArgs>)>>))

Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9,

Specifies the event that will be raised when the setup is matched.

Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>>)(Action<(Of

Specifies the event that will be raised when the setup is matched.

- Raises<(Of <<'(T1, T2>)>>)(Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, TResult>)>)
- Raises<(Of <<'(T1, T2, T3>)>>)(Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, TResult>)>))
 Raises<(Of <<'(T1, T2, T3, T4>)>>)(Action<(Of
- - Raises<(Of <<'(T1, T2, T3, T4, T5>)>>)(Action<(Of
- - Raises<(Of <<'(T1, T2, T3, T4, T5, T6>)>>)
- (Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, T5, T6, TResult>)>)>)
 - Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7>)>>)
- (Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, TResult>)>)>)
 - Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8>)>>)
- (Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, TResult>)>)>)

Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, EventArgs>)>>))

Specifies the event that will be raised when the setup is matched.

IRaise<(Of <(<'T>)>)> Interface
Moq.Language Namespace

☐ C#☐ Include Protected Members

□ Include Inherited Members

Moq

 $IRaise < (Of < (< \top >) >) >Raises Method$

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Overload List

	Name	Description
≅	Raises(Action<(Of <<'(T>)>>), EventArgs)	Specifies the event that will be raised when the setup is met.
=0	<u>Raises(Action<(Of <<'(T>)>>), Func<(Of <((</u>	
≟	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≡	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≅	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≟	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
≡	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.
∃	Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>)(Action<(Of <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, EventArgs>)>>))	Specifies the event that will be raised when the setup is matched.

Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, Specifies the T10, T11, T12, T13, T14, T15>)>>)(Action<(Of event that will be <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, raised when the T7, T8, T9, T10, T11, T12, T13, T14, T15, setup is matched. EventArgs>)>>)) Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, Specifies the T10, T11, T12, T13, T14, T15, T16>)>>)(Action<(Of event that will be <<'(T>)>>), Func<(Of <<'(T1, T2, T3, T4, T5, T6, raised when the T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, setup is matched. EventArgs>)>>)) Raises<(Of <<'(T1>)>>)(Action<(Of <<'(T>)>>), <u>Func<(Of <(<'T, TResult>)>)>)</u> Raises<(Of <<'(T1, T2>)>>)(Action<(Of <<'(T>)>>), <u>Func<(Of <(<'T1, T2, TResult>)>)></u>) Raises<(Of <<'(T1, T2, T3>)>>)(Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, TResult>)>)>) Raises<(Of <<'(T1, T2, T3, T4>)>>)(Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, ≣📦 <u>TResult>)>)>)</u> Raises<(Of <<'(T1, T2, T3, T4, T5>)>>)(Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, T5, <u>TResult>)>)>)</u> Raises<(Of <<'(T1, T2, T3, T4, T5, T6>)>>) (Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, T5, T6, TResult>)>)>) Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7>)>>) (Action<(Of <<'(T>)>>), Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, TResult>)>)>) Raises<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8>)>>) (Action < (Of << '(T>)>>), Func < (Of < (< 'T1, T2, T3, T4, T5, T6, T7, T8, TResult>)>)>)

Raises(Action<(Of <<'(T>)>>), array<Object>[]()[]

Specifies the custom event that will be raised when the setup is matched.

IRaise<(Of <(<'T>)>)> Interface
IRaise<(Of <(<'T>)>)> Members
Moq.Language Namespace

_

□ C#

Moq

IRaise<(Of <(<'T>)>)>..:..Raises Method (Action<(Of <(<'T>)>)>, EventArgs)

IRaise<(Of <(<'T>)>)> Interface Example See Also Send Feedback

Specifies the event that will be raised when the setup is met.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

```
C#
```

Parameters

```
event Expression\\
```

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

An expression that represents an event attach or detach action.

args

Type: System..:..EventArgs

The event arguments to pass for the raised event.

Examples

The following example shows how to raise an event when the setup is met:

C#

```
var mock = new Mock<IContainer>();
mock.Setup(add => add.Add(It.IsAny<string>(), It.IsAny<object>()))
    .Raises(add => add.Added += null, EventArgs.Empty);
```

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

```
□ C#

Moq

IRaise<(Of <(<'T>)>)>..::.Raises Method (Action<(Of <(<'T>)>)>, Func)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback
```

Namespace: Moq.Language

[Missing <summary> documentation for

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

"M:Moq.Language.IRaise`1.Raises(System.Action{`0},System.Func`1)"]

Parameters

```
eventExpression
```

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

func

Type: <u>System..:..Func<(Of <(<'TResult>)>)></u>

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

```
⊟
□ C#
```

Moq

IRaise<(Of <(<'T>)>)>..::..Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9>)>)> Method (Action<(Of <(<'T>)>)>, Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, EventArgs>)>)>)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Specifies the event that will be raised when the setup is matched.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

C#

Parameters

```
eventExpression
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

The expression that represents an event attach or detach action.

func

```
Type: <u>System..:..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, <u>EventArgs</u>>)>)>
```

The function that will build the **EventArgs** to pass when raising the event.

Type Parameters

T1	
T2	The type of the first argument received by the expected invocation.
	The type of the second argument received by the expected invocation.
Т3	The type of the third argument received by the expected invocation.
T4	The type of the fourth argument received by the expected invocation.
T5	The type of the fifth argument received by the expected invocation.
Т6	
T7	The type of the sixth argument received by the expected invocation.
Т8	The type of the seventh argument received by the expected invocation.
Т9	The type of the eighth argument received by the expected invocation.
10	The type of the nineth argument received by the expected invocation.

```
IRaise<(Of <(<'T>)>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <(<'T>)>)>..::..Raises(Action<(Of <<'(T>)>>), EventArgs)
```

⊟ □ C#

Moq

IRaise<(Of <(<'T>)>)>..:..Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>)> Method (Action<(Of <(<'T>)>)>, Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, EventArgs>)>)>)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Specifies the event that will be raised when the setup is matched.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

C#

Parameters

```
event Expression\\
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

The expression that represents an event attach or detach action.

func

```
Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, <u>EventArgs</u>>)>)>
```

The function that will build the **EventArgs** to pass when raising the event.

Type Parameters

T1	
T2	The type of the first argument received by the expected invocation.
	The type of the second argument received by the expected invocation.
T3	The type of the third argument received by the expected invocation.
T4	The type of the fourth argument received by the expected invocation.
T5	The type of the fifth argument received by the expected invocation.
Т6	The type of the sixth argument received by the expected invocation.
T7	The type of the seventh argument received by the expected invocation
T8	The type of the eighth argument received by the expected invocation.
T9	The type of the nineth argument received by the expected invocation.
T10	The type of the tenth argument received by the expected invocation.

```
IRaise<(Of <(<'T>)>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <(<'T>)>)>..::..Raises(Action<(Of <<'(T>)>>), EventArgs)
```

⊟ □ C#

Moq

IRaise<(Of <(<'T>)>)>..::..Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>)> Method (Action<(Of <(<'T>)>)>, Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, EventArgs>)>)>)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Specifies the event that will be raised when the setup is matched.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

C#

Parameters

```
eventExpression
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

The expression that represents an event attach or detach action.

func

```
Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, <u>EventArgs</u>>)>)>
```

The function that will build the **EventArgs** to pass when raising the event.

Type Parameters

T1	
	The type of the first argument received by the expected invocation.
T2	The type of the second argument received by the expected invocation.
Т3	The type of the second argument received by the expected invocation.
 .	The type of the third argument received by the expected invocation.
T4	The type of the fourth argument received by the expected invocation.
T5	The type of the fourth argument received by the expected invocation.
TTC.	The type of the fifth argument received by the expected invocation.
T6	The type of the sixth argument received by the expected invocation.
T7	The type of the shift argument received by the empected invocation.
то	The type of the seventh argument received by the expected invocation.
T8	The type of the eighth argument received by the expected invocation.
Т9	
T10	The type of the nineth argument received by the expected invocation.
110	The type of the tenth argument received by the expected invocation.
T11	
	The type of the eleventh argument received by the expected invocation

```
IRaise<(Of <(<'T>)>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <(<'T>)>)>..::..Raises(Action<(Of <<'(T>)>>), EventArgs)
```

⊟ □ C#

Moq

IRaise<(Of <(<'T>)>)>..:..Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>)> Method (Action<(Of <(<'T>)>)>, Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, EventArgs>)>)>)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Specifies the event that will be raised when the setup is matched.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

C#

Parameters

```
eventExpression
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

The expression that represents an event attach or detach action.

func

```
Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, <u>EventArgs</u>>)>)>
```

The function that will build the **EventArgs** to pass when raising the event.

Type Parameters

T1	
T2	The type of the first argument received by the expected invocation.
	The type of the second argument received by the expected invocation.
T3	The type of the third argument received by the expected invocation.
T4	The type of the fourth argument received by the expected invocation.
T5	The type of the fifth argument received by the expected invocation.
T6	The type of the sixth argument received by the expected invocation.
T7	The type of the seventh argument received by the expected invocation.
T8	The type of the eighth argument received by the expected invocation.
T9	
T10	The type of the nineth argument received by the expected invocation.
T11	The type of the tenth argument received by the expected invocation.
T12	The type of the eleventh argument received by the expected invocation
	The type of the twelfth argument received by the expected invocation.

```
IRaise<(Of <(<'T>)>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <(<'T>)>)>..::..Raises(Action<(Of <<'(T>)>>), EventArgs)
```

⊟ □ C#

Moq

IRaise<(Of <(<'T>)>)>..::..Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>)> Method (Action<(Of <(<'T>)>)>, Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, EventArgs>)>)>)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Specifies the event that will be raised when the setup is matched.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

C#

Parameters

```
event Expression\\
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

The expression that represents an event attach or detach action.

func

```
Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, <u>EventArgs</u>>)>)>
```

The function that will build the **EventArgs** to pass when raising the event.

Type Parameters

T1	
T2	The type of the first argument received by the expected invocation.
T3	The type of the second argument received by the expected invocation.
	The type of the third argument received by the expected invocation.
T4	The type of the fourth argument received by the expected invocation.
T5	The type of the fifth argument received by the expected invocation.
T6	The type of the sixth argument received by the expected invocation.
T7	The type of the seventh argument received by the expected invocation.
T8	The type of the eighth argument received by the expected invocation.
T9	The type of the nineth argument received by the expected invocation.
T10	
T11	The type of the tenth argument received by the expected invocation.
T12	The type of the eleventh argument received by the expected invocation.
	The type of the twelfth argument received by the expected invocation.
T13	The type of the thirteenth argument received by the expected invocation

```
IRaise<(Of <(<'T>)>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <(<'T>)>)>..::..Raises(Action<(Of <<'(T>)>>), EventArgs)
```

_

□ C#

Moq

IRaise<(Of <(<'T>)>)>..::..Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>)> Method (Action<(Of <(<'T>)>)>, Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, EventArgs>)>)>)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Specifies the event that will be raised when the setup is matched.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

C#

Parameters

```
eventExpression
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

The expression that represents an event attach or detach action.

func

```
Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, <u>EventArgs</u>>)>)>
```

The function that will build the **EventArgs** to pass when raising the event.

Type Parameters

T1	The type of the first argument received by the expected investign
T2	The type of the first argument received by the expected invocation.
Т3	The type of the second argument received by the expected invocation.
13	The type of the third argument received by the expected invocation.
T4	The type of the fourth argument received by the expected invocation.
T5	The type of the fifth argument received by the expected invocation.
T6	The type of the sixth argument received by the expected invocation.
T7	The type of the seventh argument received by the expected invocation.
T8 T9	The type of the eighth argument received by the expected invocation.
T10	The type of the nineth argument received by the expected invocation.
T11	The type of the tenth argument received by the expected invocation.
T12	The type of the eleventh argument received by the expected invocation.
T13	The type of the twelfth argument received by the expected invocation.
	The type of the thirteenth argument received by the expected invocation.
T14	The type of the fourteenth argument received by the expected invocation

```
IRaise<(Of <(<'T>)>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <(<'T>)>)>..::..Raises(Action<(Of <<'(T>)>>), EventArgs)
```

_

□ C#

Moq

IRaise<(Of <(<'T>)>)>..::..Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>)> Method (Action<(Of <(<'T>)>)>, Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, EventArgs>)>)>)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Specifies the event that will be raised when the setup is matched.

Namespace: Moq.Language

C#

Parameters

```
eventExpression
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

The expression that represents an event attach or detach action.

func

```
Type: <u>System..:..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, <u>EventArgs</u>>)>)>
```

The function that will build the **EventArgs** to pass when raising the event.

T1	
T2	The type of the first argument received by the expected invocation.
Т3	The type of the second argument received by the expected invocation.
т3 Т4	The type of the third argument received by the expected invocation.
	The type of the fourth argument received by the expected invocation.
T5 T6	The type of the fifth argument received by the expected invocation.
T7	The type of the sixth argument received by the expected invocation.
T7 T8	The type of the seventh argument received by the expected invocation.
TO	The type of the eighth argument received by the expected invocation.
T9 T10	The type of the nineth argument received by the expected invocation.
	The type of the tenth argument received by the expected invocation.
T11 T12	The type of the eleventh argument received by the expected invocation.
T13	The type of the twelfth argument received by the expected invocation.
T14	The type of the thirteenth argument received by the expected invocation.
	The type of the fourteenth argument received by the expected invocation
T15	The type of the fifteenth argument received by the expected invocation.

```
IRaise<(Of <(<'T>)>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <(<'T>)>)>..::..Raises(Action<(Of <<'(T>)>>), EventArgs)
```

_

□ C#

Moq

IRaise<(Of <(<'T>)>)>..::..Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>)> Method (Action<(Of <(<'T>)>)>, Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, EventArgs>)>)>)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Specifies the event that will be raised when the setup is matched.

Namespace: Moq.Language

C#

Parameters

```
eventExpression
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

The expression that represents an event attach or detach action.

func

```
Type: <u>System..:..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, <u>EventArgs</u>>)>)>
```

The function that will build the **EventArgs** to pass when raising the event.

T1	
T2	The type of the first argument received by the expected invocation.
 Т3	The type of the second argument received by the expected invocation.
	The type of the third argument received by the expected invocation.
T4	The type of the fourth argument received by the expected invocation.
T5	The type of the fifth argument received by the expected invocation.
T6	The type of the sixth argument received by the expected invocation.
T7	The type of the seventh argument received by the expected invocation.
Т8	
Т9	The type of the eighth argument received by the expected invocation.
T10	The type of the nineth argument received by the expected invocation.
T11	The type of the tenth argument received by the expected invocation.
T12	The type of the eleventh argument received by the expected invocation.
T13	The type of the twelfth argument received by the expected invocation.
	The type of the thirteenth argument received by the expected invocation.
T14	The type of the fourteenth argument received by the expected invocation
T15	The type of the fifteenth argument received by the expected invocation.
T16	The type of the sixteenth argument received by the expected invocation.
	71

```
IRaise<(Of <(<'T>)>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <(<'T>)>)>..::..Raises(Action<(Of <<'(T>)>>), EventArgs)
```

□ C#

_

Moq

IRaise<(Of <(<'T>)>)>..::..Raises<(Of <(<'T1>)>)> Method (Action<(Of <(<'T>)>)>, Func)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IRaise`1.Raises``1(System.Action{`0},System.Func`2)"]

Namespace: Moq.Language

C#

Parameters

```
eventExpression
```

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

func

Type: <u>System..:..Func<(Of <(<'T, TResult>)>)></u>

T1

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

⊟ □ C#

Moq

IRaise<(Of <(<'T>)>)>..:..Raises<(Of <(<'T1, T2>)>)> Method (Action<(Of <(<'T>)>)>, Func)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IRaise`1.Raises``2(System.Action{`0},System.Func`3)"]

Namespace: Moq.Language

C#

Parameters

```
eventExpression
```

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

func

Type: <u>System.::..Func<(Of <(<'T1, T2, TResult>)>)></u>

T1 T2

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

⊟ □ C#

Moq

IRaise<(Of <(<'T>)>)>.....Raises<(Of <(<'T1, T2, T3>)>)> Method (Action<(Of <(<'T>)>)>, Func)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IRaise`1.Raises``3(System.Action{`0},System.Func`4)"]

Namespace: Moq.Language

C#

Parameters

```
event Expression\\
```

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

func

Type: <u>System..:..Func<(Of <(<'T1, T2, T3, TResult>)>)></u>

T1

T2

T3

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

□ C#

_

Moq

IRaise<(Of <(<'T>)>)>.....Raises<(Of <(<'T1, T2, T3, T4>)>)> Method (Action<(Of <(<'T>)>)>, Func)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IRaise`1.Raises``4(System.Action{`0},System.Func`5)"]

Namespace: Moq.Language

C#

Parameters

```
event Expression\\
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

func

Type: <u>System..:..Func<(Of <(<'T1, T2, T3, T4, TResult>)>)></u>

T1

T2

T3

T4

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

□ C#

_

Moq

IRaise<(Of <(<'T>)>)>..::.Raises<(Of <(<'T1, T2, T3, T4, T5>)>)> Method (Action<(Of <(<'T>)>)>, Func)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IRaise`1.Raises``5(System.Action{`0},System.Func`6)"]

Namespace: Moq.Language

C#

Parameters

```
event Expression\\
```

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

func

Type: <u>System..:..Func<(Of <(<'T1, T2, T3, T4, T5, TResult>)>)></u>

T1

T2

Т3

T4

T5

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

```
□ C#

Moq

IRaise<(Of <(<'T>)>)>..::..Raises<(Of <(<'T1, T2, T3, T4, T5, T6>)>)>

Method (Action<(Of <(<'T>)>)>, Func)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback
```

[Missing <summary> documentation for "M:Moq.Language.IRaise`1.Raises``6(System.Action{`0},System.Func`7)"]

Namespace: Moq.Language

C#

Parameters

```
event Expression\\
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

func

Type: <u>System.::..Func<(Of <(<'T1, T2, T3, T4, T5, T6, TResult>)>)></u>

T1

T2

T3

T4

T5

T6

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

```
⊟
□ C#
```

Moq

IRaise<(Of <(<'T>)>)>.....Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)> Method (Action<(Of <(<'T>)>)>, Func)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IRaise`1.Raises``7(System.Action{`0},System.Func`8)"]

Namespace: Moq.Language

C#

Parameters

```
event Expression\\
```

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

func

Type: <u>System..:..Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, TResult>)>)></u>

T1

T2

Т3

T4

T5

T6

T7

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

⊟ □ C#

Moq

IRaise<(Of <(<'T>)>)>..::..Raises<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8>)>)> Method (Action<(Of <(<'T>)>)>, Func)

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IRaise`1.Raises``8(System.Action{`0},System.Func`9)"]

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
event Expression\\
```

Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>

func

Type: <u>System..:..Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, </u>

<u>TResult>)>)></u>

Type Parameters

T1

T2

T3

T4

T5

T6

T7

T8

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

□ C#

_

Moq

IRaise<(Of <(<'T>)>)>..::..Raises Method (Action<(Of <(<'T>)>)>, array<Object>[]()[][])

IRaise<(Of <(<'T>)>)> Interface See Also Send Feedback

Specifies the custom event that will be raised when the setup is matched.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
event Expression\\
```

```
Type: <u>System..:..Action</u><(Of <(<'<u>T</u>>)>)>
```

An expression that represents an event attach or detach action.

args

Type: array<<u>System..:..Object</u>>[]()[][]

The arguments to pass to the custom delegate (non EventHandler-compatible).

IRaise<(Of <(<'T>)>)> Interface Raises Overload Moq.Language Namespace

_

□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)> Interface

Members See Also Send Feedback

Defines the Returns verb.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

public interface IReturns<TMock, TResult> : IHideObjectMembers
where TMock : class

Type Parameters

TMock

Mocked type.

TResult

Type of the return value from the expression.

IReturns<(Of <(<'TMock, TResult>)>)> Members
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
IReturns<(Of <(<'TMock, TResult>)>)> Members

IReturns<(Of <(<'TMock, TResult>)>)> Interface Methods See Also Send Feedback

Methods

	Name	Description
≡	<pre>Returns(Func<(Of <(<'TResult>)>)>)</pre>	
≡	Returns(TResult)	Specifies the value to return.
≡	<u>Returns<(Of <<'(T>)>>)(Func<(Of <(<'T, TResult>)>)>)</u>	
∉∳	Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
≡	Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
≡	Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
≡	Returns < (Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>) (Func < (Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
≡	Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>) (Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
	Returns < (Of < < '(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14,	Specifies a function that will calculate the value to

T15>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, TResult>)>>))

Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15,

- T16>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, TResult>)>>))
- Returns<(Of <<'(T1, T2>)>>)(Func<(Of <('T1, T2, TResult>)>)>)
- Returns<(Of <<'(T1, T2, T3>)>>)
 (Func<(Of <(<'T1, T2, T3, TResult>)>)>)
 Returns<(Of <<'(T1, T2, T3, T4>)>>)
- (Func<(Of <(<'T1, T2, T3, T4, TResult>)>)>)
 - <u>Returns<(Of <<'(T1, T2, T3, T4, T5>)>>)</u>
- - <u>Returns<(Of <<'(T1, T2, T3, T4, T5,</u>
- T6>)>>)(Func<(Of <(<'T1, T2, T3, T4, T5, T6, TResult>)>)>)
- Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7>)>>)(Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, TResult>)>)>)
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6,
- T7, T8>)>>)(Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, TResult>)>)>)
- Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T4, T5, T6, T7, T8, T9, TResult>)>>))

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.

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Moq.Language Namespace

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

Methods

	Name	Description
≡	<pre>Returns(Func<(Of <(<'TResult>)>)>)</pre>	
≡	Returns(TResult)	Specifies the value to return.
≡	<u>Returns<(Of <<'(T>)>>)(Func<(Of <(<'T, TResult>)>)>)</u>	
∉∳	Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
≡	Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
≡	Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
≡	Returns < (Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>) (Func < (Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
≡	Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>>) (Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult>)>>))	Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.
	Returns < (Of < < '(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14,	Specifies a function that will calculate the value to

T15>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, TResult>)>>))

Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15,

- T16>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, TResult>)>>))
- Returns<(Of <<'(T1, T2>)>>)(Func<(Of <('T1, T2, TResult>)>)>)
- Returns<(Of <<'(T1, T2, T3>)>>)
 (Func<(Of <(<'T1, T2, T3, TResult>)>)>)
 Returns<(Of <<'(T1, T2, T3, T4>)>>)
- (Func<(Of <(<'T1, T2, T3, T4, TResult>)>)>)
 - <u>Returns<(Of <<'(T1, T2, T3, T4, T5>)>>)</u>
- - <u>Returns<(Of <<'(T1, T2, T3, T4, T5,</u>
- T6>)>>)(Func<(Of <(<'T1, T2, T3, T4, T5, T6, TResult>)>)>)
- Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7>)>>)(Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, TResult>)>)>)
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6,
- T7, T8>)>>)(Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, TResult>)>)>)
- Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T4, T5, T6, T7, T8, T9, TResult>)>>))

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.

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Moq.Language Namespace

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

Overload List

•

Name

- Returns(Func<(Of <(<'TResult>)>)>)
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, TResult>)>>))
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult>)>>))
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, TResult>)>>))
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, TResult>)>>))
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>>) (Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, TResult>)>>))

Description

- 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 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

- T7, T8, T9, T10, T11, T12, T13, T14, T15>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, TResult>)>>))
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15,
- T16>)>>)(Func<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, TResult>)>>))
- Returns<(Of <<'(T>)>>)(Func<(Of <(<'T, TResult>)>)>)
- Returns<(Of <<'(T1, T2>)>>)(Func<(Of <('T1, T2, TResult>)>)>)
- Returns<(Of <<'(T1, T2, T3>)>>)
 (Func<(Of <(<'T1, T2, T3, TResult>)>)>)
 - <u>Returns<(Of <<'(T1, T2, T3, T4>)>>)</u>
- (Func≤(Of ≤(≤'T1, T2, T3, T4, TResult≥)≥)
 - <u>Returns<(Of <<'(T1, T2, T3, T4, T5>)>>)</u>
- (Func<(Of <(<'T1, T2, T3, T4, T5, TResult>)>)>)
 - Returns<(Of <<'(T1, T2, T3, T4, T5,
- T6>)>>)(Func<(Of <(<'T1, T2, T3, T4, T5, T6, TResult>)>)>)
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6,
- T7>)>>)(Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, TResult>)>)>)
 - Returns<(Of <<'(T1, T2, T3, T4, T5, T6,
- T7, T8>)>>)(Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, TResult>)>)>)
- Returns(TResult)

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

```
□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>..::.Returns Method (Func)

IReturns<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback
```

[Missing <summary> documentation for "M:Moq.Language.IReturns`2.Returns(System.Func`1)"]

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

valueFunction

Type: <u>System..:..Func<(Of <(<'TResult>)>)></u>

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

-

□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>.::..Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9>)>)> Method (Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, TResult>)>)>)

<u>IReturns<(Of <(<'TMock, TResult>)>)> Interface Example See Also 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

valueFunction

Type: <u>System..:..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, <u>TResult</u>>)>)>

The function that will calculate the return value.

Type Parameters

TT-1	
T1	The type of the first argument of the invoked method.
T2	The type of the second argument of the invoked method.
Т3	The type of the second argument of the invoked method.
T4	The type of the third argument of the invoked method.
	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
T6	
T7	The type of the sixth argument of the invoked method.
Т8	The type of the seventh argument of the invoked method.
	The type of the eighth argument of the invoked method.
T9	The type of the nineth argument of the invoked method.

Examples

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:

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

_

□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>..:..Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>)>)> Method (Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult>)>)>)

<u>IReturns<(Of <(<'TMock, TResult>)>)> Interface Example See Also 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

valueFunction

Type: <u>System..:..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, <u>TResult</u>>)>)>

The function that will calculate the return value.

Type Parameters

The type of the first argument of the invoked method.
The type of the second argument of the invoked method.
The type of the third argument of the invoked method.
The type of the fourth argument of the invoked method.
The type of the fifth argument of the invoked method.
The type of the sixth argument of the invoked method.
The type of the seventh argument of the invoked method.
The type of the eighth argument of the invoked method.
The type of the nineth argument of the invoked method.
The type of the tenth argument of the invoked method.

Examples

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:

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

-

□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>.::..Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>)>)> Method (Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, TResult>)>)>)

<u>IReturns<(Of <(<'TMock, TResult>)>)> Interface Example See Also 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

Syntax

C#

Parameters

valueFunction

```
Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, <u>TResult</u>>)>)>
```

The function that will calculate the return value.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
Т3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	
Т6	The type of the fifth argument of the invoked method.
T7	The type of the sixth argument of the invoked method.
T8	The type of the seventh argument of the invoked method.
	The type of the eighth argument of the invoked method.
Т9	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	
	The type of the eleventh argument of the invoked method.

Examples

C#

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:

Copy mock.Setup(x => x.Execute(It.IsAny<int>(), It.IsAny<int>()))

.Returns((string arg1, string arg2, string arg3, string arg4, st

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

-

□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>..::.Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>)>)> Method (Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, TResult>)>)>)

<u>IReturns<(Of <(<'TMock, TResult>)>)> Interface Example See Also 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

Syntax

C#

Parameters

valueFunction

Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, <u>TResult</u>>)>)>

The function that will calculate the return value.

Type Parameters

T1	
TTO.	The type of the first argument of the invoked method.
T2	The type of the second argument of the invoked method.
T3	The type of the third argument of the invoked method.
T4	
T5	The type of the fourth argument of the invoked method.
Т6	The type of the fifth argument of the invoked method.
	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	The type of the nineth argument of the invoked method.
T10	
T11	The type of the tenth argument of the invoked method.
T12	The type of the eleventh argument of the invoked method.
112	The type of the twelfth argument of the invoked method.

Examples

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:

Copy C# mock.Setup(x => x.Execute(It.IsAny<int>(), It.IsAny<int>())) .Returns((string arg1, string arg2, string arg3, string arg4, st

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

_

□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>.::..Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>)>)> Method (Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, TResult>)>)>)

<u>IReturns<(Of <(<'TMock, TResult>)>)> Interface Example See Also 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

Syntax

C#

Parameters

valueFunction

Type: <u>System..:..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, <u>TResult</u>>)>)>

The function that will calculate the return value.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
T3	The type of the second argument of the invoked method.
	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
Т6	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	The type of the eleventh argument of the invoked method.
T12	The type of the twelfth argument of the invoked method.
T13	The type of the thirteenth argument of the invoked method.

Examples

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:

Copy C# mock.Setup(x => x.Execute(It.IsAny<int>(), It.IsAny<int>())) .Returns((string arg1, string arg2, string arg3, string arg4, st

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

-

□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>..::..Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>)>)> Method (Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult>)>)>)

IReturns<(Of <(<'TMock, TResult>)>)> Interface Example See Also Send Feedback

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

Namespace: Moq.Language

Syntax

C#

Parameters

valueFunction

Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, <u>TResult</u>>)>)>

The function that will calculate the return value.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
T)	The type of the second argument of the invoked method.
T3	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
T6	The type of the sixth argument of the invoked method.
T7	
T8	The type of the seventh argument of the invoked method.
Т9	The type of the eighth argument of the invoked method.
	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	The type of the eleventh argument of the invoked method.
T12	
T13	The type of the twelfth argument of the invoked method.
T14	The type of the thirteenth argument of the invoked method.
114	The type of the fourteenth argument of the invoked method

Examples

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

```
mock.Setup(x => x.Execute(
                      It.IsAny<int>(),
                     It.IsAny<int>()))
    .Returns((string arg1, string arg2, string arg3, string arg4, st
```

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

_

□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>.::..Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>)> Method (Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, TResult>)>)>)

<u>IReturns<(Of <(<'TMock, TResult>)>)> Interface Example See Also 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

Syntax

C#

Parameters

valueFunction

Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, <u>TResult</u>>)>)>

The function that will calculate the return value.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
T3	The type of the second argument of the invoked method.
	The type of the third argument of the invoked method.
T4	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
Т6	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	The type of the nineth argument of the invoked method.
T10	The type of the tenth argument of the invoked method.
T11	The type of the eleventh argument of the invoked method.
T12	
T13	The type of the twelfth argument of the invoked method.
T14	The type of the thirteenth argument of the invoked method.
T15	The type of the fourteenth argument of the invoked method.
	The type of the fifteenth argument of the invoked method.

Examples

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

```
mock.Setup(x => x.Execute(
                     It.IsAny<int>(),
                     It.IsAny<int>()))
    .Returns((string arg1, string arg2, string arg3, string arg4, st
```

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

-

□ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>.::..Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)>)> Method (Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, TResult>)>)>)

<u>IReturns<(Of <(<'TMock, TResult>)>)> Interface Example See Also 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

Syntax

C#

Parameters

valueFunction

Type: <u>System.::..Func</u><(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, <u>TResult</u>>)>)>

The function that will calculate the return value.

Type Parameters

T1	
T2	The type of the first argument of the invoked method.
	The type of the second argument of the invoked method.
T3	The type of the third argument of the invoked method.
T4 	The type of the fourth argument of the invoked method.
T5	The type of the fifth argument of the invoked method.
Т6	The type of the sixth argument of the invoked method.
T7	The type of the seventh argument of the invoked method.
Т8	The type of the eighth argument of the invoked method.
Т9	The type of the nineth argument of the invoked method.
T10	
T11	The type of the tenth argument of the invoked method.
T12	The type of the eleventh argument of the invoked method.
T13	The type of the twelfth argument of the invoked method.
T14	The type of the thirteenth argument of the invoked method.
T15	The type of the fourteenth argument of the invoked method.
T16	The type of the fifteenth argument of the invoked method.
110	The type of the sixteenth argument of the invoked method.

Examples

C#

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:

Copy

.Returns((string arg1, string arg2, string arg3, string arg4, st

It.IsAny<int>(),
It.IsAny<int>(),
It.IsAny<int>(),
It.IsAny<int>())

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

⊟ □ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>..:..Returns<(Of <(<'T>)>)> Method (Func)

IReturns<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IReturns`2.Returns`1(System.Func`2)"]

Namespace: Moq.Language

Syntax

C#

Parameters

valueFunction

Type: <u>System..::..Func<(Of <(<'T, TResult>)>)></u>

Type Parameters

T

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

```
□ C#
```

_

Moq

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

IReturns<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IReturns`2.Returns`2(System.Func`3)"]

Namespace: Moq.Language

C#

Parameters

valueFunction

Type: <u>System..:..Func</u><(<u>Of</u> <(<u><'T1, T2, TResult</u>>)>)>

T1 T2

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

```
⊟
□ C#
```

Moq

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

IReturns<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IReturns`2.Returns`3(System.Func`4)"]

Namespace: Moq.Language

C#

Parameters

valueFunction

Type: <u>System..:..Func<(Of <(<'T1, T2, T3, TResult>)>)></u>

T1

T2

T3

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

```
□ C#

Moq

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

IReturns<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback
```

[Missing <summary> documentation for "M:Moq.Language.IReturns`2.Returns`4(System.Func`5)"]

Namespace: Moq.Language

C#

Parameters

valueFunction

Type: <u>System..:..Func<(Of <(<'T1, T2, T3, T4, TResult>)>)></u>

T1

T2

T3

T4

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

⊟ □ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>..:..Returns<(Of <(<'T1, T2, T3, T4, T5>)>)> Method (Func)

IReturns<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IReturns`2.Returns`5(System.Func`6)"]

Namespace: Moq.Language

C#

Parameters

valueFunction

Type: <u>System..:..Func</u><(<u>Of</u> <(<u><</u>'<u>T1, T2, T3, T4, T5, TResult</u>>)>)>

T1

T2

Т3

T4

T5

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

⊟ □ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>.::..Returns<(Of <(<'T1, T2, T3, T4, T5, T6>)>)> Method (Func)

IReturns<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IReturns`2.Returns``6(System.Func`7)"]

Namespace: Moq.Language

C#

Parameters

valueFunction

Type: <u>System.::..Func<(Of <(<'T1, T2, T3, T4, T5, T6, TResult>)>)></u>

T1

T2

T3

T4

T5

T6

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

⊟ □ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>..:..Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7>)>)> Method (Func)

IReturns<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IReturns`2.Returns`7(System.Func`8)"]

Namespace: Moq.Language

C#

Parameters

valueFunction

Type: <u>System..:..Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, TResult>)>)></u>

T1

T2

T3

T4

T5

T6

T7

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

⊟ □ C#

Moq

IReturns<(Of <(<'TMock, TResult>)>)>..:..Returns<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8>)>)> Method (Func)

IReturns<(Of <(<'TMock, TResult>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Language.IReturns`2.Returns`8(System.Func`9)"]

Namespace: Moq.Language

C#

Parameters

```
valueFunction
```

```
Type: <u>System..:..Func<(Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, TResult>)>)></u>
```

T1

T2

T3

T4

T5

T6

T7

T8

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

-

□ C#

Moq

 $IReturns < (Of < (< 'TMock, TResult>)>) >Returns Method (TResult) \\ \underline{IReturns} < (Of < (< 'TMock, TResult>)>) > \underline{Interface} \ \underline{Example} \ \underline{See \ Also} \ \underline{Send}$

Feedback

Specifies the value to return.

Namespace: Moq.Language

\blacksquare Syntax

C#

Parameters

value

Type: TResult

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

Examples

Return a true value from the method call:

C#

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

IReturns<(Of <(<'TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

-

□ C#

Moq

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Interface
Members See Also Send Feedback

Defines the Returns verb for property get setups.

Namespace: Moq.Language

C#

public interface IReturnsGetter<TMock, TProperty> : IHideObjectMemb{ where TMock : class}

TMock
Mocked type.
TProperty
Type of the property.

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Members
Moq.Language Namespace

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Members

IReturnsGetter<(Of <(<'TMock, TProperty>)>)>
Interface Methods See Also Send Feedback
```

Methods

Name Description

- Returns(Func<(Of <(<'TResult>)>)>)
- Returns(TProperty)
 Specifies the value to return.

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Interface
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Methods

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Interface See Also Send Feedback

Methods

Name Description

- Returns(Func<(Of <(<'TResult>)>)>)
- Returns(TProperty)
 Specifies the value to return.

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Interface
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
IReturnsGetter<(Of <(<'TMock, TProperty>)>)>..:..Returns Method

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Interface See Also Send
Feedback

Overload List

Name

Description

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

Specifies the value to return.

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Interface IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Members Moq.Language Namespace

-

□ C#

Moq

IReturnsGetter<(Of <(<'TMock, TProperty>)>)>..:..Returns Method (Func)

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Interface See Also Send

Feedback

[Missing <summary> documentation for "M:Moq.Language.IReturnsGetter`2.Returns(System.Func`1)"]

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

valueFunction

Type: <u>System..:..Func<(Of <(<'TResult>)>)></u>

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Interface Returns Overload Moq.Language Namespace

-

□ C#

Moq

IReturnsGetter<(Of <(<'TMock, TProperty>)>)>.....Returns Method (TProperty)

IReturnsGetter<(Of <(<'TMock, TProperty>)>)>
Interface Example See Also Send Feedback

Specifies the value to return.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare 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#

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

IReturnsGetter<(Of <(<'TMock, TProperty>)>)> Interface Returns Overload Moq.Language Namespace

-

□ C#

Moq

ISetupConditionResult < (Of < (< 'T>)>) > Interface

Members See Also Send Feedback

Implements the fluent API.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public interface ISetupConditionResult<T>
where T : class

Type Parameters

T

ISetupConditionResult<(Of <(<'T>)>)> Members
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ISetupConditionResult<(Of <(<'T>)>)> Members

ISetupConditionResult<(Of <(<'T>)>)> Interface Methods See Also Send Feedback

The $\underline{ISetupConditionResult} \leq (\underline{Of} \leq (\leq '\underline{T} \geq) \geq) \geq$ type exposes the following members.

\blacksquare Methods

	Name	Description
=	Setup(Expression<(Of <<'(Action<(Of <<'(T>)>>)>))	The expectation will be considered only in the former condition.
≡	<u>Setup<(Of <<'(TResult>)>>)</u> (<u>Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>)</u>)	
=	<pre>SetupGet<(Of <<'(TProperty>)>>)</pre>	
=	$\underline{SetupSet(Action < (Of < < '(T >) > >))}$	Setups the set.
≅	<u>SetupSet<(Of <<'(TProperty>)>>)</u> (<u>Action<(Of <<'(T>)>>)</u>)	Setups the set.

<u>ISetupConditionResult<(Of <(<'T>)>)> Interface</u> <u>Moq.Language Namespace</u>

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ISetupConditionResult<(Of <(<'T>)>)> Methods

ISetupConditionResult<(Of <(<'T>)>)> Interface See Also Send Feedback

The $\underline{\mathsf{ISetupConditionResult}} \leq (\underline{\mathsf{Of}} \leq (\leq '\underline{\mathsf{T}} \geq) \geq) \geq \mathsf{type}$ exposes the following members.

\blacksquare Methods

	Name	Description
=	Setup(Expression<(Of <<'(Action<(Of <<'(T>)>>)>))	The expectation will be considered only in the former condition.
≡	<u>Setup<(Of <<'(TResult>)>>)</u> (<u>Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)>>)</u>)	
=	<pre>SetupGet<(Of <<'(TProperty>)>>)</pre>	
=	$\underline{SetupSet(Action < (Of < < '(T >) > >))}$	Setups the set.
≅	<u>SetupSet<(Of <<'(TProperty>)>>)</u> (<u>Action<(Of <<'(T>)>>)</u>)	Setups the set.

<u>ISetupConditionResult<(Of <(<'T>)>)> Interface</u> <u>Moq.Language Namespace</u>

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ISetupConditionResult<(Of <(<'T>)>)>..::..Setup Method

ISetupConditionResult<(Of <(<'T>)>)> Interface See Also Send Feedback

Overload List

Name

- $\frac{\text{Setup}(\text{Expression} < (\text{Of} <<' (\text{Action} < (\text{Of} <<' (\text{Of} <<' (\text{Action} < (\text{Of} <<' (\text{Action} < (\text{Of} <<' () <<' (\text{Of} <<' (\text{Of} <<' (\text{Of} <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' () <<' ()$
 - Setup<(Of <<'(TResult>)>>)
- (Expression<(Of <<'(Func<(Of <(<'T, TResult>)>)>>)))

Description

The expectation will be considered only in the former condition.

ISetupConditionResult<(Of <(<'T>)>)> Interface
ISetupConditionResult<(Of <(<'T>)>)> Members
Moq.Language Namespace

-

□ C#

Moq

ISetupConditionResult < (Of < (<'T>)>)>.....Setup Method (Expression < (Of < (<'Action < (Of < (<'T>)>)>>)>))

ISetupConditionResult<(Of <(<'T>)>)> Interface See Also Send Feedback

The expectation will be considered only in the former condition.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression
```

Return Value

ISetupConditionResult<(Of <(<'T>)>)> Interface Setup Overload Moq.Language Namespace

□ C#

_

Moq

ISetupConditionResult<(Of <(<'T>)>)>..:..Setup<(Of <(<'TResult>)>)> Method (Expression<(Of <(<'Func>)>)>)

ISetupConditionResult<(Of <(<'T>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for

 $"M: Moq. Language. I Setup Condition Result`1. Setup``1 (System. Linq. Expressions. Expression \{System. Fig. 1) and the properties of th$

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression \\
```

Type Parameters

TResult

ISetupConditionResult<(Of <(<'T>)>)> Interface Setup Overload Moq.Language Namespace

⊟ □ C#

Moq

ISetupConditionResult < (Of < (< 'T>)>)> SetupGet < (Of < (< 'TProperty>)>)> Method

ISetupConditionResult<(Of <(<'T>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for

 $"M: Moq. Language. I Setup Condition Result `1. Setup Get ``1 (System. Linq. Expressions. Expression \{System. Linq. Expression \} \} = \{System. Linq. Expression \{System. Linq. Expression \} \} = \{System. Linq. Expression \} = \{Syst$

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
expression\\
```

Type Parameters

TProperty

ISetupConditionResult<(Of <(<'T>)>)> Interface
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ISetupConditionResult<(Of <(<'T>)>)>..::..SetupSet Method

ISetupConditionResult<(Of <(<'T>)>)> Interface See Also Send Feedback

Overload List

	Name	Description
≡	$\underline{SetupSet(Action < (Of <<'(T>)>>))}$	Setups the set.
≡	<u>SetupSet<(Of <<'(TProperty>)>>)(Action<(Of <<'(T>)>>))</u>	Setups the set.

ISetupConditionResult<(Of <(<'T>)>)> Interface
ISetupConditionResult<(Of <(<'T>)>)> Members
Moq.Language Namespace

-

□ C#

Moq

ISetupConditionResult<(Of <(<'T>)>)>..:..SetupSet Method (Action<(Of <(<'T>)>)>)

 $\underline{ISetupConditionResult} \leq \underline{(Of \leq (\leq 'T \geq) \geq)} \geq \underline{Interface \ See \ Also \ Send \ Feedback}$ Setups the set.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

Parameters

```
\label{eq:continuity} \begin{array}{c} \text{setterExpression} \\ \text{Type: } \underline{\text{System..:..}Action} < (\text{Of } < (<'\underline{\text{T}}>)>)> \\ \text{The setter expression.} \end{array}
```

Return Value

<u>ISetupConditionResult<(Of <(<'T>)>)> Interface</u> <u>SetupSet Overload</u> <u>Moq.Language Namespace</u>

-

□ C#

Moq

 $ISetupConditionResult<(Of<(<'T>)>)>.....SetupSet<(Of<(<'TProperty>)>)> \\ Method~(Action<(Of<(<'T>)>)>)$

ISetupConditionResult<(Of <(<'T>)>)> Interface See Also Send Feedback

Setups the set.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
setter Expression\\
```

Type: <u>System.....Action</u><(Of <(<'<u>T</u>>)>)>

The setter expression.

Type Parameters

TProperty

The type of the property.

Return Value

<u>ISetupConditionResult<(Of <(<'T>)>)> Interface</u> <u>SetupSet Overload</u> <u>Moq.Language Namespace</u>

-

□ C#

Moq

ISetupSequentialResult<(Of <(<'TResult>)>)> Interface

Members See Also Send Feedback

Language for ReturnSequence

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public interface ISetupSequentialResult<TResult>

Type Parameters

TResult

ISetupSequentialResult<(Of <(<'TResult>)>)> Members
Moq.Language Namespace

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ISetupSequentialResult<(Of <(<'TResult>)>)> Members

ISetupSequentialResult<(Of <(<'TResult>)>)>
Interface Methods See Also Send Feedback
```

The $\underline{ISetupSequentialResult} \le (\underline{Of} \le (\le \underline{TResult} \ge) \ge) \ge$ type exposes the following members.

\blacksquare Methods

	Name Description	
≡	<u>Returns</u>	Returns value
≡	Throws(Exception)	Throws an exception
≡	Throws<(Of <<'(TException>)>	≥)()()() Throws an exception

ISetupSequentialResult<(Of <(<'TResult>)>)> Interface
Moq.Language Namespace

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ISetupSequentialResult<(Of <(<'TResult>)>)> Methods

ISetupSequentialResult<(Of <(<'TResult>)>)> Interface See Also Send Feedback
```

The $\underline{ISetupSequentialResult} \le (\underline{Of} \le (\le \underline{TResult} \ge) \ge) \ge type$ exposes the following members.

\blacksquare Methods

	Name Description	
≡	<u>Returns</u>	Returns value
≡	Throws(Exception)	Throws an exception
≡	Throws<(Of <<'(TException>)>	≥)()()() Throws an exception

ISetupSequentialResult<(Of <(<'TResult>)>)> Interface
Moq.Language Namespace

-

□ C#

Moq

ISetupSequentialResult<(Of <(<'TResult>)>)>..::..Returns Method

ISetupSequentialResult<(Of <(<'TResult>)>)> Interface See Also Send
Feedback

Returns value

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

```
ISetupSequentialResultTResult value
)
```

Parameters

value

Type: TResult

ISetupSequentialResult<(Of <(<'TResult>)>)> Interface
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
ISetupSequentialResult<(Of <(<'TResult>)>)>..:..Throws Method

ISetupSequentialResult<(Of <(<'TResult>)>)> Interface See Also Send
Feedback

Overload List

Name Description

- <u>Throws<(Of <<'(TException>)>>)()()()</u> Throws an exception
- Throws(Exception)
 Throws an exception

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

```
⊟
□ C#
```

Moq

ISetupSequentialResult<(Of <(<'TResult>)>)>..:..Throws<(Of <(<'TException>)>)> Method

<u>ISetupSequentialResult<(Of <(<'TResult>)>)> Interface See Also Send Feedback</u>

Throws an exception

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

```
void Throws<TException>()
where TException : new(), Exception
```

Type Parameters

TException

<u>ISetupSequentialResult<(Of <(<'TResult>)>)> Interface</u>
<u>Throws Overload</u>
<u>Moq.Language Namespace</u>

-

□ C#

Moq

ISetupSequentialResult<(Of <(<'TResult>)>)>..:..Throws Method (Exception)

<u>ISetupSequentialResult<(Of <(<'TResult>)>)> Interface See Also Send</u> <u>Feedback</u>

Throws an exception

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

Parameters

exception

Type: <u>System..:..Exception</u>

<u>ISetupSequentialResult<(Of <(<'TResult>)>)> Interface</u>
<u>Throws Overload</u>
<u>Moq.Language Namespace</u>

_

□ C#

Moq

IThrows Interface

Members See Also Send Feedback

Defines the Throws verb.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public interface IThrows : IHideObjectMembers

IThrows Members
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
IThrows Members

IThrows Interface Methods See Also Send Feedback

Methods

	Name	Description
=	Throws(Exception)	Specifies the exception to throw when the method is invoked.
≓	Throws<(Of <<'(TException>)>>)()(Specifies the type of exception to throw ()()() when the method is invoked.

IThrows Interface Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
IThrows Methods

IThrows Interface See Also Send Feedback

Methods

	Name	Description
=	Throws(Exception)	Specifies the exception to throw when the method is invoked.
≓	Throws<(Of <<'(TException>)>>)()(Specifies the type of exception to throw ()()() when the method is invoked.

IThrows Interface Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
IThrows..:..Throws Method

IThrows Interface See Also Send Feedback

Overload List

	Name	Description
	Throws<(Of <<'(TException>)>>)()()()()	Specifies the type of exception to throw when the method is invoked.
≟∳	Throws(Exception)	Specifies the exception to throw when the method is invoked.

IThrows Interface
IThrows Members
Moq.Language Namespace

-

□ 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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

IThrowsResult Throws<TException>()
where TException : new(), Exception

Type Parameters

TException

Type of exception to instantiate and throw when the setup is matched.

Examples

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

C# Copy

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

IThrows Interface
Throws Overload
Moq.Language Namespace

_

□ C#

Moq

IThrows..:..Throws Method (Exception)

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: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

Parameters

exception

Type: <u>System..:..Exception</u> Exception instance to throw.

Examples

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

C# Copy

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

IThrows Interface
Throws Overload
Moq.Language Namespace

-

□ C#

Moq

IVerifies Interface

Members See Also Send Feedback

Defines the Verifiable verb.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public interface IVerifies : IHideObjectMembers

IVerifies Members
Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
IVerifies Members

IVerifies Interface Methods See Also Send Feedback

Methods

Name	Description
Verifiable()()()()	Marks the expectation as verifiable, meaning that a call to Verify()()()() will check if this particular expectation was met.
Verifiable(String)	Marks the expectation as verifiable, meaning that a call to Verify()()()() will check if this particular expectation was met, and specifies a message for failures.

IVerifies Interface Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
IVerifies Methods

IVerifies Interface See Also Send Feedback

Methods

Name	Description
Verifiable()()()()	Marks the expectation as verifiable, meaning that a call to Verify()()()() will check if this particular expectation was met.
Verifiable(String)	Marks the expectation as verifiable, meaning that a call to Verify()()()() will check if this particular expectation was met, and specifies a message for failures.

IVerifies Interface Moq.Language Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
IVerifies..:..Verifiable Method

IVerifies Interface See Also Send Feedback

Overload List

Name	Description
Verifiable()()()()	Marks the expectation as verifiable, meaning that a call to Verify()()()() will check if this particular expectation was met.
Verifiable(String)	Marks the expectation as verifiable, meaning that a call to Verify()()()() will check if this particular expectation was met, and specifies a message for failures.

IVerifies Interface
IVerifies Members
Moq.Language Namespace

-

□ C#

Moq

IVerifies..:..Verifiable Method

IVerifies Interface Example See Also Send Feedback

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: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

void Verifiable()

Examples

The following example marks the expectation as verifiable:

C#

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

IVerifies Interface
Verifiable Overload
Moq.Language Namespace

_

□ C#

Moq

IVerifies..:..Verifiable Method (String)

IVerifies Interface Example See Also Send Feedback

Marks the expectation as verifiable, meaning that a call to <u>Verify()()()()</u> will check if this particular expectation was met, and specifies a message for failures.

Namespace: Moq.Language

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

```
C#
```

Parameters

failMessage

Type: <u>System..:..String</u>

Examples

The following example marks the expectation as verifiable:

C#

```
mock.Expect(x => x.Execute("ping"))
   .Returns(true)
   .Verifiable("Ping should be executed always!");
```

IVerifies Interface
Verifiable Overload
Moq.Language Namespace

☐ C#
Moq
Moq.Protected Namespace

Send Feedback

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
ぺ ≴ <u>ItExpr</u>	Allows the specification of a matching condition for an argument in a protected member setup, rather than a specific argument value. "ItExpr" refers to the argument being matched.
* ProtectedExtension	Enables the Protected() method on Mock<(Of <(<'T>)>)>, allowing setups 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

Allows setups to be specified for protected IProtectedMock<(Of members by using their name as a string, rather (<'TMock>)>)> than strong-typing them which is not possible due to their visibility.

-

□ C#

Moq

IProtectedMock<(Of <(<'TMock>)>)> Interface

Members See Also Send Feedback

Allows setups to be specified 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

public interface IProtectedMock<TMock> : IHideObjectMembers

where TMock : class

Type Parameters

TMock

IProtectedMock<(Of <(<'TMock>)>)> Members
Moq.Protected Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
IProtectedMock<(Of <(<'TMock>)>)> Members

IProtectedMock<(Of <(<'TMock>)>)> Interface Methods See Also Send Feedback

The $\underline{IProtectedMock} \le (\underline{Of} \le (\underline{\le'TMock} \ge) \ge) \ge$ type exposes the following members.

\blacksquare Methods

	Name	Description
=	Setup(String, array <object>[]()[][])</object>	Specifies a setup for a void method invocation with the given voidMethodName, optionally specifying arguments for the method call.
≘©	Setup<(Of <<'(TResult>)>>)(String, array <object>[]()[][])</object>	Specifies a setup for an invocation on a property or a non void method with the given methodOrPropertyName, optionally specifying arguments for the method call.
=	<pre>SetupGet<(Of <<'(TProperty>)>>)</pre>	Specifies a setup for an invocation on a property getter with the given propertyName.
=	<pre>SetupSet<(Of <<'(TProperty>)>>)</pre>	Specifies a setup for an invocation on a property setter with the given propertyName.
≡©	Verify(String, Times, array <object>[]()[][])</object>	Specifies a verify for a void method with the given methodName, optionally specifying arguments for the method call. Use in conjuntion with the default Loose.
≟		Specifies a verify for an invocation on a property or a non void method with the given methodName, optionally specifying arguments for the method call.
≡©	<u>VerifyGet<(Of</u> <<'(TProperty>)>>)	Specifies a verify for an invocation on a property getter with the given propertyName. The invocation was not call the times specified by times.
≅	<u>VerifySet<(Of</u> <<'(TProperty>)>>)	Specifies a setup for an invocation on a property setter with the given propertyName.

IProtectedMock<(Of <(<'TMock>)>)> Interface
Moq.Protected Namespace

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
IProtectedMock<(Of <(<'TMock>)>)> Methods

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback
```

The $\underline{IProtectedMock} \le (\underline{Of} \le (\underline{\le'TMock} \ge) \ge) \ge$ type exposes the following members.

\blacksquare Methods

	Name	Description
=	Setup(String, array <object>[]()[][])</object>	Specifies a setup for a void method invocation with the given voidMethodName, optionally specifying arguments for the method call.
≘©	Setup<(Of <<'(TResult>)>>)(String, array <object>[]()[][])</object>	Specifies a setup for an invocation on a property or a non void method with the given methodOrPropertyName, optionally specifying arguments for the method call.
=	<pre>SetupGet<(Of <<'(TProperty>)>>)</pre>	Specifies a setup for an invocation on a property getter with the given propertyName.
=	<pre>SetupSet<(Of <<'(TProperty>)>>)</pre>	Specifies a setup for an invocation on a property setter with the given propertyName.
≡©	Verify(String, Times, array <object>[]()[][])</object>	Specifies a verify for a void method with the given methodName, optionally specifying arguments for the method call. Use in conjuntion with the default Loose.
≟		Specifies a verify for an invocation on a property or a non void method with the given methodName, optionally specifying arguments for the method call.
≡©	<u>VerifyGet<(Of</u> <<'(TProperty>)>>)	Specifies a verify for an invocation on a property getter with the given propertyName. The invocation was not call the times specified by times.
≅	<u>VerifySet<(Of</u> <<'(TProperty>)>>)	Specifies a setup for an invocation on a property setter with the given propertyName.

IProtectedMock<(Of <(<'TMock>)>)> Interface
Moq.Protected Namespace

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
IProtectedMock<(Of <(<'TMock>)>)>..::..Setup Method

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback
```

Overload List

Name	Description
<u>Setup<(OI</u> <<' <u>(TResult>)>>)(String,</u> array <object>[10[1])</object>	Specifies a setup for an invocation on a property or a non void method with the given methodOrPropertyName, optionally specifying arguments for the method call.
Setup(String, array <object>[]()[][])</object>	Specifies a setup for a void method invocation with the given voidMethodName, optionally specifying arguments for the method call.

IProtectedMock<(Of <(<'TMock>)>)> Interface
IProtectedMock<(Of <(<'TMock>)>)> Members
Moq.Protected Namespace

□ C#

-

Moq

IProtectedMock<(Of <(<'TMock>)>)>..::..Setup<(Of <(<'TResult>)>)> Method (String, array<Object>[]()[][])

<u>IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback</u>

Specifies a setup for an invocation 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
method Or Property Name \\
```

Type: System.::.String

The name of the method or property to be invoked.

args

Type: array<<u>System..:..Object</u>>[]()[][]

The optional arguments for the invocation. If argument matchers are used, remember to use ItExpr rather than It.

Type Parameters

TResult

The return type of the method or property.

IProtectedMock<(Of <(<'TMock>)>)> Interface
Setup Overload
Moq.Protected Namespace

-

□ C#

Moq

IProtectedMock<(Of <(<'TMock>)>)>..:..Setup Method (String, array<Object>[]()[][])

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback

Specifies a setup for a void method invocation with the given voidMethodName, optionally specifying arguments for the method call.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

voidMethodName

Type: System.::.String

The name of the void method to be invoked.

args

Type: array<<u>System..:..Object</u>>[]()[][]

The optional arguments for the invocation. If argument matchers are used, remember to use ItExpr rather than It.

IProtectedMock<(Of <(<'TMock>)>)> Interface
Setup Overload
Moq.Protected Namespace

-

□ C#

Moq

IProtectedMock < (Of < (<'TMock>)>)>SetupGet < (Of < (<'TProperty>)>)> Method

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback

Specifies a setup for an invocation on a property getter with the given propertyName.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

propertyName
Type: <u>System..:..String</u>

The name of the property.

Type Parameters

TProperty

The type of the property.

IProtectedMock<(Of <(<'TMock>)>)> Interface
Moq.Protected Namespace

-

□ C#

Moq

IProtectedMock<(Of <(<'TMock>)>)>..::..SetupSet<(Of <(<'TProperty>)>)> Method

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback

Specifies a setup for an invocation on a property setter with the given propertyName.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
property Name \\
```

Type: <u>System..:..String</u>
The name of the property.

value

Type: <u>System..:..Object</u>

The property value. If argument matchers are used, remember to use <u>ItExpr</u> rather than <u>It</u>.

Type Parameters

TProperty

The type of the property.

IProtectedMock<(Of <(<'TMock>)>)> Interface
Moq.Protected Namespace

```
☐ C#
☐ Include Protected Members
☐ Include Inherited Members

Moq
IProtectedMock<(Of <(<'TMock>)>)>..::..Verify Method

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback
```

Overload List

	Name	Description
	<u>Verify<(Of</u>	Specifies a verify for an invocation on a
=0	<<' <u>(TResult>)>>)(String</u>	g property or a non void method with the given
	<u>Times, array<object>[](</object></u>) methodName, optionally specifying
		arguments for the method call.
		Specifies a verify for a void method with the
= 🖎	Verify(String, Times,	given methodName, optionally specifying
=	<u>array<object>[]()[][]</object></u>)	arguments for the method call. Use in
		conjuntion with the default Loose .

IProtectedMock<(Of <(<'TMock>)>)> Interface
IProtectedMock<(Of <(<'TMock>)>)> Members
Moq.Protected Namespace

```
□ C#
```

_

Moq

IProtectedMock<(Of <(<'TMock>)>)>.....Verify<(Of <(<'TResult>)>)> Method (String, Times, array<Object>[]()[][])

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback

Specifies a verify for an invocation on a property or a non void method with the given methodName, optionally specifying arguments for the method call.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
methodName
```

Type: System..:.String

The name of the method or property to be invoked.

times

Type: Moq..:..Times

The number of times a method is allowed to be called.

args

Type: array<<u>System..:..Object</u>>[]()[][]

The optional arguments for the invocation. If argument matchers are used, remember to use ItExpr rather than It.

Type Parameters

TResult

The type of return value from the expression.

Exceptions

Exception

Condition

Moq..:..MockException The invocation was not call the times specified by times.

<u>IProtectedMock<(Of <(<'TMock>)>)> Interface</u> <u>Verify Overload</u> <u>Moq.Protected Namespace</u>

-

□ C#

Moq

IProtectedMock<(Of <(<'TMock>)>)>.....Verify Method (String, Times, array<Object>[]()[][])

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback

Specifies a verify for a void method with the given methodName, optionally specifying arguments for the method call. Use in conjuntion with the default Loose.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

The optional arguments for the invocation. If argument matchers are used,

remember to use ItExpr rather than It.

Exceptions

Exception

Condition

Moq..:..MockException The invocation was not call the times specified by times.

<u>IProtectedMock<(Of <(<'TMock>)>)> Interface</u> <u>Verify Overload</u> <u>Moq.Protected Namespace</u>

□ C#

_

Moq

IProtectedMock<(Of <(<'TMock>)>)>..:..VerifyGet<(Of <(<'TProperty>)>)> Method

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback

Specifies a verify for an invocation on a property getter with the given propertyName. The invocation was not call the times specified by times.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
propertyName
```

Type: System..:..String
The name of the property.

times

Type: Moq..:..Times

The number of times a method is allowed to be called.

Type Parameters

TProperty

The type of the property.

IProtectedMock<(Of <(<'TMock>)>)> Interface
Moq.Protected Namespace

-

□ C#

Moq

IProtectedMock<(Of <(<'TMock>)>)>..:..VerifySet<(Of <(<'TProperty>)>)> Method

IProtectedMock<(Of <(<'TMock>)>)> Interface See Also Send Feedback

Specifies a setup for an invocation on a property setter with the given propertyName.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

```
propertyName
```

Type: <u>System.::.String</u>
The name of the property.

times

Type: Moq..:..Times

The number of times a method is allowed to be called.

value

Type: <u>System..:..Object</u> The property value.

Type Parameters

TProperty

The type of the property. If argument matchers are used, remember to use ItExpr rather than It.

Exceptions

Exception

Condition

Moq..:..MockException The invocation was not call the times specified by times.

IProtectedMock<(Of <(<'TMock>)>)> Interface
Moq.Protected Namespace

-

□ C#

Moq

ItExpr Class

Members See Also Send Feedback

Allows the specification of a matching condition for an argument in a protected member setup, rather than a specific argument value. "ItExpr" refers to the argument being matched.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

$\blacksquare Syntax$

C#

public static class ItExpr

Remarks

Use this variant of argument matching instead of <u>It</u> for protected setups.

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

Inheritance Hierarchy

System..:..Object

Moq.Protected..:..ItExpr

ItExpr Members Moq.Protected Namespace

☐ 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
ë ∳S	<u>Is<(Of</u> <<' <u>(TValue>)>>)</u>	
=\$S	<u>IsAny<(Of</u> <<' <u>(TValue</u> >)>>)	Matches any value of the given TValue type.
ĕŵS	<u>IsInRange<(Of</u> <<' <u>(TValue>)>></u>)	Matches any value that is in the range specified.
ĕŵS	<u>IsNull<(Of</u> <<' <u>(TValue>)>></u>)	Matches a null value of the given TValue type.
ë ∳S	IsRegex(String)	Matches a string argument if it matches the given regular expression pattern.
=\$ \$	<pre>IsRegex(String, RegexOptions)</pre>	Matches a string argument if it matches the given regular expression pattern.

ItExpr Class Moq.Protected Namespace

☐ 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
ë ∳S	<u>Is<(Of</u> <<' <u>(TValue>)>>)</u>	
=\$S	<u>IsAny<(Of</u> <<' <u>(TValue</u> >)>>)	Matches any value of the given TValue type.
ĕŵS	<u>IsInRange<(Of</u> <<' <u>(TValue>)>></u>)	Matches any value that is in the range specified.
ĕŵS	<u>IsNull<(Of</u> <<' <u>(TValue>)>></u>)	Matches a null value of the given TValue type.
ë ∳S	IsRegex(String)	Matches a string argument if it matches the given regular expression pattern.
=\$ \$	<pre>IsRegex(String, RegexOptions)</pre>	Matches a string argument if it matches the given regular expression pattern.

ItExpr Class Moq.Protected Namespace

```
-
```

□ C#

Moq

ItExpr..:..Is<(Of <(<'TValue>)>)> Method

ItExpr Class See Also Send Feedback

[Missing <summary> documentation for "M:Moq.Protected.ItExpr.Is``1(System.Linq.Expressions.Expression{System.Func`2})"]

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

match

```
Type: System.Linq.Expressions..::.Expression < (Of < (< 'Func< (Of < 'Substituting (System Linq.Expression)).
```

Type Parameters

TValue

ItExpr Class Moq.Protected Namespace

-

□ C#

Moq

ItExpr..:..IsAny<(Of <(<'TValue>)>)> Method

<u>ItExpr Class Example See Also Send Feedback</u>

Matches any value of the given TValue type.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public static <u>Expression</u> IsAny<TValue>()

Type Parameters

TValue

Type of the value.

Remarks

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

Examples

C#

```
// Throws an exception for a call to Remove with any string value.
mock.Protected()
    .Setup("Remove", ItExpr.IsAny<string>())
    .Throws(new InvalidOperationException());
```

ItExpr Class Moq.Protected Namespace

_

□ C#

Moq

ItExpr..:..IsInRange<(Of <(<'TValue>)>)> Method

<u>ItExpr Class Example See Also Send Feedback</u>

Matches any value that is in the range specified.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

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

ItExpr Class Moq.Protected Namespace

_

□ C#

Moq

ItExpr..::.IsNull < (Of < (< 'TValue >) >) > Method

<u>ItExpr Class Example See Also Send Feedback</u>

Matches a null value of the given TValue type.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public static <u>Expression</u> IsNull<TValue>()

Type Parameters

TValue

Type of the value.

Remarks

Required for protected mocks as the null value cannot be used directly as it prevents proper method overload selection.

Examples

C#

```
// Throws an exception for a call to Remove with a null string value
mock.Protected()
    .Setup("Remove", ItExpr.IsNull<string>())
    .Throws(new InvalidOperationException());
```

ItExpr Class Moq.Protected Namespace

☐ C#
☐ Include Protected Members
☐ Include Inherited Members
Moq
ItExpr..:..IsRegex Method

<u>ItExpr Class See Also Send Feedback</u>

Overload List

	Name	Description
= QS	IsRegex(String)	Matches a string argument if it matches the given regular expression pattern.
= QS	IsRegex(String, RegexOptions)	Matches a string argument if it matches the given regular expression pattern.

ItExpr Class
ItExpr Members
Moq.Protected Namespace

_

□ C#

Moq

ItExpr..:..IsRegex Method (String)

ItExpr Class 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

regex

Type: <u>System.....String</u>

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# Copy

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

ItExpr Class
IsRegex Overload
Moq.Protected Namespace

-

□ C#

Moq

ItExpr..:..IsRegex Method (String, RegexOptions)

ItExpr Class 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: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

regex

Type: System..:..String

The pattern to use to match the string argument value.

options

Type: <u>System.Text.RegularExpressions..:..RegexOptions</u>

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

ItExpr Class
IsRegex Overload
Moq.Protected Namespace

_

□ C#

Moq

ProtectedExtension Class

Members See Also Send Feedback

Enables the Protected() method on $\underline{\text{Mock}} \leq (\underline{\text{Of}} \leq (\underline{\text{C'T}}) \geq) \geq$, allowing setups to be set for protected members by using their name as a string, rather than strongtyping them which is not possible due to their visibility.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

\blacksquare Syntax

C#

public static class ProtectedExtension

Inheritance Hierarchy

System..:..Object

Moq.Protected..:..ProtectedExtension

ProtectedExtension Members Moq.Protected Namespace

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

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

Methods

Name

Description

\$5

<u>Protected \leq (Of $\leq \leq$ '(T \geq) $\geq \geq$)</u> Enable protected setups for the mock.

ProtectedExtension Class Moq.Protected Namespace

□ 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

Name

Description

\$5

<u>Protected \leq (Of $\leq \leq$ '(T \geq) $\geq \geq$)</u> Enable protected setups for the mock.

ProtectedExtension Class Moq.Protected Namespace

-

□ C#

Moq

Protected Extension...:..Protected < (Of < (< 'T >) >) > Method

ProtectedExtension Class See Also Send Feedback

Enable protected setups for the mock.

Namespace: Moq.Protected

Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)

Syntax

C#

Parameters

mock

Type: Moq..:..Mock<(Of <(<'T>)>)>
The mock to set the protected setups on.

Type Parameters

T

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

ProtectedExtension Class
Moq.Protected Namespace