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 online quickstarts for more examples than those available in this code documentation.

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

You can also read blog entries from around the world about Moq.
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>It</td>
<td>Allows the specification of a matching condition for an argument in a method invocation, rather than a specific argument value. &quot;It&quot; refers to the argument being matched.</td>
</tr>
<tr>
<td>Mock</td>
<td>Base class for mocks and static helper class with methods that apply to mocked objects, such as Get(Of Of (&lt;'T'&gt;)&gt;(T) to retrieve a Mock(Of Of (&lt;'T'&gt;)&gt;(T) from an object instance.</td>
</tr>
<tr>
<td>Mock(Of (&lt;'T'&gt;)&gt;)</td>
<td>Provides a mock implementation of T.</td>
</tr>
<tr>
<td>MockException</td>
<td>Exception thrown by mocks when setups are not matched, the mock is not properly setup, etc.</td>
</tr>
<tr>
<td>MockExtensions</td>
<td>Provides additional methods on mocks. <strong>Obsolete.</strong></td>
</tr>
<tr>
<td>MockFactory</td>
<td>Utility factory class to use to construct multiple mocks when consistent verification is desired for all of them.</td>
</tr>
<tr>
<td>MockRepository</td>
<td>Utility repository class to use to construct multiple mocks when consistent verification is desired for all of them.</td>
</tr>
<tr>
<td>Mocks</td>
<td>Allows querying the universe of mocks for those that behave according to the LINQ query specification.</td>
</tr>
<tr>
<td>MockSequence</td>
<td>Helper class to setup a full trace between many mocks</td>
</tr>
<tr>
<td>MockSequenceHelper</td>
<td>define nice api</td>
</tr>
<tr>
<td>SequenceExtensions</td>
<td>Helper for sequencing return values in the same method.</td>
</tr>
</tbody>
</table>
## Structures

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times</td>
<td>Defines the number of invocations allowed by a mocked method.</td>
</tr>
</tbody>
</table>
## Enumerations

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

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

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public enum Default

Value
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td></td>
</tr>
<tr>
<td>Mock</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Moq Namespace

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

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
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.. System.Object
    Moq.. It
See Also

It Members
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Is&lt;(Of &lt;'(TValue)&gt;)&gt;</code></td>
<td>Matches any value of the given TValue type.</td>
</tr>
<tr>
<td><code>IsAny&lt;(Of &lt;'(TValue)&gt;)&gt;</code></td>
<td>Matches any value that is in the range specified.</td>
</tr>
<tr>
<td><code>IsInRange&lt;(Of &lt;'(TValue)&gt;)&gt;</code></td>
<td>Matches any value that is in the range specified.</td>
</tr>
<tr>
<td><code>IsRegex(String)</code></td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
<tr>
<td><code>IsRegex(String, RegexOptions)</code></td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
</tbody>
</table>
See Also

**It Class**

**Moq Namespace**

Send comments on this topic to moqdisc@googlegroups.com
The **It** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is&lt;(Of&lt;br&gt;&quot;&lt;(TValue)&gt;&gt;)</td>
<td>Matches any value of the given TValue type.</td>
</tr>
<tr>
<td>IsAny&lt;(Of&lt;br&gt;&quot;&lt;(TValue)&gt;&gt;)</td>
<td>Matches any value that is in the range specified.</td>
</tr>
<tr>
<td>IsInRange&lt;(Of&lt;br&gt;&quot;&lt;(TValue)&gt;&gt;)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
<tr>
<td>IsRegex(String)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
<tr>
<td>IsRegex(String, RegexOptions)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
</tbody>
</table>
See Also

It Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
It...\text{Is\langle(Of \langle\langle T\text{Value}\rangle\rangle\rangle\rangle)} Method

[Missing \textit{<summary>} documentation for

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

**C#**

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

**Parameters**

match

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

TValue
See Also

It Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
**It.IsAny(Of (TValue)) Method**

Matches any value of the given TValue type.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
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 => x.Remove(It.IsAny<string>())).Throws(new InvalidOperationException());
See Also

It Class
Moq Namespace

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

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
**Syntax**

**C#**

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

**Parameters**

*from*

Type: TValue  
The lower bound of the range.

*to*

Type: TValue  
The upper bound of the range.

*rangeKind*

Type: Moq:::Range  
The kind of range. See Range.
Type Parameters

TValue
  Type of the argument to check.
Examples

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

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

It Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
It is not clear what the text is referring to. It could be discussing a method named 'IsRegex' in the context of C# programming. The text is cut off and does not provide enough context to determine the meaning accurately.
## Overload List

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

It Class
It Members
Moq Namespace

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

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
**Syntax**

C#

```csharp
public static string IsRegex(
            string regex
)
```

**Parameters**

regex

Type: `System.String`

The pattern to use to match the string argument value.
Examples

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

C#  

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

It Class
IsRegex Overload
Moq Namespace

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

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

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

C#

```csharp
public static string IsRegex(
    string regex,
    RegexOptions options
)
```

Parameters

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

options
Type: `System.Text.RegularExpressions::RegexOptions`
The options used to interpret the pattern.
Examples

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

C#

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

It Class
IsRegex Overload
Moq Namespace

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

Base class for mocks and static helper class with methods that apply to mocked objects, such as `Get(Of <<(T)>>>(T)` to retrieve a `Mock(Of <<(T)>>)` from an object instance.

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

C#

public abstract class Mock : IHideObjectMembers
Inheritance Hierarchy

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

Mock Members
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mock</td>
<td>Initializes a new instance of the Mock class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>As&lt;(Of &lt;&lt;'(TInterface)&gt;&gt;)&gt;</code></td>
<td>Adds an interface implementation to the mock, allowing setups to be specified for it.</td>
</tr>
<tr>
<td><code>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</code></td>
<td>Retrieves the mock object for the given object instance.</td>
</tr>
<tr>
<td><code>Of&lt;(Of &lt;&lt;'(T))&gt;&gt;())()</code></td>
<td>Creates a mock object of the indicated type.</td>
</tr>
<tr>
<td><code>(Expression&lt;(Of &lt;&lt;'(Func&lt;(Of &lt;&lt;'(T, TResult))&gt;&gt;)&gt;&gt;))&gt;</code></td>
<td>Returns the mocked object value.</td>
</tr>
<tr>
<td><code>OnGetObject</code></td>
<td>Verifies that all verifiable expectations have been met.</td>
</tr>
<tr>
<td><code>SetReturnsDefault&lt;(Of &lt;&lt;'(TReturn)&gt;&gt;)</code></td>
<td>Verifies all expectations regardless of whether they have been flagged as verifiable.</td>
</tr>
<tr>
<td><code>Verify</code></td>
<td></td>
</tr>
<tr>
<td><code>VerifyAll</code></td>
<td></td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior</td>
<td>Behavior of the mock, according to the value set in the constructor.</td>
</tr>
<tr>
<td></td>
<td>Whether the base member virtual implementation will be called for mocked</td>
</tr>
<tr>
<td></td>
<td>classes if no setup is matched. Defaults to false in Visual Basic.</td>
</tr>
<tr>
<td>CallBase</td>
<td>Specifies the behavior to use when returning default values for unexpected</td>
</tr>
<tr>
<td></td>
<td>invocations on loose mocks.</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Gets the mocked object instance.</td>
</tr>
</tbody>
</table>
See Also

Mock Class
Moq Namespace

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

Mock Class See Also Send Feedback

Initializes a new instance of the Mock class.

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

C#

protected Mock()
See Also

Mock Class
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![As](Of &lt;&lt;'(TInterface)&gt;)&gt;&gt;()</td>
<td>Adds an interface implementation to the mock, allowing setups to be specified for it.</td>
</tr>
<tr>
<td>![Get](Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>Retrieves the mock object for the given object instance.</td>
</tr>
<tr>
<td>![Of](Of &lt;&lt;'(T)&gt;&gt;())()()()</td>
<td>Creates an mock object of the indicated type.</td>
</tr>
<tr>
<td>![OnGetObject](Expression(Of &lt;&lt;'(Func&lt;(Of &lt;&lt;'(T, TResult)&gt;)&gt;&gt;)&gt;&gt;))</td>
<td>Returns the mocked object value.</td>
</tr>
<tr>
<td>![SetReturnsDefault](Of &lt;&lt;'(TReturn)&gt;&gt;)</td>
<td>Verifies that all verifiable expectations have been met.</td>
</tr>
<tr>
<td>![Verify](</td>
<td>Verifies all expectations regardless of whether they have been flagged as verifiable.</td>
</tr>
<tr>
<td>![VerifyAll](</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Mock Class
Moq Namespace

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

public virtual Mock<TInterface> As<TInterface>()
where TInterface : class
Type Parameters

TInterface
Type of interface to cast the mock to.
Remarks

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

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

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

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

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

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

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)
**Syntax**

C#

```csharp
public static Mock<T> Get<T>(
    T mocked
)
where T : class
```

**Parameters**

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

T
Type of the mock to retrieve. Can be omitted as it's inferred from the object instance passed in as the mocked instance.

Return Value

The mock associated with the mocked object.
Examples

The following example shows how to add a new setup to an object instance which is not the original \texttt{Mock\langle\text{Of}\langle\langle'T\rangle\rangle\rangle} but rather the object associated with it:

\begin{verbatim}
// Typed instance, not the mock, is retrieved from some test API.
HttpContextBase context = GetMockContext();

// context.Request is the typed object from the "real" API
// so in order to add a setup to it, we need to get
// the mock that "owns" it
Mock<HttpRequestBase> request = Mock.Get(context.Request);
mock.Setup(req => req.AppRelativeCurrentExecutionFilePath)
  .Returns(tempUrl);
\end{verbatim}
## Exceptions

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

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock:::Of Method
Mock Class See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Of&lt;(Of &lt;&lt;'T'&gt;)&gt;&gt;()0000</code></td>
<td>Creates a mock object of the indicated type.</td>
</tr>
<tr>
<td><code>Of&lt;(Of &lt;&lt;'T'&gt;)&gt;&gt;()(Expression&lt;(Of &lt;&lt;'Func&lt;(Of &lt;&lt;'T', TResult'&gt;)&gt;&gt;()&gt;&gt;()&gt;&gt;))</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

Mock Class
Mock Members
Moq Namespace

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

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

Mock 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)
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.
See Also

Mock Class
Of Overload
Moq Namespace

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

Moq

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

Mock Class See Also Send Feedback

[Missing <summary> documentation for

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

**C#**

```csharp
public static T Of<T>(
    Expression<Func> predicate
)
where T : class
```

### Parameters

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

T
See Also

Mock Class
Of Overload
Moq Namespace

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

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

C#

protected abstract Object OnGetObject()
See Also

Mock Class
Moq Namespace

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

Moq

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

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

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

C#

public void SetReturnsDefault<TReturn>(
    TReturn value
)

Parameters

value
    Type: TReturn
<table>
<thead>
<tr>
<th>Type Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>TReturn</td>
</tr>
</tbody>
</table>
See Also

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Verifies that all verifiable expectations have been met.

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

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

```csharp
var mock = new Mock<IWarehouse>();
this.Setup(x => x.HasInventory(TALISKER, 50)).Verifiable().Returns(true);
...  
// other test code  
...  
// Will throw if the test code has didn't call HasInventory.
this.Verify();
```
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq.....MockException</td>
<td>Not all verifiable expectations were met.</td>
</tr>
</tbody>
</table>
See Also

Mock Class
Moq Namespace

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

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

```csharp
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

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

Mock Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `Mock` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior</td>
<td>Behavior of the mock, according to the value set in the constructor.</td>
</tr>
<tr>
<td>CallBase</td>
<td>Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to falseFalsefalsefalse (False in Visual Basic).</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Specifies the behavior to use when returning default values for unexpected invocations on loose mocks.</td>
</tr>
<tr>
<td>Object</td>
<td>Gets the mocked object instance.</td>
</tr>
</tbody>
</table>
See Also

Mock Class
Moq Namespace

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

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

**Namespace**: Moq
**Assembly**: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

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

Mock Class
Moq Namespace

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

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

C#

public virtual bool CallBase { get; set; }
See Also

Mock Class
Moq Namespace

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

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)
Syntax

C#

public virtual DefaultValue DefaultValue { get; set; }
See Also

Mock Class
Moq Namespace

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

Mock..::..Object Property

See Also

Gets the mocked object instance.

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

C#

```csharp
public Object Object { get; }
```
See Also

Mock Class
Moq Namespace

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

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
C#

public class Mock<T> : 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 `MockBehavior` that can be passed to the `Mock<Of <('T')>>(MockBehavior)` constructor.
Examples

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

C#

// 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 It class to specify conditions for arguments instead of specific values:

C#

// 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>)}
See Also

Mock<(<T>)> Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `Mock<Of (<'T'>)>` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Mock(Of (&lt;T&gt;))()</code></td>
<td>Initializes an instance of the mock with default behavior.</td>
</tr>
<tr>
<td><code>Mock(Of (&lt;T&gt;))&gt;(array&lt;Object&gt;[][])</code></td>
<td>Initializes an instance of the mock with default behavior and with the given constructor arguments for the class. (Only valid when T is a class)</td>
</tr>
<tr>
<td><code>Mock(Of (&lt;T&gt;))&gt;(MockBehavior)</code></td>
<td>Initializes an instance of the mock with the specified behavior.</td>
</tr>
<tr>
<td><code>Mock(Of (&lt;T&gt;))&gt;(MockBehavior, array&lt;Object&gt;[][])</code></td>
<td>Initializes an instance of the mock with a specific behavior with the given constructor arguments for the class.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>As&lt;&lt;'(TInterface))&gt;&gt;</td>
<td>Adds an interface implementation to the mock, allowing setups to be specified for it. (Inherited from Mock.)</td>
</tr>
<tr>
<td>OnGetObject</td>
<td>Returns the mocked object value. (Overrides Mock::&lt;OnGetObject()()&gt;.)</td>
</tr>
<tr>
<td>Raise&lt;Action&lt;&lt;'(T)&gt;&gt;, EventArgs&gt;</td>
<td>Raises the event referenced in eventExpression using the given args argument.</td>
</tr>
<tr>
<td>Raise&lt;Action&lt;&lt;'(T)&gt;&gt;, array&lt;Object&gt;[][])</td>
<td>Raises the event referenced in eventExpression using the given args argument for a non-EventHandler typed event.</td>
</tr>
<tr>
<td>SetReturnsDefault&lt;&lt;'(TReturn)&gt;&gt;</td>
<td>(Inherited from Mock.)</td>
</tr>
<tr>
<td>Setup&lt;Expression&lt;&lt;'(Action&lt;&lt;'(T)&gt;&gt;)&gt;&gt;())&gt;</td>
<td>Specifies a setup on the mocked type for a call to to a void method.</td>
</tr>
<tr>
<td>Setup&lt;&lt;'(TResult)&gt;&gt;</td>
<td>Specifies that the all properties on the mock should have &quot;property behavior&quot;, meaning that setting its value will cause it to be saved and later returned when the property is requested. (this is also known as &quot;stubbing&quot;). The default value for each property will be the one generated as specified by the DefaultValue property for the mock.</td>
</tr>
<tr>
<td>SetupAllProperties</td>
<td></td>
</tr>
</tbody>
</table>
SetupGet(Of <<(TProperty)>>)

SetupProperty(Of <<(TProperty)>>)(Expression(Of <<(Func(Of <<(T, TResult)>>)>>)))

SetupProperty(Of <<(TProperty)>>)(Expression(Of <<(Func(Of <<(T, TResult)>>)>>, Boolean))

SetupSet(Action(Of <<(T)>>))

SetupSet(Of <<(TProperty)>>)(Action(Of <<(T)>>))

Verify()()

Verify(Expression(Of <<(Action(Of <<(T)>>)>>)>>))

Verify(Expression(Of <<(Action(Of <<(T)>>)>>)>>, Times)

Verify(Expression(Of <<(Action(Of <<(T)>>)>>)>>, String)

Verify(Expression(Of ...)

Verifies that all verifiable expectations have been met. (Inherited from Mock.)

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

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

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

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

VerifySet(Action(Of(Of(T)>)>>), Times, String)

VerifyGet(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(Of(O
**VerifySet** *(Action<Of <<(T)>>, Times, String)* the mock, specifying a failure message.
### Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>InSequence</strong>&lt; Of &lt;&lt;(TMock)&gt;&gt;</td>
<td>Perform an expectation in the trace. (Defined by <a href="#">MockSequenceHelper</a>).</td>
</tr>
<tr>
<td><strong>Protected</strong>&lt; Of &lt;&lt;(T)&gt;&gt;</td>
<td>Enable protected setups for the mock. (Defined by <a href="#">ProtectedExtension</a>).</td>
</tr>
<tr>
<td><strong>SetupSequence</strong>&lt; Of &lt;&lt;(TMock, TResult)&gt;&gt;</td>
<td>(Defined by <a href="#">SequenceExtensions</a>).</td>
</tr>
<tr>
<td><strong>SetupSet</strong>&lt; Of &lt;&lt;(T, TProperty)&gt;&gt;</td>
<td>(Defined by <a href="#">MockExtensions</a>).</td>
</tr>
<tr>
<td><strong>VerifySet</strong>&lt; Of &lt;&lt;(T, TProperty)&gt;&gt;</td>
<td>Overloaded. (Defined by <a href="#">MockExtensions</a>).</td>
</tr>
<tr>
<td><strong>VerifySet</strong>&lt; Of &lt;&lt;(T, TProperty)&gt;&gt;</td>
<td>Overloaded. (Defined by <a href="#">MockExtensions</a>).</td>
</tr>
<tr>
<td><strong>VerifySet</strong>&lt; Of &lt;&lt;(T, TProperty)&gt;&gt;</td>
<td>Overloaded. (Defined by <a href="#">MockExtensions</a>).</td>
</tr>
<tr>
<td><strong>VerifySet</strong>&lt; Of &lt;&lt;(T, TProperty)&gt;&gt;</td>
<td>Overloaded. (Defined by <a href="#">MockExtensions</a>).</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior</td>
<td>Behavior of the mock, according to the value set in the constructor.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Mock</a>).</td>
</tr>
<tr>
<td>CallBase</td>
<td>Whether the base member virtual implementation will be called for mocked classes if no setup is matched. Defaults to <code>false</code> (False in Visual Basic). (Inherited from <a href="#">Mock</a>).</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Specifies the behavior to use when returning default values for unexpected invocations on loose mocks. (Inherited from <a href="#">Mock</a>).</td>
</tr>
<tr>
<td>Object</td>
<td>Exposes the mocked object instance.</td>
</tr>
</tbody>
</table>
See Also

Mock(Of (T)> Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock<(Of <('T')?>)> Constructor

Mock<(Of <('T')?>)> Class See Also Send Feedback
## Overload List

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

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

Send comments on this topic to moqdisc@googlegroups.com
Mock(Of<'T>)}> Constructor

Initializes an instance of the mock with default behavior.

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

C#

public Mock()
Examples

C#

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

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

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

Moq

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

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

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

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

C#

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

### Parameters

**args**

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

Optional constructor arguments if the mocked type is a class.
Remarks

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

C#

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

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

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

Mock Class Example See Also Send Feedback

Initializes an instance of the mock with the specified behavior.

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

C#

public Mock(
    MockBehavior behavior
)

Parameters

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

C#

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

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

Send comments on this topic to moqdisc@googlegroups.com
C# Moq
Mock<Of <('T')>> Constructor (MockBehavior, array<Object>[]()[[]])

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

Initializes an instance of the mock with a specific behavior 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#

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

Parameters

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

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

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

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

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

Send comments on this topic to moqdisc@googlegroups.com
The `Mock<Of <('T')>>` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>As&lt;('TInterface')&gt;</td>
<td>Adds an interface implementation to the mock, allowing setups to be specified for it. (Inherited from <strong>Mock</strong>.)</td>
</tr>
<tr>
<td>OnGetObject</td>
<td>Returns the mocked object value. (Overrides <strong>Mock</strong>::&lt;&gt;'Mock::OnGetObject()'().)</td>
</tr>
<tr>
<td>Raise(Action&lt;('T')&gt;, EventArgs)</td>
<td>Raises the event referenced in eventExpression using the given args argument.</td>
</tr>
<tr>
<td>Raise(Action&lt;('T')&gt;, array&lt;Object&gt;[][])</td>
<td>Raises the event referenced in eventExpression using the given args argument for a non-EventHandler typed event.</td>
</tr>
<tr>
<td>SetReturnsDefault&lt;('TReturn')&gt;</td>
<td>(Inherited from <strong>Mock</strong>.) Specifies a setup on the mocked type for a call to to a void method.</td>
</tr>
<tr>
<td>Setup(Expression&lt;('Action&lt;('T')&gt;')&gt;)</td>
<td>Specifies that the all properties on the mock should have &quot;property behavior&quot;, meaning that setting its value will cause it to be saved and later returned when the property is requested. (this is also known as &quot;stubbing&quot;). The default value for each property will be the one generated as specified by the <strong>DefaultValue</strong> property for the mock.</td>
</tr>
<tr>
<td>Setup&lt;('TResult')&gt;</td>
<td></td>
</tr>
<tr>
<td>(Expression&lt;('Func&lt;('T', 'TResult')&gt;')&gt;)</td>
<td></td>
</tr>
<tr>
<td>SetupAllProperties</td>
<td></td>
</tr>
</tbody>
</table>
SetupGet<(Of <<'(TProperty>>)>>()

SetupProperty<(Of
"'(TProperty>)>>(Expression<(Of
"'(Func<(Of <<'T,
TResult>)>>)>>)>)

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

SetupSet(Action<(Of <<'(T)>>>))

SetupSet<(Of <<'(TProperty>)>>)(Action<(Of <<'(T)>>)>))

Verify()()

Verify(Expression<(Of
"'(Action<(Of <<'(T)>>)>>)>>))

Verify(Expression<(Of
"'(Action<(Of <<'(T)>>)>>)>>), Times)

Verify(Expression<(Of
"'(Action<(Of <<'(T)>>)>>)>>), String)

Verify(Expression<(Of
"'(Action<(Of <<'(T)>>)>>)>>)

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

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

Verifies that all verifiable expectations have been met. (Inherited from Mock.)

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

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

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

Verifies that a specific invocation matching the given expression
was performed on the mock, specifying a failure error message. Use in conjunction with the default **Loose**.

**VerifySet**

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

**Verify**

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

**VerifyAll**

Verifies that a property was set on the mock.
- **VerifySet**(`Action<Of <<'(T)>>, Times, String>`), the mock, specifying a failure message.

- **When**
## Extension Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InSequence(Of &lt;&lt;(TMock)&gt;&gt;)</code></td>
<td>Perform an expectation in the trace. (Defined by <code>MockSequenceHelper</code>.)</td>
</tr>
<tr>
<td><code>Protected(Of &lt;&lt;(T)&gt;&gt;)</code></td>
<td>Enable protected setups for the mock. (Defined by <code>ProtectedExtension</code>.)</td>
</tr>
<tr>
<td><code>SetupSequence(Of &lt;&lt;(TMock, TResult)&gt;&gt;)</code></td>
<td>(Defined by <code>SequenceExtensions</code>.)</td>
</tr>
<tr>
<td><code>SetupSet(Of &lt;&lt;(T, TProperty)&gt;&gt;)</code></td>
<td>(Defined by <code>MockExtensions</code>.)</td>
</tr>
<tr>
<td><code>VerifySet(Of &lt;&lt;(T, TProperty)&gt;&gt;)</code></td>
<td>Overloaded. (Defined by <code>MockExtensions</code>.)</td>
</tr>
<tr>
<td><code>VerifySet(Of &lt;&lt;(Expression(Of &lt;&lt;(Func(Of &lt;&lt;(T, TResult)&gt;&gt;)&gt;&gt;)&gt;&gt;, Boolean)</code>)</td>
<td>Overloaded. (Defined by <code>MockExtensions</code>.)</td>
</tr>
<tr>
<td><code>VerifySet(Of &lt;&lt;(Expression(Of &lt;&lt;(Func(Of &lt;&lt;(T, TResult)&gt;&gt;)&gt;&gt;)&gt;&gt;, Boolean)</code>)</td>
<td>Overloaded. (Defined by <code>MockExtensions</code>.)</td>
</tr>
<tr>
<td><code>VerifySet(Of &lt;&lt;(T, TProperty)&gt;&gt;)</code></td>
<td>Overloaded. (Defined by <code>MockExtensions</code>.)</td>
</tr>
</tbody>
</table>
See Also

Mock(Of T) Class
Moq Namespace

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

**Namespace:** Moq

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

C#

protected override Object OnGetObject()
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
Mock<(Of <('T')>)>..::Raise Method

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Raise(Action&lt;Of &lt;'(T)'&gt;, EventArgs)</code></td>
<td>Raises the event referenced in eventExpression using the given args argument.</td>
</tr>
<tr>
<td><code>Raise(Action&lt;Of &lt;'(T)'&gt;, array&lt;Object&gt;[][])</code></td>
<td>Raises the event referenced in eventExpression using the given args argument for a non-EventHandler typed event.</td>
</tr>
</tbody>
</table>
See Also

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

Send comments on this topic to moqdisc@googlegroups.com

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

C#

```csharp
public void Raise(
    Action<T> eventExpression,
    EventArgs args
)
```

Parameters

eventExpression
    Type: System...Action<((T)>)

args
    Type: System...EventArgs
Examples

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

```csharp
var mock = new Mock<IViewModel>();
mock.Raise(x => x.PropertyChanged -= null, new PropertyChangedEventArgs(
```
This example shows how to invoke an event with a custom event arguments class in a view that will cause its corresponding presenter to react by changing its state:

C#

```csharp
var mockView = new Mock<IOrdersView>();
var 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 order
Assert.NotNull(presenter.SelectedOrder);
Assert.Equal("moq", presenter.SelectedOrder.ProductName);
```
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System::..::ArgumentException</code></td>
<td>The <code>args</code> argument is invalid for the target event invocation, or the <code>eventExpression</code> is not an event attach or detach expression.</td>
</tr>
</tbody>
</table>
See Also

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

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

**C#**

```csharp
public void Raise(
    Action<T> eventExpression,
    params Object[] args
)
```

**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#

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

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System..::..ArgumentException</code></td>
<td>The args arguments are invalid for the target event invocation, or the <code>eventExpression</code> is not an event attach or detach expression.</td>
</tr>
</tbody>
</table>
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
Mock<Of <(T)>>...Setup Method
Mock<Of <(T)>> Class See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Setup(Expression(Of &lt;&lt;'(Action(Of &lt;&lt;'(T&gt;))&gt;&gt;))&gt;&gt;))</code></td>
<td>Specifies a setup on the mocked type for a call to a void method.</td>
</tr>
<tr>
<td><code>Setup(Of &lt;&lt;'(TResult&gt;)&gt;&gt;)</code></td>
<td></td>
</tr>
<tr>
<td><code>(Expression(Of &lt;&lt;'(Func(Of &lt;&lt;'(T, TResult&gt;)&gt;&gt;))&gt;&gt;))</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
Specifies a setup on the mocked type for a call to a void method.

**Namespace:** Moq

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

C#

```csharp
public ISetup<T> Setup(
    Expression<Action<T>> expression
)
```

**Parameters**

**expression**
Type: [System.Linq.Expressions..::.Expression](Of (<'Action')(Of (<'T'>)>)>)
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#  

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

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

Send comments on this topic to moqdisc@googlegroups.com
Mock<(Of <('T')>)>...Setup<(Of <('TResult')>)> Method
(Expression<(Of <('Func')>)>)
Mock<(Of <('T')>)> Class See Also Send Feedback

[Missing <summary> documentation for

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

C#

```csharp
public ISetup<T, TResult> Setup<TResult>(
    Expression<Func> expression
)
```

Parameters

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

TResult
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
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 theDefaultValue property for the mock.

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

C#

public Mock<T> SetupAllProperties()
Remarks

If the mock `DefaultValue` is set to `Mock`, the mocked default values will also get all properties setup recursively.
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
Mock<Of <('T')>>...SetupGet<Of <('TProperty')>> Method

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

[Missing <summary> documentation for
]

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

C#

public ISetupGetter<T, TProperty> SetupGet<TProperty>(
    Expression<Func> expression
)

Parameters

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

TProperty
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
Mock<Of (<'T'>)>>......SetupProperty Method
Mock<Of (<'T'>)>> Class See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetupProperty&lt;(Of &lt;&lt;'(TProperty&gt;')&gt;&gt;())(Expression&lt;(Of &lt;-'(Func&lt;(Of «('T, TResult'&gt;)&gt;)&gt;&gt;)&gt;&gt;))</td>
<td></td>
</tr>
<tr>
<td>SetupProperty&lt;(Of &lt;&lt;'(TProperty&gt;')&gt;&gt;())(Expression&lt;(Of &lt;-'(Func&lt;(Of «('T, TResult'&gt;)&gt;)&gt;&gt;)&gt;&gt;, Boolean)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
Mock<Of <(T)>::..SetupProperty<Of <TProperty>>() Method
(Expression<Of <Func>>()

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

[Missing <summary> documentation for

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

C#

```csharp
public Mock<T> SetupProperty<TProperty>(
    Expression<Func<TProperty>> property
)
```

Parameters

property
    Type: System.Linq.Expressions...::Expression(Of (Of (Func(Of (T, TResult)>)>)>)}

```csharp
public Mock<T> SetupProperty<TProperty>(
    Expression<Func<TProperty>> property
)
```
Type Parameters

TProperty
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
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

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

C#

public Mock<T> SetupProperty<TProperty>(
    Expression<Func> property,
    bool initialValue
)

Parameters

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

initialValue
    Type: System.....Boolean
Type Parameters

TProperty
See Also

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

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

Include Protected Members
Include Inherited Members

Moq
Mock<Of (<'T'>)> SETUPSET Method

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SetupSet&lt;(Of &lt;'(TProperty)&gt;)(Action&lt;(Of &lt;'(T)&gt;&gt;&gt;))</code></td>
<td>Specifies a setup on the mocked type for a call to to a property setter.</td>
</tr>
<tr>
<td><code>SetupSet(Action&lt;(Of &lt;'(T)&gt;&gt;))</code></td>
<td>Specifies a setup on the mocked type for a call to to a property setter.</td>
</tr>
</tbody>
</table>
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
Mock<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 a property setter.

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

C#

```csharp
public ISetupSetter<T, TProperty> SetupSet<TProperty>(
    Action<T> setterExpression
)
```

Parameters

setterExpression
Type: `System..::.Action`<Of<((<T>))>>
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);
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
Specifies a setup on the mocked type for a call to a property setter.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
**Syntax**

C#

```csharp
public ISetup<T> SetupSet(
    Action<T> setterExpression
)
```

**Parameters**

setterExpression
- Type: `System...Action<>`< Of `(<<T>>)`>
  - 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);
See Also

Mock(Of T>) Class
SetupSet Overload
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify()()()</td>
<td>Verifies that all verifiable expectations have been met. (Inherited from Mock.)</td>
</tr>
<tr>
<td>Verify(Expression&lt;(Of '&lt;(Action&lt;(Of '&quot;(T)&quot;&quot;&gt;&quot;&gt;&quot;&gt;&quot;)&gt;)&gt;)&gt;)</td>
<td>Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjunction with the default Loose.</td>
</tr>
<tr>
<td>Verify&lt;(Of '&quot;(TResult)&quot;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>Verify&lt;(Expression&lt;(Of '&quot;(Func&lt;(Of '&quot;(T, TResult)&quot;&gt;&gt;()&gt;&gt;()()&gt;)&gt;)</td>
<td></td>
</tr>
<tr>
<td>Verify&lt;(Expression&lt;(Of '&quot;(Action&lt;(Of '&quot;(T)&quot;&quot;&gt;&quot;&gt;&quot;&gt;&quot;)&gt;, Times))&gt;)</td>
<td>Verifies that a specific invocation matching the given expression was performed on the mock. Use in conjunction with the default Loose.</td>
</tr>
<tr>
<td>Verify&lt;(Expression&lt;(Of '&quot;(Action&lt;(Of '&quot;(T)&quot;&quot;&gt;&quot;&gt;&quot;&gt;&quot;)&gt;, String))&gt;)</td>
<td>Verifies that a specific invocation matching the given expression was performed on the mock, specifying a failure error message. Use in conjunction with the default Loose.</td>
</tr>
<tr>
<td>Verify&lt;(Of '&quot;(TResult)&quot;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>Verify&lt;(Expression&lt;(Of '&quot;(Func&lt;(Of '&quot;(T, TResult)&quot;&gt;&gt;()&gt;&gt;()()&gt;)&gt;)</td>
<td></td>
</tr>
<tr>
<td>Verify&lt;(Of '&quot;(TResult)&quot;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>Verify&lt;(Expression&lt;(Of '&quot;(Func&lt;(Of '&quot;(T, TResult)&quot;&gt;&gt;()&gt;&gt;()()&gt;)&gt;)</td>
<td></td>
</tr>
<tr>
<td>Verify&lt;(Expression&lt;(Of '&quot;(Func&lt;(Of '&quot;(T, TResult)&quot;&gt;&gt;()&gt;&gt;()()&gt;)&gt;)</td>
<td></td>
</tr>
<tr>
<td>Verify&lt;(Expression&lt;(Of '&quot;(Func&lt;(Of '&quot;(T, TResult)&quot;&gt;&gt;()&gt;&gt;()()&gt;)&gt;)</td>
<td></td>
</tr>
<tr>
<td>Verify&lt;(Expression&lt;(Of '&quot;(Action&lt;(Of '&quot;(T)&quot;&quot;&gt;&quot;&gt;&quot;&gt;&quot;)&gt;, Times, String))&gt;)</td>
<td>Verifies that a specific invocation matching the given expression was performed on the mock, specifying a failure error message. Use in conjunction with the default Loose.</td>
</tr>
</tbody>
</table>
Verify<(Of <<'(TResult>>)>
(Expression<(Of <<'(Func<(Of
<<(T, TResult>>)>>)>>)>>,
Boolean, T)
See Also

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

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

**Namespace:** Moq
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
## Syntax

**C#**

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

### Parameters

`expression`  
Type: `System.Linq.Expressions...:..Expression(Of (Of <'Action<Of <(<'T)>)>>)>

Expression to verify.
Examples

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

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

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

Mock(Of T) Class
Verify Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Moq
Mock<Of (<'T'>)>...Verify<Of (<'TResult'>)> Method
(Expression<Of (<'Func'>)>)
Mock<Of (<'T'>)> Class See Also Send Feedback

[Missing <summary> documentation for

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

C#

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

Parameters

description
Type: `System.Linq.Expressions..::..Expression<Of <(`Func<Of <-<T,
TResult>>)>>)">>`
Type Parameters

TResult
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
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 conjunction with the default Loose.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
## Syntax

### C#

```csharp
public void Verify(
    Expression<Action<T>> expression,
    Times times
)
```

### Parameters

**expression**
- Type: `System.Linq.Expressions.Expression(Of Action(Of 'T))`
- Expression to verify.

**times**
- Type: `Moq.Times`
- The number of times a method is allowed to be called.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq.....MockException</td>
<td>The invocation was not called the times specified by times.</td>
</tr>
</tbody>
</table>
See Also

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

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

**Namespace:** Moq

**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
## Syntax

**C#**

```csharp
public void Verify(
    Expression<Action<T>> expression,
    string failMessage
)
```

### Parameters

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

**failMessage**
- Type: `System.String`
- Message to show if verification fails.
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#
var mock = new Mock<IProcessor>();
// exercise mock
//... // Will throw if the test code didn't call Execute with a "ping" str
tmock.Verify(proc => proc.Execute("ping"));
```
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq::..MockException</td>
<td>The invocation was not performed on the mock.</td>
</tr>
</tbody>
</table>
See Also

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

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

public static class Moq
{
    public static Mock<T> Verify<TResult>(Expression<Func<Func<T, TResult>, Boolean>>, Boolean)
    {
        // Implementation details...
    }
}

[Missing <summary> documentation for
]

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#

```csharp
public void Verify<TResult>(
    Expression<Func> expression,
    bool times
)
```

Parameters

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

**times**
- Type: `System.....Boolean`
Type Parameters

TResult
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
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 conjunction with the default Loose.

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

C#

```csharp
public void Verify(
    Expression<Action<T>> expression,
    Times times,
    string failMessage
)
```

Parameters

expression

Type: System.Linq.Expressions..::..Expression<Of (Of (Action<Of (Of 'T)>)>)>>
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.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq......MockException</td>
<td>The invocation was not call the times specified by times.</td>
</tr>
</tbody>
</table>
See Also

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

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

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

C#

public void Verify<TResult>(
    Expression<Func<TResult>> expression,
    bool times,
    T failMessage
)

Parameters

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

times
    Type: System::Boolean

failMessage
    Type: T
Type Parameters

TResult
See Also

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

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

VerifyGet Method
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerifyGet&lt;&lt;'T&gt;(Of &lt;&lt;'(TProperty&gt;&gt;)(Expression&lt;&lt;'(Func&lt;&lt;'(T,TResult)&gt;&gt;)&gt;&gt;), Boolean)</td>
<td></td>
</tr>
<tr>
<td>VerifyGet&lt;&lt;'T&gt;(Of &lt;&lt;'(TProperty&gt;&gt;)(Expression&lt;&lt;'(Func&lt;&lt;'(T,TResult)&gt;&gt;)&gt;&gt;), Boolean)</td>
<td></td>
</tr>
<tr>
<td>VerifyGet&lt;&lt;'T&gt;(Of &lt;&lt;'(TProperty&gt;&gt;)(Expression&lt;&lt;'(Func&lt;&lt;'(T,TResult)&gt;&gt;)&gt;&gt;), Boolean, T)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
Mock<Of <('T')> >...VerifyGet<Of <('TProperty')> > Method
(Expression<Of <('Func')> > )

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

[Missing <summary> documentation for

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

C#

public void VerifyGet<TProperty>(
    Expression<Func<TResult>> expression
)

Parameters

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

TProperty
See Also

Mock(Of T) Class
VerifyGet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Mock\(\langle\langle\text{T}\rangle\rangle\)\)...\(\text{VerifyGet}\langle\langle\text{TProperty}\rangle\rangle\rangle\) Method
(\text{Expression}\langle\langle\langle\text{Func}\rangle\rangle\rangle\rangle, \text{Boolean})
\(\text{Mock}\langle\langle\text{T}\rangle\rangle\rangle\) Class

[Missing <summary> documentation for

\textbf{Namespace:} Moq
\textbf{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#**

```csharp
public void VerifyGet<TProperty>(
    Expression<Func> expression,
    bool failMessage
)
```

### Parameters

- **expression**
  Type: `System.Linq.Expressions.Expression<Func<T, TResult>>`

- **failMessage**
  Type: `System.Boolean`
Type Parameters

TProperty
See Also

Mock(Of T) Class
VerifyGet Overload
Moq Namespace

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

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

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

[Missing <summary> documentation for

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

C#

```csharp
public void VerifyGet<TProperty>(
    Expression<Func> expression,
    bool times,
    T failMessage
)
```

Parameters

description

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

times

Type: `System...;Boolean`

failMessage

Type: `T`
Type Parameters

TProperty
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
C#
Include Protected Members
Include Inherited Members
Moq
Mock<
(Of
(Of
(Of
(<'
<T>
>)
>)
>)
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerifySet(Action&lt;Of &lt;&lt;'(T))&gt;&gt;)</td>
<td>Verifies that a property was set on the mock.</td>
</tr>
<tr>
<td>VerifySet(Action&lt;Of &lt;&lt;'(T)&gt;&gt;, Times)</td>
<td>Verifies that a property was set on the mock.</td>
</tr>
<tr>
<td>VerifySet(Action&lt;Of &lt;&lt;'(T)&gt;&gt;, String)</td>
<td>Verifies that a property was set on the mock, specifying a failure message.</td>
</tr>
<tr>
<td>VerifySet(Action&lt;Of &lt;&lt;'(T)&gt;&gt;, Times, String)</td>
<td>Verifies that a property was set on the mock, specifying a failure message.</td>
</tr>
</tbody>
</table>
See Also

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

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

public void VerifySet(
    Action<T> setterExpression
)

Parameters

setterExpression
    Type: System..::.Action<Of<<'T'>>>
    Expression to verify.
Examples

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

C#

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

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

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

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

```csharp
public void VerifySet(
    Action<T> setterExpression,
    Times times
)
```

### Parameters

**setterExpression**
- Type: `System..::.Action<Of (<'T'>)>
- Expression to verify.

**times**
- Type: `Moq..::.Times
- The number of times a method is allowed to be called.`
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq.::::MockException</td>
<td>The invocation was not call the times specified by times.</td>
</tr>
</tbody>
</table>
See Also

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

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

```csharp
public void VerifySet(
    Action<T> setterExpression,
    string failMessage
)
```

Parameters

setterExpression
Type: `System.Action`<Of `(Of `<T`>`)>>
Expression to verify.

failMessage
Type: `System.String`
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#

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

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

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

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

public void VerifySet(
    Action<T> setterExpression,
    Times times,
    string failMessage
)

Parameters

setterExpression
Type: System...Action(Of (<'T'>)>)
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.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq::__MockException</td>
<td>The invocation was not call the times specified by times.</td>
</tr>
</tbody>
</table>
See Also

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

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

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#

class ISetupConditionResult<T> When(
    Func condition
)

Parameters

type: System:::Func(Of <'TResult>')

See Also

Mock(Of (T)>) Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `Mock<(Of <('T')>)>` type exposes the following members.
## Properties

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

Mock(Of (T)> Class
Moq Namespace

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

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public virtual T Object { get; }
See Also

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

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

Options to customize the behavior of the mock.

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

C#

public enum MockBehavior
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strict</td>
<td></td>
</tr>
<tr>
<td>Loose</td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Moq Namespace

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

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)
Syntax

C#

[SerializableAttribute]
public class MockException : Exception
Remarks

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

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

System..::..Object
System..::..Exception
Moq..::..MockException
See Also

MockException Members
Moq Namespace

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

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

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

MockException Class
Moq Namespace

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

```csharp
protected MockException(
    [System.Runtime.Serialization.SerializationInfo] info,
)
```

### Parameters

**info**

Type: `System.Runtime.Serialization.SerializationInfo`  
Serialization information.

**context**

Type: `System.Runtime.Serialization.StreamingContext`  
Streaming context.
See Also

MockException Class
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetObjectData</td>
<td>Supports the serialization infrastructure. (Overrides Exception::GetObjectData(SerializationInfo, StreamingContext).)</td>
</tr>
</tbody>
</table>
See Also

MockException Class
Moq Namespace

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

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

Parameters

info
Type: `System.Runtime.Serialization.SerializationInfo`
Serialization information.

collection
Type: `System.Runtime.Serialization.StreamingContext`
Streaming context.

Implements

`ISerializable`, `GetObjectData(SerializationInfo, StreamingContext)`
`Exception`, `GetObjectData(SerializationInfo, StreamingContext)`
See Also

MockException Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Provides additional methods on mocks.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public static class MockExtensions
Inheritance Hierarchy

System:::Object
Moq:::MockExtensions
See Also

MockExtensions Members
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetupSet&lt;(Of &lt;&lt;'(T, TProperty')&gt;&gt;)</td>
<td>Obsolete.</td>
</tr>
<tr>
<td>VerifySet&lt;(Of &lt;&lt;'(T, TProperty')&gt;&gt;())(Mock&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;())), Expression&lt;(Of &lt;&lt;'(Func&lt;(Of &lt;&lt;'(T, TResult')&gt;&gt;()&gt;&gt;))&gt;&gt;))</td>
<td>Obsolete.</td>
</tr>
<tr>
<td>VerifySet&lt;(Of &lt;&lt;'(T, TProperty')&gt;&gt;())(Mock&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;())), Expression&lt;(Of &lt;&lt;'(Func&lt;(Of &lt;&lt;'(T, TResult')&gt;&gt;()&gt;&gt;))&gt;&gt;, Boolean)</td>
<td>Obsolete.</td>
</tr>
<tr>
<td>VerifySet&lt;(Of &lt;&lt;'(T, TProperty')&gt;&gt;())(Mock&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;())), Expression&lt;(Of &lt;&lt;'(Func&lt;(Of &lt;&lt;'(T, TResult')&gt;&gt;()&gt;&gt;))&gt;&gt;, Boolean, T)</td>
<td>Obsolete.</td>
</tr>
</tbody>
</table>
See Also

MockExtensions Class
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetupSet&lt; Of &lt;&lt;'(T, TProperty)&gt;)&gt;&gt;</td>
<td>Obsolete.</td>
</tr>
<tr>
<td>VerifySet&lt; Of &lt;&lt;'(T, TProperty)&gt;&gt;&gt;(Mock&lt; Of &lt;&lt;'(T)&gt;&gt;), Expression&lt; Of &lt;&lt;'(Func&lt; Of &lt;(T, TResult)&gt;)&gt;&gt;)&gt;&gt;)</td>
<td>Obsolete.</td>
</tr>
<tr>
<td>VerifySet&lt; Of &lt;&lt;'(T, TProperty)&gt;&gt;&gt;(Mock&lt; Of &lt;&lt;'(T)&gt;&gt;), Expression&lt; Of &lt;&lt;'(Func&lt; Of &lt;(T, TResult)&gt;)&gt;&gt;)&gt;&gt;&gt;, Boolean)</td>
<td>Obsolete.</td>
</tr>
<tr>
<td>VerifySet&lt; Of &lt;&lt;'(T, TProperty)&gt;&gt;&gt;(Mock&lt; Of &lt;&lt;'(T)&gt;&gt;), Expression&lt; Of &lt;&lt;'(Func&lt; Of &lt;(T, TResult)&gt;)&gt;&gt;)&gt;&gt;&gt;, Boolean, T)</td>
<td>Obsolete.</td>
</tr>
</tbody>
</table>
See Also

MockExtensions Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockExtensions.SetupSet(Of (<'T, TProperty'>))> Method

MockExtensions Class See Also Send Feedback

[Missing <summary> documentation for

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
**Syntax**

**C#**

```csharp
[ObsoleteAttribute("Replaced by SetupSet(Action)")] public static ISetupSetter<T, TProperty> SetupSet<T, TProperty>(
    this Mock<T> mock,
    Expression<Func> expression
) where T : class
```

**Parameters**

mock
   Type: `Moq::Mock<Of (<'T'>)>`

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

T
TProperty
See Also

MockExtensions Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockExtensions...VerifySet Method

MockExtensions Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerifySet&lt;Of &lt;&lt;'(T, TProperty))&gt;&gt;)(Mock&lt;Of &lt;&lt;'(T)&gt;&gt;, Expression&lt;Of &lt;&lt;'(Func&lt;Of &lt;&lt;'(T, TResult))&gt;&gt;, Boolean)&gt;&gt;, Boolean, T</td>
<td>Obsolete.</td>
</tr>
</tbody>
</table>
See Also

MockExtensions Class
MockExtensions Members
Moq Namespace

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

[Missing <summary> documentation for

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

C#

```csharp
[ObsoleteAttribute("Replaced by VerifySet(Action)")]
public static void VerifySet<T, TProperty>(
    this Mock<T> mock,
    Expression<Func> expression
)
where T : class
```

Parameters

mock
Type: `Mock<Of (<'T'>)>`

equation
Type: `Expression<Of (<'Func<Of (<'T', TResult'>)>)>`
Type Parameters

T
TProperty
See Also

MockExtensions Class
VerifySet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
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

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#

```csharp
[ObsoleteAttribute("Replaced by VerifySet(Action, Times)")]
public static void VerifySet<T, TProperty>(
    this Mock<T> mock,
    Expression<Func> expression,
    bool times
)
where T : class
```

#### Parameters

- **mock**
  - Type: `Moq::Mock<Of<('T)>>`

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

- **times**
  - Type: `System::Boolean`
Type Parameters

T
TProperty
See Also

MockExtensions Class
VerifySet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
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

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
**Syntax**

**C#**

```csharp
[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: `Moq..::.Mock<Of <('T)>>`

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

**times**
Type: `System..::.Boolean`

**failMessage**
Type: `T`
Type Parameters

T
TProperty
See Also

MockExtensions Class
VerifySet Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Utility factory class to use to construct multiple mocks when consistent verification is desired for all of them.

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

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

C#

[ObsoleteAttribute("This class has been renamed to MockRepository. ^
false)]
public class MockFactory
Remarks

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

This factory class helps in that scenario by providing a simplified creation of multiple mocks with a default MockBehavior (unless overridden by calling Create(Of <<'(T)>>)(MockBehavior)) and posterior verification.
# Examples

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

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

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

// no need to call Verifiable() on the 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 MockException.
```

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

```csharp
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#

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

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

// specify setups

// exercise the mocks here

factory.Verify();
```
Inheritance Hierarchy

System..::..Object
Moq..::..MockFactory
  Moq..::..MockRepository
See Also

MockFactory Members
Moq Namespace
Moq::MockBehavior

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Create(Of &lt;&lt;'(T)&gt;&gt;)()()</code></td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time.</td>
</tr>
<tr>
<td><code>Create(Of &lt;&lt;'(T)&gt;&gt;) (array&lt;Object&gt;[][])</code></td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time and with the the given constructor arguments for the class.</td>
</tr>
<tr>
<td><code>Create(Of &lt;&lt;'(T)&gt;&gt;) (MockBehavior)</code></td>
<td>Creates a new mock with the given behavior.</td>
</tr>
<tr>
<td><code>Create(Of &lt;&lt;'(T)&gt;&gt;) (MockBehavior, array&lt;Object&gt;[][])</code></td>
<td>Creates a new mock with the given behavior and with the the given constructor arguments for the class.</td>
</tr>
<tr>
<td><code>CreateMock(Of &lt;&lt;'(T)&gt;&gt;)</code></td>
<td>Implements creation of a new mock within the factory.</td>
</tr>
<tr>
<td><code>Verify</code></td>
<td>Verifies all verifiable expectations on all mocks created by this factory.</td>
</tr>
<tr>
<td><code>VerifyAll</code></td>
<td>Verifies all verifiable expectations on all mocks created by this factory.</td>
</tr>
<tr>
<td><code>VerifyMocks</code></td>
<td>Invokes <code>verifyAction</code> for each mock in <code>Mocks</code>, and accumulates the resulting <code>MockVerificationException</code> that might be thrown from the action.</td>
</tr>
</tbody>
</table>
## Properties

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

MockFactory Class
Moq Namespace

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

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

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

C#

```csharp
public MockFactory(
    MockBehavior defaultBehavior
)
```

Parameters

defaultBehavior
Type: `Moq::MockBehavior`
The behavior to use for mocks created using the `Create<Of <<(T)>>QQ QQ` factory method if not overriden by using the `Create<Of <<(T)>>)` overload.
See Also

MockFactory Class
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Create&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;()&lt;&gt;()</code></td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time.</td>
</tr>
<tr>
<td><code>Create&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;(array&lt;Object&gt;[][])</code></td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time and with the the given constructor arguments for the class.</td>
</tr>
<tr>
<td><code>Create&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;(MockBehavior)</code></td>
<td>Creates a new mock with the given behavior.</td>
</tr>
<tr>
<td><code>Create&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;(MockBehavior, array&lt;Object&gt;[][])</code></td>
<td>Creates a new mock with the given behavior and with the the given constructor arguments for the class.</td>
</tr>
<tr>
<td><code>CreateMock&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;()</code></td>
<td>Implements creation of a new mock within the factory.</td>
</tr>
<tr>
<td><code>Verify</code></td>
<td>Verifies all verifiable expectations on all mocks created by this factory.</td>
</tr>
<tr>
<td><code>VerifyAll</code></td>
<td>Verifies all verifiable expectations on all mocks created by this factory.</td>
</tr>
<tr>
<td><code>VerifyMocks</code></td>
<td>Invokes <code>verifyAction</code> for each mock in <code>Mocks</code>, and accumulates the resulting <code>MockVerificationException</code> that might be thrown from the action.</td>
</tr>
</tbody>
</table>
See Also

MockFactory Class
Moq Namespace

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

MockFactory Class See Also Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)(){()()()}</td>
<td>Creates a new mock with the default MockBehavior specified at factory construction time.</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)(array&lt;Object&gt;[][][][])</td>
<td>Creates a new mock with the default MockBehavior specified at factory construction time and with the the given constructor arguments for the class.</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;) (MockBehavior)</td>
<td>Creates a new mock with the given behavior.</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;) (MockBehavior, array&lt;Object&gt;[][][])</td>
<td>Creates a new mock with the given behavior and with the the given constructor arguments for the class.</td>
</tr>
</tbody>
</table>
See Also

MockFactory Class
MockFactory Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockFactory..::.Create(Of '(<T>)')> Method

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

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
C#

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

T
   Type to mock.

Return Value

A new Mock<(Of <+T+>)>. 
Examples

C#

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

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

factory.VerifyAll();
```
See Also

MockFactory Class
Create Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Moq MockFactory::<...
MockFactory Class Example See Also Send Feedback
Creates a new mock with the default MockBehavior 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#

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

Parameters

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

T
    Type to mock.

Return Value

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

Remarks

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

C#  

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

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

factory.Verify();
```
See Also

MockFactory Class
Create Overload
Moq Namespace

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

Moq

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

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#

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

**Parameters**

**behavior**

Type: `Moq::::::::MockBehavior`

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

T
  Type to mock.

Return Value

A new `Mock<((Of<'(T)'>))>`. 
**Examples**

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

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

MockFactory Class
Create Overload
Moq Namespace

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

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

### Parameters

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

**args**
- Type: `array<Object>[]` (empty)
- Constructor arguments for mocked classes.
### Type Parameters

**T**

Type to mock.

### Return Value

A new `Mock<((Of <(>'T')>))>`. 
Remarks

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

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

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

MockFactory Class
Create Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockFactory..::.CreateMock(Of <(<'T'>)>)) Method

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

C#

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

**Parameters**

**behavior**
- Type: `Moq::MockBehavior`
  - The behavior for the new mock.

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

T

Type to mock.
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
MockFactory...Verify Method
MockFactory Class See Also Send Feedback

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

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

C#

public virtual void Verify()
## Exceptions

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

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

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

MockFactory Class  See Also  Send Feedback

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

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

C#

public virtual void VerifyAll()
### Exceptions

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

MockFactory Class
Moq Namespace
Mock...Verify()000

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

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

C#

protected virtual void VerifyMocks(
    Action<Mock> verifyAction
)

Parameters

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

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The MockFactory type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallBase</td>
<td>Whether the base member virtual implementation will be called for mocked</td>
</tr>
<tr>
<td></td>
<td>classes if no setup is matched. Defaults to falseFalsefalsefalse (False in</td>
</tr>
<tr>
<td></td>
<td>Visual Basic).</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>Specifies the behavior to use when returning default values for unexpected</td>
</tr>
<tr>
<td></td>
<td>invocations on loose mocks.</td>
</tr>
<tr>
<td>Mocks</td>
<td>Gets the mocks that have been created by this factory and that will get</td>
</tr>
<tr>
<td></td>
<td>verified together.</td>
</tr>
</tbody>
</table>
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public bool CallBase { get; set; }
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

public DefaultValue DefaultValue { get; set; }
See Also

MockFactory Class
Moq Namespace

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

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

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
protected internal IEnumerable<Mock> Mocks { get; }
See Also

MockFactory Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Utility repository class to use to construct multiple mocks when consistent verification is desired for all of them.

**Namespace:** [Moq](https://github.com/moq/Moq)
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
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 `MockBehavior` (unless overridden by calling `Create<(Of <<'(T)>>)(MockBehavior)`) and posterior verification.
Examples

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

C#

```csharp
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 MockException
```

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

C#

```csharp
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:

```csharp
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
See Also

MockRepository Members
Moq Namespace
Moq::MockBehavior

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MockRepository</td>
<td>Initializes the repository with the given defaultBehavior for newly created mocks from the repository.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;('T')&gt;&gt;()&gt;()()()()</td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;('T')&gt;&gt;())(array&lt;Object&gt;<a href=""></a>[[]])</td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time and with the the given constructor arguments for the class. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;('T')&gt;())&gt;(MockBehavior)</td>
<td>Creates a new mock with the given behavior. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;('T')&gt;())&gt;(MockBehavior, array&lt;Object&gt;<a href=""></a>[[]])</td>
<td>Creates a new mock with the given behavior and with the the given constructor arguments for the class. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>CreateMock&lt;(Of &lt;&lt;('T')&gt;())&gt;</td>
<td>Implements creation of a new mock within the factory. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>Of&lt;(Of &lt;&lt;('T')&gt;&gt;()&gt;()()()()</td>
<td>Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.</td>
</tr>
<tr>
<td>Of&lt;(Of &lt;&lt;('T')&gt;())&gt;(Expression&lt;(Of &lt;&lt;'(Func&lt;(Of (('T, TResult&gt;))&gt;()())()()))&gt;())</td>
<td>Creates an mock object of the indicated type.</td>
</tr>
<tr>
<td>OneOf&lt;(Of &lt;&lt;('T')&gt;())&gt;()()()()</td>
<td></td>
</tr>
<tr>
<td>OneOf&lt;(Of &lt;&lt;('T')&gt;())&gt;(Expression&lt;(Of &lt;&lt;'(Func&lt;(Of (('T, TResult&gt;))&gt;()())()()))&gt;())</td>
<td></td>
</tr>
</tbody>
</table>
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 inMocks, and accumulates the resulting MockVerificationException that might be thrown from the action.
(Inherited from MockFactory.)
## Properties

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

MockRepository Class
Moq Namespace

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

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#

public MockRepository(  
    MockBehavior defaultBehavior
)

Parameters

defaultBehavior
Type: MockBehavior
The behavior to use for mocks created using the Create(Of <<(T)>>)QQ QQ repository method if not overriden by using the Create(Of <<(T)>>)QQ QQ (MockBehavior) overload.
See Also

MockRepository Class
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;()()()()</td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(array&lt;Object&gt;[][])</td>
<td>Creates a new mock with the default <code>MockBehavior</code> specified at factory construction time and with the the given constructor arguments for the class. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(MockBehavior)</td>
<td>Creates a new mock with the given behavior. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>Create&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(MockBehavior, array&lt;Object&gt;[][])</td>
<td>Creates a new mock with the given behavior and with the the given constructor arguments for the class. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>CreateMock&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;()</td>
<td>Implements creation of a new mock within the factory. (Inherited from <code>MockFactory</code>.)</td>
</tr>
<tr>
<td>Of&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;()()()()</td>
<td>Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.</td>
</tr>
<tr>
<td>Of&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(Expression&lt;(Of &lt;&lt;'(Func&lt;((Of &lt;(O&lt;T,TResult&gt;&gt;)&gt;&gt;()[]))))</td>
<td>Creates an mock object of the indicated type.</td>
</tr>
<tr>
<td>OneOf&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;()()()()</td>
<td></td>
</tr>
<tr>
<td>OneOf&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(Expression&lt;(Of &lt;&lt;'(Func&lt;((Of &lt;(O&lt;T,TResult&gt;&gt;)&gt;&gt;()[]))))</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Verify</td>
<td>Verifies all verifiable expectations on all mocks created by this factory. (Inherited from MockFactory.)</td>
</tr>
<tr>
<td>VerifyAll</td>
<td>Verifies all verifiable expectations on all mocks created by this factory. (Inherited from MockFactory.)</td>
</tr>
<tr>
<td>VerifyMocks</td>
<td>Invokes verifyAction for each mock in Mocks, and accumulates the resulting MockVerificationException that might be thrown from the action. (Inherited from MockFactory.)</td>
</tr>
</tbody>
</table>
See Also

MockRepository Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockRepository.....Create Method

MockRepository Class See Also Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create&lt;(&lt;'T'&gt;)&gt;()()()</td>
<td>Creates a new mock with the default MockBehavior specified at factory construction time. (Inherited from MockFactory.)</td>
</tr>
<tr>
<td>Create&lt;(&lt;'T'&gt;)&gt;(array&lt;Object&gt;[])[][]]</td>
<td>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.)</td>
</tr>
<tr>
<td>Create&lt;(&lt;'T'&gt;)&gt;(MockBehavior)</td>
<td>Creates a new mock with the given behavior. (Inherited from MockFactory.)</td>
</tr>
<tr>
<td>Create&lt;(&lt;'T'&gt;)&gt;(MockBehavior, array&lt;Object&gt;[][])</td>
<td>Creates a new mock with the given behavior and with the the given constructor arguments for the class. (Inherited from MockFactory.)</td>
</tr>
</tbody>
</table>
See Also

MockRepository Class
MockRepository Members
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Of&lt;(Of &lt;&lt;'T'}}&gt;&gt;&gt;()()</code></td>
<td>Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.</td>
</tr>
</tbody>
</table>
See Also

MockRepository Class
MockRepository Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockRepository..::Of<(Of <('T)>)> Method

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)
Syntax

C#

public IQueryable<T> Of<T>()
where T : class
Type Parameters

T
The type of the mocked object to query.
See Also

MockRepository Class
Of Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockRepository.........Of<(Of <('T')>)> Method (Expression<(Of <('Func')>)>)

MockRepository Class See Also Send Feedback


Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

**C#**

```csharp
public IQueryable<T> Of<T>(
    Expression<Func> specification
)
where T : class
```

### Parameters

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

T
See Also

MockRepository Class
Of Overload
Moq Namespace

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

Include Protected Members
Include Inherited Members
Moq
MockRepository......OneOf Method
MockRepository Class See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OneOf&lt;&lt;'(T)&gt;&gt;()()()</td>
<td>Creates a mock object of the indicated type.</td>
</tr>
<tr>
<td>OneOf&lt;&lt;'(T)&gt;&gt;)(Expression&lt;&lt;'(Of &lt;&lt;'(Func&lt;&lt;'(&lt;T, TResult&gt;&gt;))&gt;&gt;)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MockRepository Class
MockRepository Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

```
public T OneOf<T>()
where T : class
```
Type Parameters

T
   The type of the mocked object.

Return Value

The mocked object created.
See Also

MockRepository Class
OneOf Overload
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
MockRepository...OneOf<(Of <('T')>)> Method (Expression<(Of <('Func')>)>)

MockRepository Class See Also Send Feedback


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

C#

public T OneOf<T>(
    Expression<Func> specification
)
where T : class

Parameters

specification
    Type: System.Linq.Expressions...Expression<(Of<(Fun<(Of<(T,TResult)>)>)>)>
Type Parameters

T
See Also

MockRepository Class
OneOf Overload
Moq Namespace

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

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

MockRepository Class
Moq Namespace

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

Moq

Mocks Class

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)
Syntax

C#

public static class Mocks
Inheritance Hierarchy

System..::..Object
Moq..::..Mocks
See Also

Mocks Members
Moq Namespace

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

Include Protected Members
Include Inherited Members

Moq

Mocks Members

Mocks Class Methods See Also Send Feedback
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Of&lt;(Of &lt;*&gt;('T)&gt;&gt;)QQQQ</code></td>
<td>Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.</td>
</tr>
<tr>
<td><code>Of&lt;(Of &lt;*&gt;('T)&gt;&gt;)</code> (Expression&lt;(Of `&lt;'(Func&lt;(Of &lt;*&gt;('T, TResult&gt;)&gt;)&gt;)&gt;&gt;))</td>
<td>Obsolete. Creates an mock object of the indicated type.</td>
</tr>
<tr>
<td><code>OneOf&lt;(Of &lt;*&gt;('T)&gt;&gt;)QQQ</code></td>
<td>Obsolete.</td>
</tr>
</tbody>
</table>
See Also

Mocks Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Include Protected Members
Include Inherited Members
Moq
Mocks Methods
Mocks Class See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Of&lt;(Of &lt;&lt;'(T)&gt;&gt;,&gt;)(),&gt;()()</code></td>
<td>Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.</td>
</tr>
<tr>
<td><code>Of&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(Expression&lt;(Of &lt;&lt;'(Func&lt;(Of &lt;&lt;'(T, TResult&gt;&gt;)&gt;&gt;)&gt;&gt;)</code>)`</td>
<td><strong>Obsolete.</strong> Creates an mock object of the indicated type.</td>
</tr>
<tr>
<td><code>OneOf&lt;(Of &lt;&lt;'(T)&gt;&gt;,&gt;)(),&gt;()()</code></td>
<td><strong>Obsolete.</strong></td>
</tr>
</tbody>
</table>
See Also

Mocks Class
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Of&lt;(Of &lt;&lt;'T'&gt;)&gt;&gt;)(0000)</code></td>
<td>Access the universe of mocks of the given type, to retrieve those that behave according to the LINQ query specification.</td>
</tr>
<tr>
<td><code>Of&lt;(Of &lt;&lt;'T'&gt;)&gt;&gt;)(Expression&lt;(Of &lt;&lt;'(Func&lt;(Of &lt;('T, TResult&gt;)&gt;&gt;&gt;()))</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

Mocks Class
Mocks Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

public static IQueryable<T> Of<T>()
where T : class
Type Parameters

T

The type of the mocked object to query.
See Also

Mocks Class
Of Overload
Moq Namespace

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

Moq

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

Mocks Class  See Also  Send Feedback

[Missing <summary> documentation for
]

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
**Syntax**

C#

```csharp
public static IQueryable<T> Of<T>(
    Expression<Func> specification
)
where T : class
```

**Parameters**

- specification
  - Type: `System.Linq.Expressions.Expression<Func<T, TResult>>`
Type Parameters

T
See Also

Mocks Class
Of Overload
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OneOf&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;()()()</td>
<td>Obsolete. Creates an mock object of the indicated type.</td>
</tr>
<tr>
<td>OneOf&lt;Of &lt;&lt;'(T))&gt;&gt;&gt;(Expression&lt;Of &lt;='(Func&lt;Of &lt;='(T, TResult)&gt;&gt;)&gt;&gt;)</td>
<td>Obsolete.</td>
</tr>
</tbody>
</table>
See Also

Mocks Class
Mocks Members
Moq Namespace

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

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

Creates an mock object of the indicated type.

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

C#

```csharp
[ObsoleteAttribute("Moved to Mock.Of<T>, as it's a single one, so no true")]  
public static T OneOf<T>()  
where T : class
```
Type Parameters

T
   The type of the mocked object.

Return Value

The mocked object created.
See Also

Mocks Class
OneOf Overload
Moq Namespace

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

Moq

Mocks....OneOf<(Of <("T")>)> Method (Expression<(Of <("Func")>)>)

Mocks Class See Also Send Feedback

[Missing <summary> documentation for

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

C#

```csharp
[ObsoleteAttribute("Moved to Mock.Of<T>, as it's a single one, so no true")] public static T OneOf<T>(
    Expression<Func> specification
) where T : class

Parameters

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

T
See Also

Mocks Class
OneOf Overload
Moq Namespace

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

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)
Syntax

C#

public class MockSequence
Inheritance Hierarchy

System..Object
Moq..MockSequence
See Also

MockSequence Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockSequence` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MockSequence</td>
<td>Initialize a trace setup</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>Allow sequence to be repeated</td>
</tr>
</tbody>
</table>
See Also

MockSequence Class
Moq Namespace

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

Initialize a trace setup

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

C#

public MockSequence()
See Also

MockSequence Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The **MockSequence** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>Allow sequence to be repeated</td>
</tr>
</tbody>
</table>
See Also

MockSequence Class
Moq Namespace

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

C#

```csharp
public bool Cyclic { get; set; }
```
See Also

MockSequence Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
define nice api

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

C#

public static class MockSequenceHelper
Inheritance Hierarchy

System..::..Object
    Moq..::..MockSequenceHelper
See Also

MockSequenceHelper Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `MockSequenceHelper` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InSequence(Of &lt;&lt;(TMock)&gt;&gt;)</code></td>
<td>Perform an expectation in the trace.</td>
</tr>
</tbody>
</table>
See Also

MockSequenceHelper Class
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InSequence&lt;Of &lt;&lt;'(TMock)&gt;&gt;)</code></td>
<td>Perform an expectation in the trace.</td>
</tr>
</tbody>
</table>
See Also

MockSequenceHelper Class
Moq Namespace

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

Moq

MockSequenceHelper...InSequence<(Of <("TMock">)> Method

MockSequenceHelper Class  See Also  Send Feedback

Perform an expectation in the trace.

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

C#

public static ISetupConditionResult<TMock> InSequence<TMock>(
    this Mock<TMock> mock,
    MockSequence sequence
)
where TMock : class

Parameters

mock
    Type: Moq::Mock(Of (<'TMock'>))>

sequence
    Type: Moq::MockSequence
Type Parameters

TMock
See Also

MockSequenceHelper Class
Moq Namespace

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

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

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

C#

public enum Range
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
<th>Inclusive</th>
<th>Exclusive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
See Also

Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

public static class SequenceExtensions
Inheritance Hierarchy

System:::Object
Moq:::SequenceExtensions
See Also

SequenceExtensions Members
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `SequenceExtensions` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetupSequence&lt;(Of &lt;&lt;(TMock, TResult)&gt;&gt;)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SequenceExtensions Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `SequenceExtensions` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetupSequence&lt;(Of &lt;&lt;(TMock, TResult)&gt;&gt;)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SequenceExtensions Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
SequenceExtensions..:..SetupSequence<(Of <('TMock, TResult)>)> Method
SequenceExtensions Class  See Also  Send Feedback

[Missing <summary> documentation for

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

C#

```csharp
public static ISetupSequentialResult<TResult> SetupSequence<TMock, TRes
    this Mock<TMock> mock,
    Expression<Func> expression
}
where TMock : class
```

Parameters

mock
   Type: `Mock<Of ($(TMock)>)>`  

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

TMock
TResult
See Also

SequenceExtensions Class
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

public struct Times
See Also

Times Members
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtLeast</td>
<td>Specifies that a mocked method should be invoked callCount times as minimum.</td>
</tr>
<tr>
<td>AtLeastOnce</td>
<td>Specifies that a mocked method should be invoked one time as minimum.</td>
</tr>
<tr>
<td>AtMost</td>
<td>Specifies that a mocked method should be invoked callCount time as maximum.</td>
</tr>
<tr>
<td>AtMostOnce</td>
<td>Specifies that a mocked method should be invoked one time as maximum.</td>
</tr>
<tr>
<td>Between</td>
<td>Specifies that a mocked method should be invoked between callCountFrom and callCountTo times.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance.</td>
</tr>
<tr>
<td></td>
<td>(Overrides ValueType..Equals(Object).)</td>
</tr>
<tr>
<td>Exactly</td>
<td>Specifies that a mocked method should be invoked exactly callCount times.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns a hash code for this instance.</td>
</tr>
<tr>
<td></td>
<td>(Overrides ValueType..GetHashCode().)</td>
</tr>
<tr>
<td>Never</td>
<td>Specifies that a mocked method should not be invoked.</td>
</tr>
<tr>
<td>Once</td>
<td>Specifies that a mocked method should be invoked exactly one time.</td>
</tr>
</tbody>
</table>
## Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality</td>
<td>Determines whether two specified Times objects have the same value.</td>
</tr>
<tr>
<td>Inequality</td>
<td>Determines whether two specified Times objects have different values.</td>
</tr>
</tbody>
</table>
See Also

Times Structure
Moq Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtLeast</td>
<td>Specifies that a mocked method should be invoked callCount times as minimum.</td>
</tr>
<tr>
<td>AtLeastOnce</td>
<td>Specifies that a mocked method should be invoked one time as minimum.</td>
</tr>
<tr>
<td>AtMost</td>
<td>Specifies that a mocked method should be invoked callCount time as maximum.</td>
</tr>
<tr>
<td>AtMostOnce</td>
<td>Specifies that a mocked method should be invoked one time as maximum.</td>
</tr>
<tr>
<td>Between</td>
<td>Specifies that a mocked method should be invoked between callCountFrom and callCountTo times.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to this instance. (Overides ValueType..::.Equals(Object).)</td>
</tr>
<tr>
<td>Exactly</td>
<td>Specifies that a mocked method should be invoked exactly callCount times.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns a hash code for this instance. (Overides ValueType..::.GetHashCode())</td>
</tr>
<tr>
<td>Never</td>
<td>Specifies that a mocked method should not be invoked.</td>
</tr>
<tr>
<td>Once</td>
<td>Specifies that a mocked method should be invoked exactly one time.</td>
</tr>
</tbody>
</table>
See Also

Times Structure  
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
Times..::.AtLeast Method

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)
## Syntax

C#

```csharp
public static Times AtLeast(
    int callCount
)
```

### Parameters

callCount
Type: `System..::.Int32`  
The minimum number of times.

### Return Value

An object defining the allowed number of invocations.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
AtLeastOnce Method

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#**

```csharp
public static Times AtLeastOnce()
```

### Return Value

An object defining the allowed number of invocations.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies that a mocked method should be invoked callCount time as maximum.

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

C#

public static Times AtMost(
    int callCount
)

Parameters

callCount
    Type: System.Int32
    The maximum number of times.

Return Value

An object defining the allowed number of invocations.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies that a mocked method should be invoked one time as maximum.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
## Syntax

**C#**

```csharp
public static Times AtMostOnce()
```

## Return Value

An object defining the allowed number of invocations.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
Times..::..Between Method

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#**

```csharp
public static Times Between(
    int callCountFrom,
    int callCountTo,
    Range rangeKind
)
```

#### Parameters

callCountFrom
Type: `System:::Int32`
The minimum number of times.

callCountTo
Type: `System:::Int32`
The maximum number of times.

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

#### Return Value

An object defining the allowed number of invocations.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Determines whether the specified Object is equal to this instance.

Namespace: Moq
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

#### C#

```csharp
public override bool Equals(
    Object obj
)
```

#### Parameters

**obj**
- Type: System...Object
- The **Object** to compare with this instance.

#### Return Value

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

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies that a mocked method should be invoked exactly callCount times.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public static Times Exactly(
    int callCount
)

Parameters

callCount
  Type: System::..Int32
  The times that a method or property can be called.

Return Value

An object defining the allowed number of invocations.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Returns a hash code for this instance.

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

**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
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.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies that a mocked method should not be invoked.

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

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

C#

```csharp
public static Times Never()
```

Return Value

An object defining the allowed number of invocations.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies that a mocked method should be invoked exactly one time.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public static Times Once()

Return Value

An object defining the allowed number of invocations.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `Times` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equality</td>
<td>Determines whether two specified <code>Times</code> objects have the same value.</td>
</tr>
<tr>
<td>Inequality</td>
<td>Determines whether two specified <code>Times</code> objects have different values.</td>
</tr>
</tbody>
</table>
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Determines whether two specified Times objects have the same value.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public static bool operator ==(
    Times left,
    Times 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 the same as the value of right; otherwise, false.
See Also

Times Structure
Moq Namespace

Send comments on this topic to moqdisc@googlegroups.com
Determines whether two specified `Times` objects have different values.

**Namespace:** Moq  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

```csharp
public static bool operator !=(
    Times left,
    Times 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.
See Also

Times Structure
Moq Namespace

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

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICallback</td>
<td>Defines the Callback verb and overloads.</td>
</tr>
<tr>
<td>ICallback(Of &lt;'TMock, TResult&gt;)&gt;</td>
<td>Defines the Callback verb and overloads for callbacks on setups that return a value.</td>
</tr>
<tr>
<td>ICallbackGetter(Of &lt;'TMock, TProperty&gt;)&gt;</td>
<td>Defines the Callback verb for property getter setups.</td>
</tr>
<tr>
<td>ICallbackSetter(Of &lt;'TMock, TProperty&gt;)&gt;</td>
<td>Defines the Callback verb for property setter setups.</td>
</tr>
<tr>
<td>IRaise(Of &lt;'T&gt;)&gt;</td>
<td>Defines the Raises verb.</td>
</tr>
<tr>
<td>IReturns(Of &lt;'TMock, TResult&gt;)&gt;</td>
<td>Defines the Returns verb.</td>
</tr>
<tr>
<td>IReturnsGetter(Of &lt;'TMock, TProperty&gt;)&gt;</td>
<td>Defines the Returns verb for property getter setups.</td>
</tr>
<tr>
<td>ISetupConditionResult(Of &lt;'T&gt;)&gt;</td>
<td>Implements the fluent API.</td>
</tr>
<tr>
<td>ISetupSequentialResult(Of &lt;'TResult&gt;)&gt;</td>
<td>Language for ReturnSequence</td>
</tr>
<tr>
<td>IThrows</td>
<td>Defines the Throws verb.</td>
</tr>
<tr>
<td>IVerifies</td>
<td>Defines the Verifiable verb.</td>
</tr>
</tbody>
</table>

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

Defines the Callback verb and overloads.

**Namespace:** Moq.Language
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public interface ICallback : IHideObjectMembers
See Also

ICallback Members
Moq.Language Namespace

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

Include Protected Members
Include Inherited Members
Moq
ICallback Members

ICallback Interface Methods See Also Send Feedback
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback(Action)</td>
<td>Specifies a callback to invoke when the method is called.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(Action&lt;(Of &lt;&lt;'(T)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;)&gt;(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;&gt;)</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;)&gt;(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;&gt;)</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;)&gt;(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;&gt;)</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;)&gt;(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;&gt;)</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;)&gt;(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;&gt;)</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
</tbody>
</table>
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>>)>)

Callback<(Of <<'T1, T2, T3>>)> (Action<(Of <<'T1, T2, T3>>)>)

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>>)>)

Callback<(Of <<'T1, T2, T3, T4, T5, T6, 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, T8>>)>)

Arguments:
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>>)> (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.
See Also

ICallback Interface
Moq.Language Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback(Action)</td>
<td>Specifies a callback to invoke when the method is called.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T))&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
</tbody>
</table>
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>)>>))

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

Callback<(Of <<'(T1, T2)>>)>

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

Callback<(Of <<'(T1, T2, T3, T4)>>)>

Callback<(Of <<'(T1, T2, T3, T4, T5)>>)>

Callback<(Of <<'(T1, T2, T3, T4, T5, T6)>>)>

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7)>>)>

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8)>>)>

Callback<(Of <<'(T1, T2, T3, T4, T5, T6, T7, T8, T9)>>)>

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

ICallback Interface
Moq.Language Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback(Action)</td>
<td>Specifies a callback to invoke when the method is called.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T&gt;)&gt;&gt;())(Action&lt;(Of &lt;&lt;'(T&gt;)&gt;&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10&gt;)&gt;&gt;())(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10&gt;)&gt;&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11&gt;)&gt;&gt;())(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11&gt;)&gt;&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12&gt;)&gt;&gt;())(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12&gt;)&gt;&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13&gt;)&gt;&gt;())(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13&gt;)&gt;&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14&gt;)&gt;&gt;())(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14&gt;)&gt;&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
</tbody>
</table>
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)>))

Callback<(Of <'(T1, T2, T3)>))
(Action<(Of <'(T1, T2, T3)>))

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)>))

Callback<(Of <'(T1, T2, T3, T4, T5, T6, 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, T8)>))

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

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

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

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

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

ICallback Interface
ICallback Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
ICallback...Callback Method (Action)
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)
Syntax

C#

ICallbackResult Callback(
    Action 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#  

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

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
ICallback INTERFACE Callback(Of <('T')>) Method (Action(Of <('T')>))

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#

`ICallbackResult Callback<T>(
    Action<T> action
)`

**Parameters**

`action`
Type: `System::Action(Of (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#
mock.Setup(x => x.Execute(It.IsAny<string>()))
    .Callback((string command) => Console.WriteLine(command));
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

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

ICallbackResult Callback<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>(
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10> action
)

Parameters

action
    Type: System.....Action<(Of <('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)>)>  
    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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth argument of the invoked method.
T10 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#
mock.Setup(x => x.Execute(
    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
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback...::Callback<Of <('T1, T2, T3, T4, 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#**

```csharp
ICallbackResult Callback<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>(
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11> action
)
```

### Parameters

**action**

Type: `System::Action<Of (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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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

Invokes the given callback with the concrete invocation arguments values.

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

```csharp
mock.Setup(x => x.Execute(
             It.IsAny<string>(),
             It.IsAny<string>(),
             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
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

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

ICallbackResult Callback<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>(
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12> action)

Parameters

action
    Type: System:::.Action<(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
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 ninth 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 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#

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    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, string arg5))
```
See Also

ICallback Interface
Callback Overload
Mock.Language Namespace

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

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

Parameters

action
  Type: System...Action<Of<('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)>>
  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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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.

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:

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    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
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

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

Moq

ICallback...Callback<Of <('T1, T2, 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#**

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

**Parameters**

action

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

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.

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.

### 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#

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    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) => { /* callback implementation */ });
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

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

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

Parameters

action

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

    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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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.

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#
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    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
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback 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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

ICallbackResult Callback<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16> (Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>)

Parameters

action

Type: System..::.Action<((T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16)>>

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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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:

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    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
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback. Callback<Of <{"T1, T2"}>>) Method (Action)

ICallback Interface See Also Send Feedback


Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

ICallbackResult Callback<T1, T2>(
    Action action
)

Parameters

action
    Type: System..::.Action<(Of <(T1, T2)>)>
Type Parameters

T1
T2
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

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

ICallback Interface  See Also  Send Feedback

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

**Namespace:** Moq.Language
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

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

Parameters

action
    Type: System.....Action(Of <(T1, T2, T3)>)}
Type Parameters

T1
T2
T3
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback.....Callback<(Of <("T1, T2, T3, T4">)> Method (Action)
ICallback Interface See Also Send Feedback

[Missing <summary> documentation for

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

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

Parameters

action
    Type: System...Action(Of <(T1, T2, T3, T4)>)
<table>
<thead>
<tr>
<th>Type Parameters</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback......Callback<(Of <("T1, T2, T3, T4, T5")>)> Method (Action)
ICallback Interface See Also Send Feedback


Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

ICallbackResult Callback<T1, T2, T3, T4, T5>(
    Action action
)

Parameters

action
    Type: System:::Action(Of <(T1, T2, T3, T4, T5)>)
<table>
<thead>
<tr>
<th>Type Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
</tr>
<tr>
<td>T2</td>
</tr>
<tr>
<td>T3</td>
</tr>
<tr>
<td>T4</td>
</tr>
<tr>
<td>T5</td>
</tr>
</tbody>
</table>
See Also

ICallback Interface  
Callback Overload  
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback......Callback<(Of <('T1, T2, T3, T4, 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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
**Syntax**

**C#**

ICallbackResult Callback<T1, T2, T3, T4, T5, T6>(

    Action action

)

**Parameters**

action

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

T1
T2
T3
T4
T5
T6
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback::..::Callback<Of <('T1, T2, T3, T4, 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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

ICallbackResult Callback<T1, T2, T3, T4, T5, T6, T7>(
    Action action
)

Parameters

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


### Type Parameters

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td></td>
</tr>
<tr>
<td>T5</td>
<td></td>
</tr>
<tr>
<td>T6</td>
<td></td>
</tr>
<tr>
<td>T7</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

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

Moq

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

ICallback Interface See Also Send Feedback


Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

**C#**

```csharp
ICallbackResult Callback<T1, T2, T3, T4, T5, T6, T7, T8>(
    Action action
)
```

### Parameters

- **action**
  - Type: `System...Action<Of <(T1, T2, T3, T4, T5, T6, T7, T8)>>>`
Type Parameters

T1
T2
T3
T4
T5
T6
T7
T8
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

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

Moq
ICallback<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#

ICallbackResult Callback<T1, T2, T3, T4, T5, T6, T7, T8, T9>(
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9> action
)

Parameters

action
    Type: System:::Action<(Of <(T1, T2, T3, T4, T5, T6, T7, T8, T9)>)> 
    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.
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#

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    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
```
See Also

ICallback Interface
Callback Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
ICallback<((TMock, TResult)>) Interface

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#

```csharp
public interface ICallback<TMock, TResult> : IHideObjectMembers
where TMock : class
```
Type Parameters

TMock
    Mocked type.
TResult
    Type of the return value of the setup.
See Also

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback(Action)</td>
<td>Specifies a callback to invoke when the method is called.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(Action&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;)</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;&gt;&gt;(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;&gt;&gt;(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;&gt;&gt;(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;&gt;&gt;(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
</tbody>
</table>
Arguments:

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

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

Send comments on this topic to moqdisc@googlegroups.com
ICallback(Of <'TMock, TResult> )> Methods
ICallback(Of <'TMock, TResult> )> Interface See Also Send Feedback
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback(Action)</td>
<td>Specifies a callback to invoke when the method is called.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;() (Action&lt;(Of &lt;&lt;'(T)&gt;&gt;())</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;&gt;&gt;() (Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;&gt;&gt;() (Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;&gt;&gt;() (Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;&gt;&gt;() (Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;&gt;&gt;() (Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;()))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
</tbody>
</table>
Specifies a callback to invoke when the method is called that receives the original arguments.

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

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

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

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.
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback(Action)</td>
<td>Specifies a callback to invoke when the method is called.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
<tr>
<td>Callback&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;&gt;&gt;)(Action&lt;(Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;&gt;&gt;))</td>
<td>Specifies a callback to invoke when the method is called that receives the original arguments.</td>
</tr>
</tbody>
</table>
Arguments.

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

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

Send comments on this topic to moqdisc@googlegroups.com
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](https://example.com)

**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
## Syntax

### C#

```csharp
IReturnsThrows<TMock, TResult> Callback(
    Action 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#

```csharp
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.
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
ICallback<(Of <('TMock, TResult)>)> Interface

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#

IReturnsThrows<TMock, TResult> Callback<T>(
    Action<T> action
)

Parameters

action
    Type: System::..::Action<Of (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:

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

ICallback<^TMock, TResult^> Interface
Callback Overload
Moq.Language Namespace

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

```csharp
IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6, T7, 
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10> action
```

**Parameters**

- **action**
  - Type: `System:::Action<(Of <('T1, T2, T3, T4, T5, T6, T7, T8, T9, 
  T10)>)>`
  - 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.
T8
The type of the eighth argument of the invoked method.
T9
The type of the ninth 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#

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    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((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
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)>)>)

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#

IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6, T7, 
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11> action
)

Parameters

action
    Type: System:::Action<(Of <( 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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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 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#

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

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

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

IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6, T7, 
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12> action
}

Parameters

action
    Type: System..::.Action<(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  
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 ninth 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#  

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    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((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
ICallback<TMock, TResult> Method (Action<TMock, TResult>)

ICallback<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#

IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6, T7, 
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>>

Parameters

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

T8
The type of the eighth argument of the invoked method.

T9
The type of the ninth 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.

### 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:

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    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((arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9, arg10))
```
See Also

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

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

IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6, T7, Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T14>>

Parameters

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

T8
The type of the eighth argument of the invoked method.

T9
The type of the ninth 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.

**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#

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    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((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
ICallback(Of (Of (<'TMock, TResult>))...) Callback(Of (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 (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#**

```csharp
IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6, T7,
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>>
```

#### Parameters

**action**

Type: `System::Action<(<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>)>`

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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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.

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:

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    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((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
ICallback<Of <(<'TMock, TResult>>)>::Callback<Of <(<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>>)>

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#

IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>(
    Action<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>
)

Parameters

action

    Type: System:::Action<(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
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 ninth 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 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#

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    It.IsAny<string>(),
    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((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
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsThrows<TMock, TResult> Callback<T1, T2>(
    Action action
)

Parameters

action
    Type: System...Action((Of (('T1, T2)>)})
Type Parameters

T1
T2
See Also

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

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

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

C#

IReturnsThrows<TMock, TResult> Callback<T1, T2, T3>(
    Action action
)

Parameters

action
    Type: System:::Action(Of <(T1, T2, T3)>)
### Type Parameters

- T1
- T2
- T3
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
ICallback(Of (Of (TMock, TResult)>)...) Method (Action)

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

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

Parameters

action
    Type: System..::.Action<Of <(Of 'T1, T2, T3, T4)>>
<table>
<thead>
<tr>
<th>Type Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
</tr>
<tr>
<td>T2</td>
</tr>
<tr>
<td>T3</td>
</tr>
<tr>
<td>T4</td>
</tr>
</tbody>
</table>
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)>) Method (Action)

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

[Missing <summary> documentation for

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5>(
    Action action
)

Parameters

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

T1
T2
T3
T4
T5
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
ICallback(Of <('TMock, TResult)>)<...:.:.Callback(Of <('T1, T2, T3, T4, T5, T6)>)> Method (Action)
ICallback(Of <('TMock, TResult)>)> Interface See Also Send Feedback


Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

**C#**

```csharp
IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6>(
    Action action
)
```

### Parameters

**action**

Type: `System:::Action<(Of <(T1, T2, T3, T4, T5, T6)>)>`
### Type Parameters

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>T2</td>
</tr>
<tr>
<td>T3</td>
<td>T4</td>
</tr>
<tr>
<td>T4</td>
<td>T5</td>
</tr>
<tr>
<td>T5</td>
<td>T6</td>
</tr>
</tbody>
</table>
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
ICallback(Of @(TMock, TResult)>).Method (Action)


Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

#### C#

```csharp
IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6, T7>(
    Action action
)
```

### Parameters

**action**

Type: `System...Action<((T1, T2, T3, T4, T5, T6, T7)>)`
<table>
<thead>
<tr>
<th>Type Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
</tr>
<tr>
<td>T2</td>
</tr>
<tr>
<td>T3</td>
</tr>
<tr>
<td>T4</td>
</tr>
<tr>
<td>T5</td>
</tr>
<tr>
<td>T6</td>
</tr>
<tr>
<td>T7</td>
</tr>
</tbody>
</table>
See Also

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

Send comments on this topic to moqdisc@googlegroups.com

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsThrows<TMock, TResult> Callback<T1, T2, T3, T4, T5, T6, T7, TMock, TResult> action

Parameters

action
    Type: System...Action(Of ($(T1, T2, T3, T4, T5, T6, T7))>
Type Parameters

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

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#

IReturnsThrows&lt;TMock, TResult&gt; Callback&lt;T1, T2, T3, T4, T5, T6, T7, 
   Action&lt;T1, T2, T3, T4, T5, T6, T7, T8, T9&gt; action
)

Parameters

action
   Type: System:::Action&lt;(Of &lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9')&gt;&gt;
   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.
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:

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<string>(),
    It.IsAny<string>(),
    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((arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9) => Console.WriteLine(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
ICallbackGetter<Of <('TMock, TProperty)>)> Interface

Defines the Callback verb for property getter setups.

**Namespace:** Moq.Language

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

C#

```csharp
public interface ICallbackGetter<TMock, TProperty> : IHideObjectMembers
where TMock : class
```
Type Parameters

TMock
   Mocked type.
TProperty
   Type of the property.
See Also

ICallbackGetter<(Of ( Of (Mock, Property)>))> Members
Mock.Language.Namespace
MockSetupGet`1(Expression<(Of <'(Func<(Of <'(UTP, UMP)>))>)>)>

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

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

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

Send comments on this topic to moqdisc@googlegroups.com
ICallbackGetter(Of (TMock, TProperty)>>) Methods

ICallbackGetter(Of (TMock, TProperty)>>) Interface See Also Send Feedback

The ICallbackGetter(Of (TMock, TProperty)>>) type exposes the following members.
## Methods

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

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

Send comments on this topic to moqdisc@googlegroups.com
ICallbackGetter<
(OF
(OF
(TMock, TProperty>))>
dbname: Callback Method
ICallbackGetter<
(OF
(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)
Syntax

C#

IReturnsThrowsGetter<TMock, TProperty> Callback(
    Action action
)

Parameters

action
    Type: System...Action
    Callback method to invoke.
Examples

Invokes the given callback with the property value being set.

C#

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

ICallback.Getter(Of (TMock, TProperty))> Interface
Moq.Language Namespace

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

public interface ICallbackSetter<TProperty> : IHideObjectMembers
Type Parameters

TProperty
  Type of the property.
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
The `ICallbackSetter<Of <('TProperty')>>()` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback</td>
<td>Specifies a callback to invoke when the property is set that receives the property value being set.</td>
</tr>
</tbody>
</table>
See Also

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

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

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

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

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

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

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
## Syntax

### C#

```csharp
ICallbackResult Callback(
    Action<TProperty> action
)
```

### Parameters

- **action**
  - Type: `System::Action<TProperty>`
  - 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));
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

public interface IRaise<T> : IHideObjectMembers
Type Parameters

T
See Also

IRaise<(Of <(T)>)> Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members
Include Inherited Members
Moq
IRaise(Of ("T")) Members
IRaise(Of ("T")) Interface Methods See Also Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Raises(Action&lt;T&gt;, EventArgs)</code></td>
<td>Specifies the event that will be raised when the setup is met.</td>
</tr>
<tr>
<td><code>Raises(Action&lt;T&gt;, Func&lt;TResult&gt;)</code></td>
<td>Specifies the custom event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td><code>Raises(Action&lt;T&gt;, Object[])</code></td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td><code>Raises&lt;(Action&lt;T1&gt;, Func&lt;T1,TResult&gt;)&gt;</code></td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td><code>Raises&lt;(Action&lt;T1,T2,T3,T4,T5,T6,T7,T8,T9,T10&gt;, EventArgs&gt;)&gt;</code></td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td><code>Raises&lt;(Action&lt;T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11&gt;, EventArgs&gt;)&gt;</code></td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td><code>Raises&lt;(Action&lt;T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12&gt;, EventArgs&gt;)&gt;</code></td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td><code>Raises&lt;(Action&lt;T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13&gt;, EventArgs&gt;)&gt;</code></td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td><code>Raises&lt;(Action&lt;T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14&gt;, EventArgs&gt;)&gt;</code></td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
</tbody>
</table>
event that will be raised when the setup is matched.

Specifies the event that will be raised when the setup is matched.

Specifies the event that will be raised when the setup is matched.
See Also

IRaise<(Of <('T')>)> Interface
Moq.Language Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raises(Action(Of &lt;&lt;'T'&gt;&gt;&gt;), EventArgs)</td>
<td>Specifies the event that will be raised when the setup is met.</td>
</tr>
<tr>
<td>Raises(Action(Of &lt;&lt;'T'&gt;&gt;&gt;, Func(Of &lt;&lt;'TResult'&gt;&gt;))</td>
<td>Specifies the custom event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises(Action(Of &lt;&lt;'T'&gt;&gt;&gt;, array&lt;Object&gt;[])[])</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises(Of &lt;&lt;'T1'&gt;&gt;&gt;(Action(Of &lt;&lt;'T'&gt;&gt;&gt;, Func(Of &lt;&lt;'TResult'&gt;&gt;)))</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises(Of &lt;&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10'&gt;&gt;&gt;(Action(Of &lt;&lt;'T'&gt;&gt;&gt;, Func(Of &lt;&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, EventArgs&gt;&gt;)</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises(Of &lt;&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11'&gt;&gt;&gt;(Action(Of &lt;&lt;'T'&gt;&gt;&gt;, Func(Of &lt;&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, EventArgs&gt;&gt;)</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises(Of &lt;&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12'&gt;&gt;&gt;(Action(Of &lt;&lt;'T'&gt;&gt;&gt;, Func(Of &lt;&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, EventArgs&gt;&gt;)</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises(Of &lt;&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13'&gt;&gt;&gt;(Action(Of &lt;&lt;'T'&gt;&gt;&gt;, Func(Of &lt;&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, EventArgs&gt;&gt;)</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises(Of &lt;&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14'&gt;&gt;&gt;(Action(Of</td>
<td>Specifies the</td>
</tr>
</tbody>
</table>
event that will be raised when the setup is matched.

Specifies the event that will be raised when the setup is matched.

Specifies the event that will be raised when the setup is matched.
See Also

IRaise(Of <(T)>)) Interface
Moq.Language Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raises(Action&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;, EventArgs)</td>
<td>Specifies the event that will be raised when the setup is met.</td>
</tr>
<tr>
<td>Raises(Action&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;, Func(Of &lt;&lt;'TResult'&gt;&gt;))</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9)&gt;&gt;&gt;(Action&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;, Func&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, EventArgs)&gt;&gt;))</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10)&gt;&gt;&gt;(Action&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;, Func&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, EventArgs)&gt;&gt;))</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11)&gt;&gt;&gt;(Action&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;, Func&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, EventArgs)&gt;&gt;))</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12)&gt;&gt;&gt;(Action&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;, Func&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, EventArgs)&gt;&gt;))</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13)&gt;&gt;&gt;(Action&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;, Func&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, EventArgs)&gt;&gt;))</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
<tr>
<td>Raises&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14)&gt;&gt;&gt;(Action&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;, Func&lt;Of &lt;&lt;'(T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, EventArgs)&gt;&gt;))</td>
<td>Specifies the event that will be raised when the setup is matched.</td>
</tr>
</tbody>
</table>
Specifies the event that will be raised when the setup is matched.
See Also

IRaise<(Of <('T')>)> Interface
IRaise<(Of <('T')>)> Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

```csharp
IVerifies Raises(
    Action<T> eventExpression,
    EventArgs args
)
```

Parameters

eventExpression
  Type: System.....Action<(Of <(>'T')>))
  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# 

```csharp
var mock = new Mock<IContainer>();
mock.Setup(add => add.Add(It.IsAny<string>(), It.IsAny<object>())).
    Raises(add => add.Added += null, EventArgs.Empty);
```
See Also

IRaise<(Of <'T'>)> Interface
Raises Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
IRaise<(Of <(<'T'>)>)>... Raises Method (Action<(Of <(<'T'>)>), Func)
IRaise<(Of <(<'T'>)>)> Interface See Also Send Feedback

[Missing <summary> documentation for
"M:Moq.Language.IRaise`1.Raises(System.Action`0,System.Func`1)"
]

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

`<Event> Raises(
    <Action<T> eventExpression,
    Func func
)

Parameters

**eventExpression**
Type: `System....Action<Of <'T'>>)`

**func**
Type: `System....Func<Of <'TResult'>>)`
See Also

IRaise<(Of <('T')>)> Interface
Raises Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

```csharp
IVerifies Raises<T1, T2, T3, T4, T5, T6, T7, T8, T9>(
    Action<T> eventExpression,
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, EventArgs> func
)
```

Parameters

**eventExpression**
Type: `System...Action<(<'T'>)>`
The expression that represents an event attach or detach action.

**func**
Type: `System...Func<(<'T1', T2, T3, T4, T5, T6, T7, T8, T9, EventArgs>)>`
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.

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
The type of the nineth argument received by the expected invocation.
See Also

IRaise<(Of <('T')>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <('T')>)>..., Raises(Action<(Of <('T')>)>, EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
IRaise<Of <((T)>)>...:...Raise<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)
**Syntax**

**C#**

```csharp
IVerifies Raises<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10>(
    Action<T> eventExpression,
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, EventArgs> func
)
```

**Parameters**

**eventExpression**

Type: `System..::.Action(Of ((<T>))>`

The expression that represents an event attach or detach action.

**func**

Type: `System..::.Func(Of ((<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, EventArgs>))>`

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.

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
The type of the ninth argument received by the expected invocation.

T10
The type of the tenth argument received by the expected invocation.
See Also

IRaise(Of T>دوار Interface
Raises Overload
Moq.Language Namespace
IRaise(Of T>دوار Raises(Action(Of T>, EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

`IVerifies Raises<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11>(
    Action<T> eventExpression,
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, EventArgs>
)
```

Parameters

eventExpression
Type: `System::Action<Of <('T')>>`
The expression that represents an event attach or detach action.

func
Type: `System::Func<Of <<'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, EventArgs>>`
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.
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  The type of the ninth argument received by the expected invocation.
T10 The type of the tenth argument received by the expected invocation.
T11 The type of the eleventh argument received by the expected invocation.
See Also

**IRaise<(<T>)> Interface**
**Raises Overload**
**Moq.Language Namespace**
**IRaise<(<T)>>::::Raises(Action<(<T)>, EventArgs)**

Send comments on this topic to **moqdisc@googlegroups.com**
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)
### Syntax

**C#**

```csharp
IVerifies Raises<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>(
    Action<T> eventExpression,
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, EventArgs>
)
```

**Parameters**

**eventExpression**

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

The expression that represents an event attach or detach action.

**func**

Type: `System.....Func<(Of <('T1', 'T2', 'T3', 'T4', 'T5', 'T6', 'T7', 'T8', 'T9', 'T10, 'T11, 'T12, EventArgs>)>)`

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.
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  The type of the ninth argument received by the expected invocation.
T10 The type of the tenth argument received by the expected invocation.
T11 The type of the eleventh argument received by the expected invocation.
T12 The type of the twelfth argument received by the expected invocation.
See Also

IRaise<(<Of <(<'T'>)>>) Interface
Raises Overload
Moq.Language Namespace
IRaise<(<Of <(<'T'>)>>) ;::; Raises<Action<(<Of <('T'>)>>), EventArgs>

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

\[
\text{IVerifies}
\text{ Raises}\langle\text{T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13}\rangle
\text{ Action}\langle\text{T}\rangle\ \text{eventExpression,}
\text{ Func}\langle\text{T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13}\rangle
\]

Parameters

eventExpression

Type: \text{System:::Action}\langle\text{Of}\langle\langle\text{T}\rangle\text{\rangle}\rangle

The expression that represents an event attach or detach action.

func

Type: \text{System:::Func}\langle\text{Of}\langle\langle\text{T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, EventArgs}\rangle\rangle\rangle

The function that will build the \text{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.
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  The type of the ninth argument received by the expected invocation.
T10 The type of the tenth argument received by the expected invocation.
T11 The type of the eleventh argument received by the expected invocation.
T12 The type of the twelfth argument received by the expected invocation.
T13 The type of the thirteenth argument received by the expected invocation.
See Also

IRaise<(Of <*>)> Interface
Raises Overload
Mock.Language Namespace
IRaise<(Of <*>)>..::. Raises(Action<(Of <'(T)'>)..., EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
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)
Syntax

C#

```csharp
IVerifies Raises<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14>(
    Action<T> eventExpression,
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, EventArgs>
}
```

Parameters

**eventExpression**
Type: `System::Action(Of (<'T'>))`  
The expression that represents an event attach or detach action.

**func**
Type: `System::Func(Of (<'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, EventArgs>)`  
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.

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
The type of the ninth argument received by the expected invocation.

T10
The type of the tenth argument received by the expected invocation.

T11
The type of the eleventh argument received by the expected invocation.

T12
The type of the twelfth argument received by the expected invocation.

T13
The type of the thirteenth argument received by the expected invocation.

T14
The type of the fourteenth argument received by the expected invocation.
See Also

IRaise<Of (<'T'>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<Of (<'T'>)>... Raises<Action<Of <'T'>>>, EventArgs

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

**C#**

```csharp
data private _feature: IVerifies

  [C#]
  <T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15>
  
  Action<T> eventExpression,
  Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13,
  
  )

### Parameters

**eventExpression**

Type: `System::Action<Of <'T>>`  
The expression that represents an event attach or detach action.

**func**

Type: `System::Func<Of <'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, EventArgs>>`  
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.
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
  The type of the ninth argument received by the expected invocation.
T10
  The type of the tenth argument received by the expected invocation.
T11
  The type of the eleventh argument received by the expected invocation.
T12
  The type of the twelfth argument received by the expected invocation.
T13
  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.
See Also

IRaise<(Of <('T')>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <('T')>)>,...; Raises(Action<(Of <<('T')>>)>, EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

`IVerifies` Raises<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16>(
    `Action<T>` eventExpression,
    `Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, EventArgs>` func
)

Parameters

eventExpression
Type: `System.....Action<T>`
The expression that represents an event attach or detach action.

func
Type: `System.....Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, EventArgs>`
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.
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  The type of the ninth argument received by the expected invocation.
T10 The type of the tenth argument received by the expected invocation.
T11 The type of the eleventh argument received by the expected invocation.
T12 The type of the twelfth argument received by the expected invocation.
T13 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.
See Also

IRaise<(Of <('T')?>)> Interface
Raises Overload
Moq.Language Namespace
IRaise<(Of <('T')?>)>...:..: Raises(Action<(Of <'('T'>)>>, EventArgs)

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
IRaise(Of (Of('T'))...). Raises(Of (Of('T1'))) Method (Action(Of (Of('T'))), Func)

IRaise(Of (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
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

```csharp
IVerifies Raises<T1>(
    Action<T> eventExpression,
    Func func
)
```

Parameters

eventExpression
  Type: `System...Action(Of <*>>)`

func
  Type: `System...Func(Of <*>>)`
Type Parameters

T1
See Also

IRaise<(<T>)> Interface
Raises Overload
Moq.Language Namespace

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

Moq

IRaise(Of <('T1', 'T2')>)..:..Rises(Of <('T1', 'T2')>) Method (Action(Of <('T1', 'T2')>, Func)

IRaise(Of <('T1', 'T2')>) 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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

`IVerifies` `Raises<T1, T2>(
  `Action<T>` `eventExpression`,
  `Func` `func`
)`

Parameters

eventExpression
  Type: `System...Action<Of (<'T'>)>`

func
  Type: `System...Func<Of (<'T1, T2, TResult'>)>`
Type Parameters

T1
T2
See Also

IRaise<(Of 'T'>)> Interface
Raises Overload
Moq.Language Namespace

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

Moq

IRaise(Of (('T>))>

Method (Action(Of (('T>)), Func)

IRaise(Of (('T>))> Interface

See Also Send Feedback


Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

```csharp
IVerifies Raises<T1, T2, T3>(
    Action<T> eventExpression,
    Func func
)
```

Parameters

**eventExpression**
Type: `System...Action<Of (<'T'>)>`

**func**
Type: `System...Func<Of (<'T1', T2, T3, TResult'>)>`
Type Parameters

T1
T2
T3
See Also

IRaise<Of (Of 'T'>)> Interface
Raises Overload
Moq.Language Namespace

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

Moq

IRaise(Of ( Of ( Of ( Of T ) ) ) ) Method ( Action(Of ( ( Of T ) ) ), Func)

IRaise(Of ( Of T ) ) Interface See Also Send Feedback

[Missing <summary> documentation for

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

```csharp
IVerifies Raises<T1, T2, T3, T4>(
    Action<T> eventExpression,
    Func func
)
```

Parameters

eventExpression
Type: `System...Action<T>

func
Type: `System...Func<T1, T2, T3, T4, TResult>`
Type Parameters

T1
T2
T3
T4
See Also

IRaise<(Of <('T')>)> Interface
Raises Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

**IVerifies** Raises<T1, T2, T3, T4, T5>(
  **Action**<T> eventExpression,
  **Func** func
)

**Parameters**

*eventExpression*
  Type: **System..::.Action**<Of (<'T'>)>)

*func*
  Type: **System..::.Func**<Of (<'T1, T2, T3, T4, T5, TResult'>)>>
Type Parameters

T1
T2
T3
T4
T5
See Also

IRaise<(Of <(T)>)> Interface
Raises Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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


Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
**Syntax**

**C#**

```csharp
IFrames<T1, T2, T3, T4, T5, T6>(
    Action<T> eventExpression,
    Func func
)
```

**Parameters**

eventExpression
   Type: `System...Action<Of <*>>()`

func
   Type: `System...Func<Of <*>>()`
Type Parameters

T1
T2
T3
T4
T5
T6
See Also

IRaise<(Of <(T)>)> Interface
Raises Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

```csharp
IVerifies Raises<T1, T2, T3, T4, T5, T6, T7>(
    Action<T> eventExpression,
    Func func
)
```

**Parameters**

**eventExpression**
Type: `System...Action<Of<(<T>)>>`

**func**
Type: `System...Func<Of<(<T1, T2, T3, T4, T5, T6, T7, TResult>>)>`
<table>
<thead>
<tr>
<th>Type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td></td>
</tr>
<tr>
<td>T5</td>
<td></td>
</tr>
<tr>
<td>T6</td>
<td></td>
</tr>
<tr>
<td>T7</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IRaise(Of \(\langle\langle T\rangle\rangle\)\) Interface
Raises Overload
Moq.Language Namespace

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

Moq

IRaise(Of (T>))>...Rais((<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#

```csharp
IVerifies Raises<T1, T2, T3, T4, T5, T6, T7, T8>(
    Action<T> eventExpression,
    Func func
)
```

Parameters

**eventExpression**
Type: `System.....Action<Of <*>>()`

**func**
Type: `System.....Func<Of <*>>()`
Type Parameters

<table>
<thead>
<tr>
<th>T1</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
</tr>
<tr>
<td>T3</td>
</tr>
<tr>
<td>T4</td>
</tr>
<tr>
<td>T5</td>
</tr>
<tr>
<td>T6</td>
</tr>
<tr>
<td>T7</td>
</tr>
<tr>
<td>T8</td>
</tr>
</tbody>
</table>
See Also

IRaise<(Of <(T)>)> Interface
Raises Overload
Moq.Language Namespace

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

```csharp
IVerifies Raises(
    Action<T> eventExpression,
    params Object[] args
)
```

### Parameters

**eventExpression**
- Type: `System...Action<Of (<'T'>)>`
  - An expression that represents an event attach or detach action.

**args**
- Type: `array<System...Object>[][][]`
  - The arguments to pass to the custom delegate (non EventHandler-compatible).
See Also

IRaise<(Of <*T*>)> Interface
Raises Overload
Moq.Language Namespace

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

```csharp
public interface IReturns<TMock, TResult> : IHideObjectMembers
where TMock : class
```
Type Parameters

TMock
  Mocked type.
TResult
  Type of the return value from the expression.
See Also

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

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

Include Protected Members
Include Inherited Members

Moq

IReturns<(Of <(TMock, TResult)>)> Members

IReturns<(Of <(TMock, TResult)>)> Interface Methods See Also Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns(Func&lt;&lt;('TResult&gt;&gt;)&gt;)</td>
<td>Specifies the value to return.</td>
</tr>
<tr>
<td>Returns(TResult)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T)&gt;)&gt;(Func&lt;&lt;('T, TResult)&gt;)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10&gt;)&gt;)&gt;(Func&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult)&gt;)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11&gt;)&gt;)&gt;(Func&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, TResult)&gt;)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12&gt;)&gt;)&gt;(Func&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, TResult)&gt;)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13&gt;)&gt;)&gt;(Func&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, TResult)&gt;)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14&gt;)&gt;)&gt;(Func&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult)&gt;)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns&lt;&lt;('T, TResult)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, TResult)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, TResult)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, TResult)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;&lt;('T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult)&gt;)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
</tbody>
</table>
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>>>)

Returns<Of <<'T1, T2, T3, T4, T5>>>
(Func<Of <<'T1, T2, T3, T4, T5, TResult>>>)

Returns<Of <<'T1, T2, T3, T4, T5, T6, TResult>>>)

Returns<Of <<'T1, T2, T3, T4, T5, T6, T7, TResult>>>)

Returns<Of <<'T1, T2, T3, T4, T5, T6, T7, T8, T9, TResult>>>)

Returns<Of <<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, TResult>>>)

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

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

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

Include Protected Members
Include Inherited Members

Moq

IReturns<(<TMock, TResult>)> Methods

IReturns<(<TMock, TResult>)> Interface See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns(Func&lt;(&lt;'TResult'&gt;)&gt;))</td>
<td>Specifies the value to return.</td>
</tr>
<tr>
<td>Returns(TResult)</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;(&lt;'T'&gt;&gt;&gt;(Func&lt;(&lt;'T', TResult'&gt;)&gt;))</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;(&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10'&gt;&gt;&gt;(Func&lt;(&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult'&gt;)&gt;&gt;))</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;(&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11'&gt;&gt;&gt;(Func&lt;(&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, TResult'&gt;)&gt;&gt;))</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;(&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12'&gt;)&gt;&gt;)) (Func&lt;(&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, TResult'&gt;)&gt;&gt;))</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td>Returns&lt;(&lt;'T1', T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14'&gt;)&gt;&gt;)) (Func&lt;(&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult'&gt;)&gt;&gt;))</td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
</tbody>
</table>
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.
See Also

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

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members
C# Include Inherited Members
Moq
IReturns<Of <(TMock, TResult)>).Returns Method
IReturns<Of <(TMock, TResult)> Interface See Also Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Returns(Func&lt;Of &lt;'TResult&gt;&gt;)</code></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><code>Returns&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9&gt;&gt;)(Func&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult&gt;&gt;))&gt;</code></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><code>Returns&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11&gt;&gt;)(Func&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, TResult&gt;&gt;))&gt;</code></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><code>Returns&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12&gt;&gt;)(Func&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, TResult&gt;&gt;))&gt;</code></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><code>Returns&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13&gt;&gt;)(Func&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult&gt;&gt;))&gt;</code></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><code>Returns&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14&gt;&gt;)(Func&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult&gt;&gt;))&gt;</code></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><code>Returns&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult&gt;&gt;))&gt;</code></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
<tr>
<td><code>Returns&lt;(Of &lt;&lt;'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, TResult&gt;&gt;))&gt;</code></td>
<td>Specifies a function that will calculate the value to return from the method, retrieving the arguments for the invocation.</td>
</tr>
</tbody>
</table>
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.
See Also

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

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

IReturns<Of <(<<TMock, TResult>>)>...Returns

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#

IReturnsResult<TMock> Returns(
    Func valueFunction
)

Parameters

valueFunction
    Type: System:::Func<Of <(TResult)>>()
See Also

IReturns<(Of <-Mock, TResult>)> Interface
Returns Overload
Moq.Language Namespace

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

Moq

IReturns<(Of <(TMock, TResult)>)> Method (Func<(Of <(T1, T2, T3, T4, T5, T6, T7, T8, T9)>)>)

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

C#

IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7, T8, T9>(
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, TResult> valueFunction
)

Parameters

valueFunction
Type: System...Func<(Of <$T1, T2, T3, T4, T5, T6, T7, T8, T9, TResult$>)>
The function that will calculate the return value.
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.
T8  
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:

C#

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>()
).Returns((string arg1, string arg2, string arg3, string arg4, st
```
See Also

IReturns<(Of <<TMock, TResult>>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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>>)>>)

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

#### C#

```csharp
IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7, T8, T9, T1>
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult> valueFunction
```

#### Parameters

- **valueFunction**
  - Type: `System:::Func<Of <$>T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, TResult>>`
  - The function that will calculate the return value.
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.
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.
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:

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>()))
    .Returns((string arg1, string arg2, string arg3, string arg4, st
```
See Also

IReturns<(Of `<TMock, TResult>`)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)>)>)

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

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

C#

IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7, T8, T9, T1>

    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, TResult>

)

Parameters

valueFunction

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

The function that will calculate the return value.
### 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.

T8  
The type of the eighth argument of the invoked method.

T9  
The type of the ninth 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

The return value is calculated from the value of the actual method invocation arguments. Notice how the arguments are retrieved by simply declaring them as part of the lambda expression:

C#

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>()))
    .Returns((string arg1, string arg2, string arg3, string arg4, st
```
See Also

IReturns<(Of (<'TMock, TResult'>)>))> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)>)>)

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

C#

IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12>(
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, TResult>
)

Parameters

valueFunction

Type: System...Func<((T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, TResult)>>

The function that will calculate the return value.
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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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.
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#  

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>())
  .Returns((string arg1, string arg2, string arg3, string arg4, string arg5));
```


See Also

IReturns<(Of <(TMock, TResult)>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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)>)>)

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

C#

IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13>(
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, TResult>
)

Parameters

valueFunction

Type: System::Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, TResult>

The function that will calculate the return value.
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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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:

C#

mock.Setup(x => x.Execute(
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>())
    .Returns((string arg1, string arg2, string arg3, string arg4, st...
See Also

IReturns(Of (<'TMock, TResult'>))> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7, T8, T9, T1>

    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, TResult>

Parameters

valueFunction

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

    The function that will calculate the return value.
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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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.
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#  

```csharp
copy
mock.Setup(x => x.Execute(
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>())
  .Returns((
    string arg1, string arg2, string arg3, string arg4, st
See Also

IReturns<Of <(TMock, TResult)> >> Interface
Returns Overload
Moq.Language Namespace

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

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)>)))

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

IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7, T8, T9, T11, T12, T13, T14, T15>(
    Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, TResult>)

Parameters

valueFunction
Type: System::System::Func<(Of (Of <'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, TResult>)>)>
The function that will calculate the return value.
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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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.
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:

```csharp
mock.Setup(x => x.Execute(
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>(),
   It.IsAny<int>())
   .Returns((string arg1, string arg2, string arg3, string arg4, st
```
See Also

IReturns(Of (TMock, TResult))> Interface
Returns Overload
Moq.Language Namespace

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

Moq

IReturns<Of (Of (<'TMock, TResult>))>:

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

(Func<Of (Of (<'T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, TResult>))>)

IReturns<Of (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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

**C#**

```csharp
IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7, T8, T9, T1>
  Func<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, TResult>)
```

### Parameters

**valueFunction**

Type: `System::Func<(<T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, TResult>)>`

The function that will calculate the return value.
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.
T8  The type of the eighth argument of the invoked method.
T9  The type of the ninth 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.
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#  

```csharp
mock.Setup(x => x.Execute(
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>(),
    It.IsAny<int>()))
  .Returns((string arg1, string arg2, string arg3, string arg4, st
See Also

IReturns<(Of <(TMock, TResult)>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsResult<TMock> Returns<T>(
    Func valueFunction
)

Parameters

valueFunction
    Type: System..::.Func<(<T, TResult>)>
Type Parameters

T
See Also

IReturns<(Of <(<TMock, TResult>>>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsResult<TMock> Returns<T1, T2>(
    Func valueFunction
)

Parameters

valueFunction
    Type: System...Func<(Of <+T1, T2, TResult}>)>
<table>
<thead>
<tr>
<th>Type Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
</tr>
<tr>
<td>T2</td>
</tr>
</tbody>
</table>
See Also

IReturns<(Of <(‘TMock, TResult)>))> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsResult<TMock> Returns<T1, T2, T3>(
    Func valueFunction
)

Parameters

valueFunction
    Type: System...:Func<(Of <(T1, T2, T3, TResult)>)>
Type Parameters

T1
T2
T3
See Also

IReturns<(Of <(TMock, TResult)>)> Interface
Returns Overload
Moq.Language Namespace

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

 Interface IReturns

 Returns <(Of ( Of (<'TMock, TResult>)))> Method (Func)

 IReturns<(Of ( 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
 Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsResult<TMock> Returns<T1, T2, T3, T4>(
    Func valueFunction
)

Parameters

valueFunction
    Type: System...Func<(Of <(T1, T2, T3, T4, TResult)>)>
Type Parameters

T1
T2
T3
T4
See Also

IReturns(Of (Of TMock, TResult>))> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5>(
    Func valueFunction
)

Parameters

valueFunction
    Type: System...Func<(Of <(T1, T2, T3, T4, T5, TResult)>)>
Type Parameters

T1
T2
T3
T4
T5
See Also

IReturns<(Of <(TMock, TResult)>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

#### C#

```csharp
IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6>(
    Func valueFunction
)
```

### Parameters

- `valueFunction`
  - Type: `System.Func<T1, T2, T3, T4, T5, T6, TResult>`
Type Parameters

T1
T2
T3
T4
T5
T6
See Also

IReturns<(Of <(TMock, TResult)>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7>(
    Func valueFunction
)

Parameters

valueFunction
Type: System...Func<(Of <(T1, T2, T3, T4, T5, T6, T7, TResult)>)>
## Type Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td></td>
</tr>
<tr>
<td>T5</td>
<td></td>
</tr>
<tr>
<td>T6</td>
<td></td>
</tr>
<tr>
<td>T7</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IReturns<(Of ( Of 'TMock, TResult>>) Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
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
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

IReturnsResult<TMock> Returns<T1, T2, T3, T4, T5, T6, T7, T8>(
    Func valueFunction
)

Parameters

valueFunction
    Type: System...Func<(Of <(T1, T2, T3, T4, T5, T6, T7, T8, TResult)>)>
Type Parameters

T1
T2
T3
T4
T5
T6
T7
T8
See Also

IReturns<(Of <(TMock, TResult>)>)> Interface
Returns Overload
Moq.Language Namespace

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

Moq

IReturns<(<TMock, TResult>)>..:.Returns Method (TResult)

IReturns<(<TMock, TResult>)> 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)
**Syntax**

**C#**

```csharp
IReturnsResult<TMock> Returns(
    TResult value
)
```

**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#  

```csharp
mock.Setup(x => x.Execute("ping"))
    .Returns(true);
```
See Also

IReturns<(Of (('TMock, TResult>))> Interface
Returns Overload
Moq.Language Namespace

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

Moq

IReturnsGetter<(Of <('TMock, TProperty)>)> Interface

Members See Also Send Feedback

Defines the Returns verb for property get setups.

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

```csharp
public interface IReturnsGetter<TMock, TProperty> : IHideObjectMembers
where TMock : class
```
Type Parameters

TMock
   Mocked type.
TProperty
   Type of the property.
See Also

IReturnsGetter<(Of <(TMock, TProperty)>)> Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
IReturnsGetter<(Of <'(TMock, TProperty)>)> Members
IReturnsGetter<(Of <'(TMock, TProperty)>)> Interface Methods See Also Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Returns(Func&lt;(&lt;TResult&gt;)&gt;)</code></td>
<td></td>
</tr>
<tr>
<td><code>Returns(TProperty)</code></td>
<td>Specifies the value to return.</td>
</tr>
</tbody>
</table>
See Also

IReturnsGetter<(Of <(TMock, TProperty)>)> Interface
Moq.Language Namespace

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

`IReturnsGetter(Of <(TMock, TProperty)>)` Methods

`IReturnsGetter(Of <(TMock, TProperty)>)` Interface

See Also

Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns(Func&lt;Of&lt;(&lt;TResult&gt;)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>Returns(TProperty)</td>
<td>Specifies the value to return.</td>
</tr>
</tbody>
</table>
See Also

IReturnsGetter(Of <(TMock, TProperty)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Moq
IReturnsGetter(Of <('TMock, TProperty')>)..::..Returns Method
IReturnsGetter(Of <('TMock, TProperty')>) Interface See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns(Func(Of (&lt;'TResult'&gt;)&gt;))</td>
<td></td>
</tr>
<tr>
<td>Returns(TProperty)</td>
<td>Specifies the value to return.</td>
</tr>
</tbody>
</table>
See Also

IReturnsGetter<(Of '<(TMock, TProperty)>)> Interface
IReturnsGetter<(Of '<(TMock, TProperty)>)> Members
Moq.Language Namespace

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

IReturnsResult<TMock> Returns(
    Func valueFunction
)

Parameters

valueFunction
Type: System..::.Func<Of (<TResult>)>
See Also

IReturnsGetter<(<TMock, TProperty>)> Interface
Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies the value to return.

**Namespace:** Moq.Language

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

C#

IReturnsResult<TMock> Returns(
    TProperty value
)

Parameters

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

Return a true value from the property getter call:

C#

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

IReturnsGetter<(<TMock, TProperty>)> Interface Returns Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Implements the fluent API.

**Namespace:** Moq.Language
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

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

T
See Also

ISetupConditionResult<(Of <(<T>)>)> Members

Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `ISetupConditionResult<Of <('T')>>` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup(Expression&lt;(Of &lt;&lt;='&lt;(Action&lt;(Of &lt;&lt;='(T))&gt;&gt;)&gt;&gt;))</td>
<td>The expectation will be considered only in the former condition.</td>
</tr>
<tr>
<td>Setup&lt;(Of &lt;&lt;='(TResult)&gt;&gt;) (Expression&lt;(Of &lt;&lt;='(Func&lt;(Of &lt;&lt;='(T, TResult)&gt;&gt;)&gt;&gt;)&gt;&gt;))</td>
<td></td>
</tr>
<tr>
<td>SetupGet&lt;(Of &lt;&lt;='(TProperty)&gt;&gt;)</td>
<td>Setups the set.</td>
</tr>
<tr>
<td>SetupSet(Action&lt;(Of &lt;&lt;='(T)&gt;&gt;))</td>
<td>Setups the set.</td>
</tr>
<tr>
<td>SetupSet&lt;(Of &lt;&lt;='(TProperty)&gt;&gt;) (Action&lt;(Of &lt;&lt;='(T)&gt;&gt;))</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ISetupConditionResult(Of (Of 'T)>)> Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `ISetupConditionResult<Of <('T')>>` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup(Expression&lt;(Of &lt;&lt;'(Action&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;)))&gt;))</td>
<td>The expectation will be considered only in the former condition.</td>
</tr>
</tbody>
</table>
| Setup<(Of <<'(TResult)>)>))>)) (Expression<(Of <<'(Func<(Of <<'(T, TResult)>)>)))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>))>)}
See Also

ISetupConditionResult(Of (T)>) Interface
Moq.Language Namespace

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

- Include Protected Members
- Include Inherited Members

Moq

ISetupConditionResult<Of <(T)>>...Setup Method

ISetupConditionResult<Of <(T)>> Interface See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup(Expression(Of &lt;&lt;'(Action(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;)&gt;)&gt;&gt;))</td>
<td>The expectation will be considered only in the former condition.</td>
</tr>
<tr>
<td>Setup(Of &lt;&lt;'(TResult)&gt;&gt;)&gt;)</td>
<td></td>
</tr>
<tr>
<td>(Expression(Of &lt;&lt;'(Func(Of &lt;&lt;'(T, TResult)&gt;&gt;)&gt;&gt;)&gt;)&gt;&gt;))</td>
<td></td>
</tr>
</tbody>
</table>
See Also

**ISetupConditionResult(Of (<T>)>) Interface**
**ISetupConditionResult(Of (<T>)>) Members**
**Moq.Language Namespace**

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
ISetupConditionResult<(<'T'>)>...Setup Method (Expression<(<'Action'<(<'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#

```csharp
ISetup<T> Setup(
    Expression<Action<T>> expression
)
```

### Parameters

**expression**  
Type: `System.Linq.Expressions..:::Expression<Of (Of (Of (Of 'Action<Of (Of (Of 'T)>>)>>)>>)`

### Return Value
See Also

ISetupConditionResult(Of (T)) Interface
Setup Overload
Moq.Language Namespace

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

Moq

ISetupConditionResult(Of (Of (<T>))...Setup(Of (Of (<T>))>)

Method (Expression(Of (Of (<Func>))>)

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

[Missing <summary> documentation for

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

ISetup<T, TResult> Setup<TResult>(
    Expression<Func> expression
)

Parameters

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

TResult
See Also

ISetupConditionResult<((Of <(T)>))> Interface
Setup Overload
Moq.Language Namespace

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

Moq

ISetupConditionResult(Of (Of (<'T'>)>))...SetupGet(Of (<'TProperty'>)>)

Method

ISetupConditionResult(Of (Of (<'T'>)>)) Interface See Also Send Feedback

[Missing <summary> documentation for

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

ISetupGetter<T, TProperty> SetupGet<TProperty>(
    Expression<Func> expression
)

Parameters

expression
    Type: System.Linq.Expressions.Expression<Func<Func<T, TResult>>>>()
Type Parameters

TPROPERTY
See Also

ISetupConditionResult(Of (Of T)>>) Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Include Protected Members
Include Inherited Members
Moq
ISetupConditionResult<Of <(<'T'>)>)...SetupSet Method
ISetupConditionResult<Of <(<'T'>)>) Interface See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SetupSet</strong>&lt;br&gt;<strong>Setups the set.</strong></td>
<td><strong>SetupSet</strong>&lt;br&gt;Setups the set.</td>
</tr>
<tr>
<td><strong>SetupSet( Action&lt;&lt;'(T)&gt;&gt;)&gt;&gt;)</strong></td>
<td><strong>SetupSet( Of &lt;&lt;'(TProperty)&gt;&gt;)&gt;&gt;)(Action&lt;&lt;'(T)&gt;&gt;)&gt;&gt;)</strong></td>
</tr>
</tbody>
</table>
See Also

ISetupConditionResult(Of T) Interface
ISetupConditionResult(Of T) Members
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C# Moq
ISetupConditionResult<Of (Of ('T)<>)>...:..SetupSet Method (Action<Of (Of ('T)<>)>)
ISetupConditionResult<Of (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#**

```csharp
ISetup<T> SetupSet(
    Action<T> setterExpression
)
```

**Parameters**

`setterExpression`

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

The setter expression.

**Return Value**
See Also

ISetupConditionResult(Of (Of T)) Interface
SetupSet Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
ISetupConditionResult<(Of <(T)>)>...:..SetupSet<(Of <(T)Property)>)
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#

ISetupSetter<T, TProperty> SetupSet<TProperty>(
    Action<T> setterExpression
)

Parameters

setterExpression
    Type: System:::Action<Of (<>T<>)>
    The setter expression.
Type Parameters

TProperty
   The type of the property.

Return Value
See Also

ISetupConditionResult(Of (<T>))> Interface
SetupSet Overload
Moq.Language Namespace

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

ISetupSequentialResult(Of (Of TResult)>) Interface

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
**Syntax**

C#

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

TResult
See Also

ISetupSequentialResult<(Of <(‘TResult’)>)> Members
Moq.Language Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns</td>
<td>Returns value</td>
</tr>
<tr>
<td>Throws(Exception)</td>
<td>Throws an exception</td>
</tr>
<tr>
<td>Throws&lt;Of &lt;&lt;(TEexception)&gt;&gt;</td>
<td>Throws an exception</td>
</tr>
</tbody>
</table>
See Also

`ISetupSequentialResult<`<`TResult`>`>` Interface

`Moq.Language Namespace`

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

ISetupSequentialResult(Of (TResult)> methods

See Also Send Feedback

The `ISetupSequentialResult(Of (TResult)>)` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns</td>
<td>Returns value</td>
</tr>
<tr>
<td>Throws(Exception)</td>
<td>Throws an exception</td>
</tr>
<tr>
<td>Throws&lt;(Of &lt;&lt;'(TException&gt;')&gt;&gt;)()()()()</td>
<td>Throws an exception</td>
</tr>
</tbody>
</table>
See Also

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

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

**Namespace:** Moq.Language

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

C#

```csharp
ISetupSequentialResult<TResult> Returns(TResult value)
```

Parameters

value
  Type: TResult
See Also

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

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws&lt;(Of &lt;&lt;'(TException)&gt;)&gt;&gt;0000</td>
<td>Throws an exception</td>
</tr>
<tr>
<td>Throws(Exception)</td>
<td>Throws an exception</td>
</tr>
</tbody>
</table>
See Also

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

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

ISetupSequentialResult<Of (Of<TResult>...)>.::.Throws<Of (Of<TException>...)>

Method

ISetupSequentialResult<Of (Of<TResult>...)>

Interface

See Also

Send Feedback

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
See Also

ISetupSequentialResult(Of TResult) Interface
Throws Overload
Moq.Language Namespace

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

ISetupSequentialResult(Of TResult).....Throws Method (Exception)

Interface See Also Send Feedback

Throws an exception

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

void Throws(
    Exception exception
)

Parameters

exception
    Type: System....Exception
See Also

**ISetupSequentialResult(Of TResult)> Interface**

Throws Overload

**Moq.Language Namespace**

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

Moq

IThrows Interface

Members See Also Send Feedback

Defines the Throws verb.

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

**C#**

```csharp
public interface IThrows : IHideObjectMembers
```
See Also

IThrows Members
Moq.Language Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Throws(ForEach)</code></td>
<td>Specifies the exception to throw when the method is invoked.</td>
</tr>
<tr>
<td><code>Throws&lt;Of $&lt;'(TException)')&gt;&gt;()'</code></td>
<td>Specifies the type of exception to throw when the method is invoked.</td>
</tr>
</tbody>
</table>
See Also

IThrows Interface
Moq.Language Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Throws(Exception)</code></td>
<td>Specifies the exception to throw when the method is invoked.</td>
</tr>
<tr>
<td><code>Throws&lt;Of (TException)&gt;)</code></td>
<td>Specifies the type of exception to throw when the method is invoked.</td>
</tr>
</tbody>
</table>
See Also

IThrows Interface
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Include Protected Members
Include Inherited Members
Moq
IThrows...Throws Method
IThrows Interface See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws&lt;(Of 'Exception)()()()</td>
<td>Specifies the type of exception to throw when the method is invoked.</td>
</tr>
<tr>
<td>Throws(Exception)</td>
<td>Specifies the exception to throw when the method is invoked.</td>
</tr>
</tbody>
</table>
See Also

IThrows Interface
IThrows Members
Moq.Language Namespace

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

Moq

IThrows......Throws<(Of <(TException)>)> Method

IThrows Interface Example See Also Send Feedback

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

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
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#

```csharp
mock.Setup(x => x.Execute(""))
    .Throws<ArgumentException>();
```
See Also

IThrows Interface
Throws Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies the exception to throw when the method is invoked.

**Namespace:** Moq.Language

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

C#

IThrowsResult Throws(
    Exception exception
)

Parameters

exception
Type: System.Exception
Exception instance to throw.
Examples

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

C#

```csharp
mock.Setup(x => x.Execute(""))
    .Throws(new ArgumentException());
```
See Also

IThrows Interface
Throws Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Defines the Verifiable verb.

**Namespace:** Moq.Language  
**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public interface IVerifies : IHideObjectMembers
See Also

IVerifies Members
Moq.Language Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifiable()</td>
<td>Marks the expectation as verifiable, meaning that a call to <code>Verify()</code> will check if this particular expectation was met.</td>
</tr>
<tr>
<td>Verifiable(String)</td>
<td>Marks the expectation as verifiable, meaning that a call to <code>Verify()</code> will check if this particular expectation was met, and specifies a message for failures.</td>
</tr>
</tbody>
</table>
See Also

IVerifies Interface
Moq.Language Namespace

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

See Also Send Feedback
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifiable()</td>
<td>Marks the expectation as verifiable, meaning that a call to Verify() will check if this particular expectation was met.</td>
</tr>
<tr>
<td>Verifiable(String)</td>
<td>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.</td>
</tr>
</tbody>
</table>
See Also

IVerifies Interface
Moq.Language Namespace

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

IVerifies

Verifiable Method

See Also

Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifiable()</td>
<td>Marks the expectation as verifiable, meaning that a call to <code>Verify()</code> will check if this particular expectation was met.</td>
</tr>
<tr>
<td>Verifiable(String)</td>
<td>Marks the expectation as verifiable, meaning that a call to <code>Verify()</code> will check if this particular expectation was met, and specifies a message for failures.</td>
</tr>
</tbody>
</table>
See Also

IVerifies Interface
IVerifies Members
Moq.Language Namespace

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

IVerifies Interface Example See Also Send Feedback

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

Namespace: Moq.Language
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
C#

void Verifiable()
Examples

The following example marks the expectation as verifiable:

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

IVerifies Interface
Verifiable Overload
Moq.Language Namespace

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

Moq

IVerifies....Verifiable Method (String)

IVerifies Interface Example See Also Send Feedback

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

```csharp
void Verifiable(
    string failMessage
)
```

**Parameters**

`failMessage`  
Type: `System.String`
Examples

The following example marks the expectation as verifiable:

C#

```csharp
mock.Expect(x => x.Execute("ping"))
    .Returns(true)
    .Verifiable("Ping should be executed always!");
```
See Also

IVerifies Interface
Verifiable Overload
Moq.Language Namespace

Send comments on this topic to moqdisc@googlegroups.com
Importing this namespace will enable the Protected() method on mocks, enabling expectations on protected members by specifying the member name as a string.
# Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ItExpr</td>
<td>Allows the specification of a matching condition for an argument in a protected member setup, rather than a specific argument value. &quot;ItExpr&quot; refers to the argument being matched. Enables the Protected() method on Mock&lt;Of (Of &lt;(T)&gt;), 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.</td>
</tr>
<tr>
<td>ProtectedExtension</td>
<td></td>
</tr>
<tr>
<td>PrivateMethod</td>
<td></td>
</tr>
<tr>
<td>Interface</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>IProtectedMock&lt;Of (Of &lt;'TMock'&gt;)&gt;</td>
<td>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.</td>
</tr>
</tbody>
</table>

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
IProtectedMock(Of (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)
public interface IProtectedMock<TMock> : IHideObjectMembers
where TMock : class
Type Parameters

TMock
See Also

IProtectedMock<((Of (<TMock>)>)> Members
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `IProtectedMock<Of <('TMock')>>` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup(String, array&lt;Object&gt;[][])</td>
<td>Specifies a setup for a void method invocation with the given voidMethodName, optionally specifying arguments for the method call.</td>
</tr>
<tr>
<td>Setup(Of &lt;&gt;'(TResult&gt;)&gt;&gt;(String, array&lt;Object&gt;[][])</td>
<td>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.</td>
</tr>
<tr>
<td>SetupGet(Of &lt;&gt;'(TProperty&gt;)&gt;&gt;</td>
<td>Specifies a setup for an invocation on a property getter with the given propertyName.</td>
</tr>
<tr>
<td>SetupSet(Of &lt;&gt;'(TProperty&gt;)&gt;&gt;</td>
<td>Specifies a setup for an invocation on a property setter with the given propertyName.</td>
</tr>
<tr>
<td>Verify(String, Times, array&lt;Object&gt;[][])</td>
<td>Specifies a verify for a void method with the given methodName, optionally specifying arguments for the method call. Use in conjunction with the default Loose.</td>
</tr>
<tr>
<td>Verify(Of &lt;&gt;'(TResult&gt;)&gt;&gt;(String, Times, array&lt;Object&gt;[][])</td>
<td>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.</td>
</tr>
<tr>
<td>VerifyGet(Of &lt;&gt;'(TProperty&gt;)&gt;&gt;</td>
<td>Specifies a verify for an invocation on a property getter with the given propertyName. The invocation was not call the times specified by times.</td>
</tr>
<tr>
<td>VerifySet(Of &lt;&gt;'(TProperty&gt;)&gt;&gt;</td>
<td>Specifies a setup for an invocation on a property setter with the given propertyName.</td>
</tr>
</tbody>
</table>
See Also

IProtectedMock<(Of <(TMock)>)> Interface
Moq.Protected Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setup(String, array&lt;Object&gt;[][])</strong></td>
<td>Specifies a setup for a void method invocation with the given voidMethodName, optionally specifying arguments for the method call.</td>
</tr>
<tr>
<td><strong>Setup&lt;(&lt;'TResult'&gt;)&gt;(String, array&lt;Object&gt;[][])</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>SetupGet&lt;(&lt;'TProperty'&gt;)()</strong></td>
<td>Specifies a setup for an invocation on a property getter with the given propertyName.</td>
</tr>
<tr>
<td><strong>SetupSet&lt;(&lt;'TProperty'&gt;)()</strong></td>
<td>Specifies a setup for an invocation on a property setter with the given propertyName.</td>
</tr>
<tr>
<td><strong>Verify(String, Times, array&lt;Object&gt;[][])</strong></td>
<td>Specifies a verify for a void method with the given methodName, optionally specifying arguments for the method call. Use in conjunction with the default <strong>Loose</strong>.</td>
</tr>
<tr>
<td><strong>Verify&lt;(&lt;'TResult'&gt;)&gt;(String, Times, array&lt;Object&gt;[][])</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>VerifyGet&lt;(&lt;'TProperty'&gt;)()</strong></td>
<td>Specifies a verify for an invocation on a property getter with the given propertyName. The invocation was not call the times specified by times.</td>
</tr>
<tr>
<td><strong>VerifySet&lt;(&lt;'TProperty'&gt;)()</strong></td>
<td>Specifies a setup for an invocation on a property setter with the given propertyName.</td>
</tr>
</tbody>
</table>
See Also

IProtectedMock<(Of <(TMock)>)> Interface
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Include Protected Members
Include Inherited Members
Moq
IProtectedMock(Of TMock) Interface

See Also
Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Setup<
| <'(TResult)>)\>(String,
| array<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>[][]) | Specifies a setup for a void method invocation with the given voidMethodName, optionally specifying arguments for the method call. |
See Also

IProtectedMock(Of TMock) Interface
IProtectedMock(Of TMock) Members
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
IProtectedMock<Of [TMock]> Method
(String, array<Object>[])[]

IProtectedMock<Of [TMock]> Interface See Also Send Feedback

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#**

```csharp
ISetup<TMock, TResult> Setup<TResult>(
    string methodOrPropertyName,
    params Object[] args
)
```

**Parameters**

**methodOrPropertyName**
- Type: `System::::String`
- The name of the method or property to be invoked.

**args**
- Type: `array<System::::Object>[]`[]
- 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.
See Also

IProtectedMock<Of <(TMock)>)> Interface
Setup Overload
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
Specifies a setup for a void method invocation with the given voidMethodName, optionally specifying arguments for the method call.

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

**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
### Syntax

**C#**

```csharp
ISetup<TMock> Setup(
    string voidMethodName,
    params Object[] args
)
```

#### Parameters

**voidMethodName**
- Type: `System::String`
- The name of the void method to be invoked.

**args**
- Type: `array<System::Object>[]`()
- The optional arguments for the invocation. If argument matchers are used, remember to use `ItExpr` rather than `It`. 
See Also

IProtectedMock(Of ( Of ( TMock ))) Interface
Setup Overload
Moq.Protected Namespace

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

```csharp
ISetupGetter<TMock, TProperty> SetupGet<TProperty>(
    string propertyName
)
```

## Parameters

**propertyName**

Type: `System::String`

The name of the property.
Type Parameters

TProperty
    The type of the property.
See Also

IProtectedMock<(Of <(TMock)>)> Interface
Moq.Protected Namespace

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

ISetupSetter<TMock, TProperty> SetupSet<TProperty>(
    string propertyName,
    Object value
)

Parameters

propertyName
Type: System.::.String
The name of the property.

value
Type: System.::.Object
The property value. If argument matchers are used, remember to use ItExpr rather than It.
Type Parameters

TProperty
   The type of the property.
See Also

IProtectedMock<Of <(TMock)> > Interface
Moq.Protected Namespace

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

Include Protected Members
Include Inherited Members

Moq

IProtectedMock(Of (Of (<'TMock'>))>....Verify Method

IProtectedMock(Of (Of (<'TMock'>))> Interface See Also Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Verify&lt;(Of &lt;'(TResult)&gt;)&gt;&gt;(String, Times, array&lt;Object&gt;[][])</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>Verify(String, Times, array&lt;Object&gt;[][])</code></td>
<td>Specifies a verify for a void method with the given methodName, optionally specifying arguments for the method call. Use in conjunction with the default <code>[Loose]</code>.</td>
</tr>
</tbody>
</table>
See Also

IProtectedMock<(<TMock>)> Interface
IProtectedMock<(<TMock>)> Members
Moq.Protected Namespace

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

```csharp
void Verify<TResult>(
    string methodName,
    Times times,
    params Object[] args
)
```

### 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<System:::Object>[]`[]

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

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq.....MockException</td>
<td>The invocation was not call the times specified by times.</td>
</tr>
</tbody>
</table>
See Also

IProtectedMock<Of <(TMock)>) Interface
Verify Overload
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
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

C#

void Verify(
    string methodName,
    Times times,
    params Object[] args
)

Parameters

methodName
    Type: System:::String
    The name of the void method to be verified.

times
    Type: Moq:::Times
    The number of times a method is allowed to be called.

args
    Type: array<System:::Object>[]
    The optional arguments for the invocation. If argument matchers are used, remember to use ItExpr rather than It.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moq.........MockException</td>
<td>The invocation was not call the times specified by times.</td>
</tr>
</tbody>
</table>
See Also

IProtectedMock(Of (Of TMock>)>) Interface
Verify Overload
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
IProtectedMock(Of (Of (<'TMock'>)>)>. VerifyGet(Of (<'TProperty'>)>)>
Method
IProtectedMock(Of (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#

```csharp
void VerifyGet<TProperty>(
    string propertyName,
    Times times
)
```

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.
See Also

IProtectedMock<Of (<TMock>)> Interface
Moq.Protected Namespace

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

```csharp
void VerifySet<TProperty>(
    string propertyName,
    Times times,
    Object value
)
```

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.

value
  Type: System:::Object
  The property value.
Type Parameters

TProperty
The type of the property. If argument matchers are used, remember to use `ItExpr` rather than `It`. 
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Moq::MockException</code></td>
<td>The invocation was not call the times specified by times.</td>
</tr>
</tbody>
</table>
See Also

IProtectedMock<Of <(<TMock>())>> Interface
Moq.Protected Namespace

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

Allows the specification of a matching condition for an argument in a protected member 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)
Syntax

C#

public static class ItExpr
Remarks

Use this variant of argument matching instead of `It` 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
See Also

*ItExpr Members*
*Moq.Protected Namespace*

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Is</strong>&lt;(Of&lt;br&gt;`&lt;'(TValue&gt;)&gt;&gt;)**</td>
<td>Matches any value of the given TValue type.</td>
</tr>
<tr>
<td><strong>IsAny</strong>&lt;(Of&lt;br&gt;`&lt;'(TValue&gt;)&gt;&gt;)**</td>
<td>Matches any value that is in the range specified.</td>
</tr>
<tr>
<td><strong>IsNull</strong>&lt;(Of&lt;br&gt;`&lt;'(TValue&gt;)&gt;&gt;)**</td>
<td>Matches a null value of the given TValue type.</td>
</tr>
<tr>
<td><strong>IsRegex</strong>&lt;String&gt;</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
<tr>
<td><strong>IsRegex</strong>&lt;String, RegexOptions&gt;</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
</tbody>
</table>
See Also

ItExpr Class
Moq.Protected Namespace

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is&lt;(Of &lt;&lt;'(TValue&gt;')&gt;&gt;)</td>
<td>Matches any value of the given TValue type.</td>
</tr>
<tr>
<td>IsAny&lt;(Of &lt;&lt;'(TValue&gt;')&gt;&gt;)</td>
<td>Matches any value that is in the range specified.</td>
</tr>
<tr>
<td>IsInRange&lt;(Of &lt;&lt;'(TValue&gt;')&gt;&gt;)</td>
<td>Matches a null value of the given TValue type.</td>
</tr>
<tr>
<td>IsNull&lt;(Of &lt;&lt;'(TValue&gt;')&gt;&gt;)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
<tr>
<td>IsRegex(String)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
<tr>
<td>IsRegex(String, RegexOptions)</td>
<td>Matches a string argument if it matches the given regular expression pattern.</td>
</tr>
</tbody>
</table>
See Also

ItExpr Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
C#
Moq
ItExpr......Is<(Of <('TValue)>)> Method
ItExpr Class See Also Send Feedback

[Missing <summary> documentation for

Namespace: Moq.Protected
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public static Expression Is<TValue>(
    Expression<Func> match
)

Parameters

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>TValue</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ItExpr Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
ItExpr...:..IsAny<Of <('<TValue>')>> Method

Matches any value of the given TValue type.

Namespace:  Moq.Protected
Assembly:  Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public static Expression IsAny<TValue>()
Type Parameters

TValue

Type of the value.
**Remarks**

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

C#

```csharp
// Throws an exception for a call to Remove with any string value.
mock.Protected()  
  .Setup("Remove", ItExpr.IsAny<string>())  
  .Throws(new InvalidOperationException());
```
See Also

ItExpr Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
ItExpr.<::<InRange<(Of <(<TValue>)>> Method

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#**

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

**Parameters**

**from**
- Type: TValue
  - The lower bound of the range.

**to**
- Type: TValue
  - The upper bound of the range.

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

TValue
Type of the argument to check.
Examples

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

C#

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

*ItExpr Class*
*Moq.Protected Namespace*

Send comments on this topic to [moqdisc@googlegroups.com](mailto:moqdisc@googlegroups.com)
ItExpr...:.IsNull<(Of <(<TValue>>)>) Method

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)
C#

public static Expression 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.
// Throws an exception for a call to Remove with a null string value
mock.Protected()
    .Setup("Remove", ItExpr.IsNull<string>())
    .Throws(new InvalidOperationException());
See Also

ItExpr Class
Moq.Protected Namespace

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

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

ItExpr Class
ItExpr Members
Moq.Protected Namespace

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

**Namespace:** Moq.Protected

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

C#

```csharp
public static Expression IsRegex(
    string regex
)
```

Parameters

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

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

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

- ItExpr Class
- IsRegex Overload
- Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
ItExpr.IsRegex Method (String, RegexOptions)

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#

```csharp
public static Expression IsRegex(
    string regex,
    RegexOptions options
)
```

Parameters

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

options
Type: `System.Text.RegularExpressions.RegexOptions`
The options used to interpret the pattern.
Examples

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

C#

```csharp
mock.Protected()
    .Returns(1);
```
See Also

ItExpr Class
IsRegex Overload
Moq.Protected Namespace

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

Moq

ProtectedExtension Class

Members See Also Send Feedback

Enables the Protected() method on Mock<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.

Namespace: Moq.Protected
Assembly: Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
Syntax

C#

public static class ProtectedExtension
Inheritance Hierarchy

System...Object
  Moq.Protected...ProtectedExtension
See Also

ProtectedExtension Members
Moq.Protected Namespace

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

The **ProtectedExtension** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Protected&lt;(Of &lt;&lt;'(T)&gt;&gt;)</code></td>
<td>Enable protected setups for the mock.</td>
</tr>
</tbody>
</table>
See Also

ProtectedExtension Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
The `ProtectedExtension` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;s&gt;</code> Protected&lt;Of &lt;&lt;'(T)&gt;&gt;`</td>
<td>Enable protected setups for the mock.</td>
</tr>
</tbody>
</table>
See Also

ProtectedExtension Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com
Enable protected setups for the mock.

**Namespace:** Moq.Protected

**Assembly:** Moq (in Moq.dll) Version: 4.0.10827.0 (4.0.0.0)
## Syntax

### C#

```csharp
public static IProtectedMock<T> Protected<T>(
    this Mock<T> mock
)
where T : class
```

### 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.
See Also

ProtectedExtension Class
Moq.Protected Namespace

Send comments on this topic to moqdisc@googlegroups.com