

## Tungsten

W<sup>74</sup>

# Welcome to Tungsten.Suite

Tungsten.Suite is a refactoring of the original Tungsten.X libraries. Tungsten is now a suite of NuGet packages which can be referenced individually or as a whole by referencing Tungsten.Suite.

These are the NuGet packages referenced by Tungsten.Suite

- Tungsten.ArrayMethods
- Tungsten.As
- Tungsten.CallResult
- Tungsten.Console
- Tungsten.EventTemplate
- Tungsten.Encryption
- Tungsten.From
- Tungsten.IO.Pipes
- Tungsten.Lockable
- Tungsten.Logging
- Tungsten.Net
- Tungsten.Property
- Tungsten.Threading
- Tungsten.Threading.Lockers

These projects have NuGet packages, but are not included in Tungsten.Suite because they only support the .Net Framework

- Tungsten.Domains
- Tungsten.Firewall

- Tungsten.InterProcess

## ► See Also

[Other Resources](#)

[Version History](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Version History

The topics in this section describe the various changes made to Tungsten over the life of the project.

## ▪ Version History

Select a version below to see a description of its changes.

- [Version 2.0.0](#)

## ▪ See Also

Other Resources

[Welcome to Tungsten.Suite](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Version 2.0.0

Version 2.0.0 was released on 2/28/2018

## ▪ Changes in This Release

- Initial release of the refactored Tungsten libraries

## ▪ See Also

Other Resources

[Version History](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# W Namespace

[Missing <summary> documentation for "N:W"]

## ► Classes

Class	Description
 <a href="#">ArrayMethods</a>	Methods to peek and modify arrays
 <a href="#">AsExtensions</a>	Extensions which convert objects of one type to another
 <a href="#">CallResult</a>	A non-generic return value for a function. CallResult encapsulates a success/failure and an exception.
 <a href="#">CallResultTResult</a>	Generic class to be used as a return value. CallResult encapsulates a success/failure, an exception and a return value.
 <a href="#">ConsoleStringExtensions</a>	Extension methods related to the Console
 <a href="#">EventTemplateTSender</a>	Wraps the functionality of delegate, event and RaiseXXX into a single class

	<a href="#">EventTemplateTSender, Arg1</a>	Wraps the functionality of delegate, event and RaiseXXX into a single class
	<a href="#">EventTemplateTSender, Arg1, Arg2</a>	Wraps the functionality of delegate, event and RaiseXXX into a single class
	<a href="#">EventTemplateTSender, Arg1, Arg2, Arg3</a>	Wraps the functionality of delegate, event and RaiseXXX into a single class
	<a href="#">EventTemplateTSender, Arg1, Arg2, Arg3, Arg4</a>	Wraps the functionality of delegate, event and RaiseXXX into a single class
	<a href="#">EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5</a>	Wraps the functionality of delegate, event and RaiseXXX into a single class
	<a href="#">FromExtensions</a>	Extensions which convert objects of one type to another
	<a href="#">Lockable TValue</a>	Extends LockableSlim with ValueChangedDelegate notification
	<a href="#">LockableSlim TValue</a>	Uses ReaderWriterLock to provide thread-safe access to an underlying value

	<a href="#">Property TValue</a>	A Property with no owner (self-owned)
	<a href="#">Property TOwner, TValue</a>	A generic Property with an owner
	<a href="#">PropertyBase TOwner, TValue</a>	
	<a href="#">PropertyChangedNotifier</a>	This is a base class for supporting INotifyPropertyChanged
	<a href="#">PropertyHost</a>	Provides a base class to automate the IsDirty, MarkAsClean and InitializeProperties functionality Note that this class does not support INotifyPropertyChanged and is not intended to host owned properties (though nothing prevents you from doing so)
	<a href="#">PropertyHostExtensions</a>	Extension methods related to W.PropertyHost or any class which has multiple W.Property members
	<a href="#">PropertyHostNotifier</a>	Provides a base class to automate the IsDirty, MarkAsClean and InitializeProperties functionality Note that this class inherits PropertyChangedNotifier for INotifyPropertyChanged support



## PropertySlim TValue

PropertySlim extends W.Lockable by adding support for INotifyPropertyChanged

## Interfaces

Interface	Description
↳ <a href="#">IOwnedProperty</a>	Used by PropertyHostMethods.InitializeProperties to find properties on which to set the owner. This interface is not used by self-owned properties.
↳ <a href="#">IProperty</a>	The base interface which Property must support
↳ <a href="#">IPropertyValue</a>	The base interface which Property must support

## Delegates

Delegate	Description
⌚ <a href="#">ValueChangedDelegate TValue</a>	Raised when the value has changed

## Tungsten

W<sup>74</sup>

# ArrayMethods Class

Methods to peek and modify arrays

## ► Inheritance Hierarchy

[SystemObject](#) [WArrayMethods](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static class ArrayMethods
```

The [ArrayMethods](#) type exposes the following members.

## ► Methods

	Name	Description
≡ ♡ S	<a href="#">AppendT</a>	Appends the items to an array, resizing the array as necessary
≡ ♡ S	<a href="#">InsertT</a>	Appends the items to an array, resizing the array as necessary
≡ ♡ S	<a href="#">PeekT</a>	Retrieves the specified range of elements from the array
≡ ♡ S	<a href="#">PeekEndT</a>	Retrieves the specified number of elements from the end of the array

---

≡ S	<a href="#">PeekStartT</a>	Retrieves the specified number of elements from the start of the array without changing the source array
≡ S	<a href="#">TakeT</a>	Retrieves and removes the specified range of elements from the array
≡ S	<a href="#">TakeFromEndT</a>	Retrieves and removes the specified number of elements from the end of the array
≡ S	<a href="#">TakeFromStartT</a>	Retrieves and removes the specified number of elements from the start of the array
≡ S	<a href="#">TrimT</a>	Removes the specified range of elements from the array, resizing the array as necessary
≡ S	<a href="#">TrimEndT</a>	Removes the specified number of elements from the end of the array, resizing the array as necessary
≡ S	<a href="#">TrimStartT</a>	Removes the specified number of elements from the start of the array, resizing the array as necessary

---

[Top](#)

## See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethods Methods

The [ArrayMethods](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡  	<a href="#">AppendT</a>	Appends the items to an array, resizing the array as necessary
≡  	<a href="#">InsertT</a>	Appends the items to an array, resizing the array as necessary
≡  	<a href="#">PeekT</a>	Retrieves the specified range of elements from the array
≡  	<a href="#">PeekEndT</a>	Retrieves the specified number of elements from the end of the array
≡  	<a href="#">PeekStartT</a>	Retrieves the specified number of elements from the start of the array without changing the source array
≡  	<a href="#">TakeT</a>	Retrieves and removes the specified range of elements from the array
≡  	<a href="#">TakeFromEndT</a>	Retrieves and removes the specified number of elements from the end of the array
≡  	<a href="#">TakeFromStartT</a>	Retrieves and removes the

specified number of elements from the start of the array



### [TrimT](#)

Removes the specified range of elements from the array, resizing the array as necessary



### [TrimEndT](#)

Removes the specified number of elements from the end of the array, resizing the array as necessary



### [TrimStartT](#)

Removes the specified number of elements from the start of the array, resizing the array as necessary

[Top](#)

## ► See Also

Reference

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethodsAppendT Method

Appends the items to an array, resizing the array as necessary

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public static T[] Append<T>(  
    ref T[] source,  
    T[] itemsToAdd  
)
```

[Copy](#)

### Parameters

*source*

Type: [T](#)

The source array

*itemsToAdd*

Type: [T](#)

The array of items to append to the source

### Type Parameters

*T*

The data type

### Return Value

Type: [T](#)

The modified source array

## ◀ See Also

Reference

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethodsInsertT Method

Appends the items to an array, resizing the array as necessary

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public static T[] Insert<T>(  
    ref T[] source,  
    T[] itemsToInsert,  
    int index  
)
```

[Copy](#)

## Parameters

*source*

Type: [T](#)

The source array

*itemsToInsert*

Type: [T](#)

The array of items to append to the source

*index*

Type: [SystemInt32](#)

The index where the items should be inserted

## Type Parameters

*T*

The data type

## Return Value

Type: *T*

The modified source array

## ▲ See Also

Reference

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethods.Peek<T> Method

Retrieves the specified range of elements from the array

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public static T[] Peek<T>(  
    T[] source,  
    int startIndex,  
    int length  
)
```

[Copy](#)

## Parameters

*source*

Type: [T](#)

The source array

*startIndex*

Type: [SystemInt32](#)

The index from which to start retrieving elements

*length*

Type: [SystemInt32](#)

The number of elements to retrieve

## Type Parameters

*T*

The data type

## Return Value

Type: *T*

A new array containing only the specified subset of elements

## ▲ See Also

Reference

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethods.PeekEnd<T> Method

Retrieves the specified number of elements from the end of the array

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static T[] PeekEnd<T>(
    T[] source,
    int length
)
```

## Parameters

*source*

Type: [T](#)

The source array

*length*

Type: [SystemInt32](#)

The number of elements to retrieve

## Type Parameters

*T*

The data type

## Return Value

Type: [T](#)

A new array containing only the specified subset of elements

## ◀ See Also

Reference

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethods.PeekStart $T$

## Method

Retrieves the specified number of elements from the start of the array without changing the source array

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

### ► Syntax

C#

[Copy](#)

```
public static T[] PeekStart<T>(  
    T[] source,  
    int length  
)
```

### Parameters

*source*

Type: [T](#)

The source array

*length*

Type: [SystemInt32](#)

The number of elements to retrieve

### Type Parameters

$T$

The data type

### Return Value

Type: [T](#)

A new array containing only the specified subset of elements

## See Also

[Reference](#)

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethodsTake $T$ Method

Retrieves and removes the specified range of elements from the array

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public static T[] Take<T>(
    ref T[] source,
    int startIndex,
    int length
)
```

## Parameters

*source*

Type:  $T$

The source array

*startIndex*

Type: [SystemInt32](#)

The index from which to start retrieving elements

*length*

Type: [SystemInt32](#)

The number of elements to retrieve

## Type Parameters

$T$

The data type

## Return Value

Type: *T*

A new array containing only the specified subset of elements

## ▲ See Also

Reference

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethodsTakeFromEnd $T$ Method

Retrieves and removes the specified number of elements from the end of the array

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static T[] TakeFromEnd<T>(  
    ref T[] source,  
    int length  
)
```

## Parameters

*source*

Type:  $T$

The source array

*length*

Type: [SystemInt32](#)

The number of elements to retrieve

## Type Parameters

$T$

The data type

## Return Value

Type:  $T$

A new array containing only the specified subset of elements

## See Also

[Reference](#)

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethodsTakeFromStart $T$ Method

Retrieves and removes the specified number of elements from the start of the array

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static T[] TakeFromStart<T>(
    ref T[] source,
    int length
)
```

## Parameters

*source*

Type: [T](#)

The source array

*length*

Type: [SystemInt32](#)

The number of elements to retrieve

## Type Parameters

*T*

The data type

## Return Value

Type: [T](#)

A new array containing only the specified subset of elements

## See Also

[Reference](#)

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethodsTrim $T$ Method

Removes the specified range of elements from the array, resizing the array as necessary

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static  $T[]$  Trim< $T$ >(  
    ref  $T[]$  source,  
    int startIndex,  
    int length  
)
```

## Parameters

*source*

Type:  $T$

The source array

*startIndex*

Type: [SystemInt32](#)

The index from which to start removing elements

*length*

Type: [SystemInt32](#)

The number of elements to remove

## Type Parameters

$T$

The data type

Return Value

Type: *T*

The modified source array

## See Also

Reference

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethodsTrimEnd<T> Method

Removes the specified number of elements from the end of the array, resizing the array as necessary

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static T[] TrimEnd<T>(
    ref T[] source,
    int length
)
```

## Parameters

*source*

Type: [T](#)

The source array

*length*

Type: [SystemInt32](#)

The number of elements to remove

## Type Parameters

*T*

The data type

## Return Value

Type: [T](#)

The modified source array

## ◀ See Also

Reference

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ArrayMethodsTrimStart $T$ Method

Removes the specified number of elements from the start of the array, resizing the array as necessary

**Namespace:** [W](#)

**Assembly:** Tungsten.ArrayMethods (in Tungsten.ArrayMethods.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static T[] TrimStart<T>(  
    ref T[] source,  
    int length  
)
```

## Parameters

*source*

Type:  $T$

The source array

*length*

Type: [SystemInt32](#)

The number of elements to remove

## Type Parameters

$T$

The data type

## Return Value

Type:  $T$

The modified source array

## ◀ See Also

Reference

[ArrayMethods Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensions Class

Extensions which convert objects of one type to another

## ▪ Inheritance Hierarchy

[SystemObject](#) [WAsExtensions](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ▪ Syntax

C#

[Copy](#)

```
public static class AsExtensions
```

The [AsExtensions](#) type exposes the following members.

## ▪ Methods

Name	Description
   <a href="#">AsTType</a>	Use Generic syntax for the as operator.
  <a href="#">AsBase64(Byte)</a>	Converts a byte array to a Base64 encoded string
  <a href="#">AsBase64(String)</a>	Converts a string to Base64 encoding
  <a href="#">AsBase64(String, Encoding)</a>	Converts a string to Base64 encoding
  <a href="#">AsBytes(String)</a>	Converts a string to an encoded byte array

≡  	<a href="#">AsBytes(String, Encoding)</a>	Converts a string to an encoded byte array
≡  	<a href="#">AsCompressed</a>	Compresses the byte array using System.IO.Compression.DeflateStream
≡  	<a href="#">AsStream(Byte)</a>	Creates a MemoryStream object and initializes it with the specified byte array
≡  	<a href="#">AsStream(String)</a>	Creates a MemoryStream object and initializes it with the specified string
≡  	<a href="#">AsString(Byte)</a>	Converts an encoded byte array to a string
≡  	<a href="#">AsString(Byte, Encoding)</a>	Converts an encoded byte array to a string
≡  	<a href="#">AsString(Byte, Int32, Int32)</a>	Converts an encoded byte array to a string
≡  	<a href="#">AsString(Byte, Int32, Int32, Encoding)</a>	Converts an encoded byte array to a string

[Top](#)

## ▲ See Also

[Reference](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# AsExtensions Methods

The [AsExtensions](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator.
<a href="#">AsBase64(Byte)</a>	Converts a byte array to a Base64 encoded string
<a href="#">AsBase64(String)</a>	Converts a string to Base64 encoding
<a href="#">AsBase64(String, Encoding)</a>	Converts a string to Base64 encoding
<a href="#">AsBytes(String)</a>	Converts a string to an encoded byte array
<a href="#">AsBytes(String, Encoding)</a>	Converts a string to an encoded byte array
<a href="#">AsCompressed</a>	Compresses the byte array using System.IO.Compression.DeflateStream
<a href="#">AsStream(Byte)</a>	Creates a MemoryStream object and initializes it with the specified byte array
<a href="#">AsStream(String)</a>	Creates a MemoryStream object and initializes it with the specified string
<a href="#">AsString(Byte)</a>	Converts an encoded byte array to a string

≡  	<a href="#">AsString(Byte, Encoding)</a>	Converts an encoded byte array to a string
≡  	<a href="#">AsString(Byte, Int32, Int32)</a>	Converts an encoded byte array to a string
≡  	<a href="#">AsString(Byte, Int32, Int32, Encoding)</a>	Converts an encoded byte array to a string

---

[Top](#)

## ▲ See Also

Reference

[AsExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsTType Method

Use Generic syntax for the as operator.

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static TType As<TType>(  
    this Object this  
)  
where TType : class
```

[Copy](#)

### Parameters

*this*

Type: [SystemObject](#)

The item to convert to type TType

### Type Parameters

*TType*

The type to convert the item reference to.

### Return Value

Type: [TType](#)

Null if @this cannot be referenced as TType. Otherwise, the item as TType

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance method syntax to call this method, omit the first parameter. For more

information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► Examples

```
expression as type
```

[Copy](#)

becomes

```
expression.As<type>()
```

[Copy](#)

## ► See Also

[Reference](#)

[AsExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsBase64 Method

## ▪ Overload List

	Name	Description
≡ 	<a href="#">AsBase64(Byte)</a>	Converts a byte array to a Base64 encoded string
≡ 	<a href="#">AsBase64(String)</a>	Converts a string to Base64 encoding
≡ 	<a href="#">AsBase64(String, Encoding)</a>	Converts a string to Base64 encoding

[Top](#)

## ▪ See Also

[Reference](#)

[AsExtensions Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# AsExtensionsAsBase64 Method (Byte)

Converts a byte array to a Base64 encoded string

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static string AsBase64(  
    this byte[] this  
)
```

### Parameters

*this*

Type: [SystemByte](#)

The string to convert to Base64 encoding

### Return Value

Type: [String](#)

The Base64 encoded string

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

## Reference

[AsExtensions Class](#)  
[AsBase64 Overload](#)  
[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsBase64 Method (String)

Converts a string to Base64 encoding

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static string AsBase64(  
    this string this  
)
```

## Parameters

*this*

Type: [System.String](#)

The string to convert to Base64 encoding

## Return Value

Type: [String](#)

The Base64 encoded string

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [String](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

## Reference

[AsExtensions Class](#)  
[AsBase64 Overload](#)  
[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsBase64 Method (String, Encoding)

Converts a string to Base64 encoding

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static string AsBase64(  
    this string this,  
    Encoding encoding  
)
```

## Parameters

*this*

Type: [System.String](#)

The string to convert to Base64 encoding

*encoding*

Type: [System.TextEncoding](#)

The encoding to use

## Return Value

Type: [String](#)

The Base64 encoded string

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [String](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension](#)

[Methods \(C# Programming Guide\)](#).

## See Also

[Reference](#)

[AsExtensions Class](#)

[AsBase64 Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsBytes Method

## ▪ Overload List

	Name	Description
≡  	<a href="#">AsBytes(String)</a>	Converts a string to an encoded byte array
≡  	<a href="#">AsBytes(String, Encoding)</a>	Converts a string to an encoded byte array

[Top](#)

## ▪ See Also

[Reference](#)

[AsExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsBytes Method (String)

Converts a string to an encoded byte array

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public static byte[] AsBytes(  
    this string this  
)
```

## Parameters

*this*

Type: [System.String](#)

The string to convert to an encoded byte array

## Return Value

Type: [Byte](#)

A byte array encoding of the specified string

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [String](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

## Reference

[AsExtensions Class](#)

[AsBytes Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsBytes Method (String, Encoding)

Converts a string to an encoded byte array

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public static byte[] AsBytes(  
    this string this,  
    Encoding encoding  
)
```

## Parameters

*this*

Type: [System.String](#)

The string to convert to an encoded byte array

*encoding*

Type: [System.TextEncoding](#)

The encoding to use

## Return Value

Type: [Byte](#)

A byte array encoding of the specified string

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [String](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension](#)

[Methods \(C# Programming Guide\)](#).

## See Also

[Reference](#)

[AsExtensions Class](#)

[AsBytes Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsCompressed Method

Compresses the byte array using  
System.IO.Compression.DeflateStream

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static byte[] AsCompressed(  
    this byte[] bytes  
)
```

### Parameters

*bytes*

Type: [SystemByte](#)

The byte array to compress

### Return Value

Type: [Byte](#)

A byte array of compressed data

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

Reference

[AsExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsStream Method

## ▪ Overload List

	Name	Description
 <b>S</b>	<a href="#">AsStream(Byte)</a>	Creates a MemoryStream object and initializes it with the specified byte array
 <b>S</b>	<a href="#">AsStream(String)</a>	Creates a MemoryStream object and initializes it with the specified string

[Top](#)

## ▪ See Also

Reference

[AsExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsStream Method (Byte)

Creates a MemoryStream object and initializes it with the specified byte array

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static MemoryStream AsStream(  
    this byte[] this  
)
```

## Parameters

*this*

Type: [SystemByte](#)

The byte array used in creating the MemoryStream

## Return Value

Type: [MemoryStream](#)

A new MemoryStream initialized with the specified byte array

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

## Reference

[AsExtensions Class](#)  
[AsStream Overload](#)  
[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsStream Method (String)

Creates a MemoryStream object and initializes it with the specified string

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public static MemoryStream AsStream(  
    this string this  
)
```

## Parameters

*this*

Type: [System.String](#)

The string used in creating the MemoryStream

## Return Value

Type: [MemoryStream](#)

A new MemoryStream initialized with the specified string

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [String](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

## Reference

[AsExtensions Class](#)  
[AsStream Overload](#)  
[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsString Method

## ▪ Overload List

	Name	Description
≡  	<a href="#">AsString(Byte)</a>	Converts an encoded byte array to a string
≡  	<a href="#">AsString(Byte, Encoding)</a>	Converts an encoded byte array to a string
≡  	<a href="#">AsString(Byte, Int32, Int32)</a>	Converts an encoded byte array to a string
≡  	<a href="#">AsString(Byte, Int32, Int32, Encoding)</a>	Converts an encoded byte array to a string

[Top](#)

## ▪ See Also

[Reference](#)

[AsExtensions Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# AsExtensionsAsString Method (Byte)

Converts an encoded byte array to a string

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static string AsString(  
    this byte[] this  
)
```

## Parameters

*this*

Type: [SystemByte](#)

The encoded byte array to conver to a string

## Return Value

Type: [String](#)

The string representation of the encoded byte array

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

## Reference

[AsExtensions Class](#)

[AsString Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsString Method (Byte, Encoding)

Converts an encoded byte array to a string

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public static string AsString(  
    this byte[] this,  
    Encoding encoding  
)
```

## Parameters

*this*

Type: [SystemByte](#)

The encoded byte array to conver to a string

*encoding*

Type: [System.TextEncoding](#)

The encoding to use

## Return Value

Type: [String](#)

The string representation of the encoded byte array

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension](#)

[Methods \(C# Programming Guide\)](#).

## See Also

[Reference](#)

[AsExtensions Class](#)

[AsString Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsString Method (Byte, Int32, Int32)

Converts an encoded byte array to a string

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static string AsString(  
    this byte[] this,  
    int index,  
    int count  
)
```

## Parameters

*this*

Type: [SystemByte](#)

The encoded byte array to conver to a string

*index*

Type: [SystemInt32](#)

The starting index

*count*

Type: [SystemInt32](#)

The number of bytes to convert

## Return Value

Type: [String](#)

The string representation of the encoded byte array

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

- [AsExtensions Class](#)
- [AsString Overload](#)
- [W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AsExtensionsAsString Method (Byte, Int32, Int32, Encoding)

Converts an encoded byte array to a string

**Namespace:** [W](#)

**Assembly:** Tungsten.As (in Tungsten.As.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static string AsString(  
    this byte[] this,  
    int index,  
    int count,  
    Encoding encoding  
)
```

[Copy](#)

## Parameters

*this*

Type: [SystemByte](#)

The encoded byte array to conver to a string

*index*

Type: [SystemInt32](#)

The starting index

*count*

Type: [SystemInt32](#)

The number of bytes to convert

*encoding*

Type: [System.TextEncoding](#)

The encoding to use

## Return Value

Type: [String](#)

The string representation of the encoded byte array

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[AsExtensions Class](#)

[AsString Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResult Class

A non-generic return value for a function. CallResult encapsulates a success/failure and an exception.

## ► Inheritance Hierarchy

[SystemObject](#) [WCallResult](#)  
[WCallResultTResult](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public class CallResult
```

The [CallResult](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">CallResult</a>	Default constructor, initializes Success to false
≡	<a href="#">CallResult(Boolean)</a>	Constructor which accepts an initial value for Success
≡	<a href="#">CallResult(Boolean, Exception)</a>	Constructor which accepts an initial value for Success and an initial value for

## Exception

[Top](#)

### Properties

	Name	Description
	<a href="#">Empty</a>	Provides a new instance of an uninitialized CallResult
	<a href="#">Exception</a>	Provide exception data to the caller if desired
	<a href="#">Success</a>	Set to True if the function succeeds, otherwise False

[Top](#)

### Methods

	Name	Description
	<a href="#">AsTuple</a>	Creates a Tuple from the properties
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)

	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
🔑	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Extension Methods

Name	Description
≡ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
≡ <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResult Constructor

## ▪ Overload List

	Name	Description
≡	<a href="#">CallResult</a>	Default constructor, initializes Success to false
≡	<a href="#">CallResult(Boolean)</a>	Constructor which accepts an initial value for Success
≡	<a href="#">CallResult(Boolean, Exception)</a>	Constructor which accepts an initial value for Success and an initial value for Exception

[Top](#)

## ▪ See Also

Reference

[CallResult Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# CallResult Constructor

Default constructor, initializes Success to false

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public CallResult()
```

## ► See Also

Reference

[CallResult Class](#)

[CallResult Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResult Constructor (Boolean)

Constructor which accepts an initial value for Success

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

```
C#  
public CallResult(  
    bool success  
)
```

[Copy](#)

### Parameters

*success*

Type: [SystemBoolean](#)

The initial value for Success

## ► See Also

Reference

[CallResult Class](#)

[CallResult Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# CallResult Constructor (Boolean, Exception)

Constructor which accepts an initial value for Success and an initial value for Exception

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

```
C#  
public CallResult(  
    bool success,  
    Exception e  
)
```

[Copy](#)

## Parameters

*success*

Type: [SystemBoolean](#)

The initial value for Success

*e*

Type: [SystemException](#)

The initial value for Exception

## ► See Also

[Reference](#)

[CallResult Class](#)

[CallResult Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResult Properties

The [CallResult](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Empty</a>	Provides a new instance of an uninitialized CallResult
	<a href="#">Exception</a>	Provide exception data to the caller if desired
	<a href="#">Success</a>	Set to True if the function succeeds, otherwise False

[Top](#)

## ► See Also

[Reference](#)

[CallResult Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# CallResultEmpty Property

Provides a new instance of an uninitialized CallResult

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public static CallResult Empty { get; }
```

Property Value

Type: [CallResult](#)

## ► See Also

[Reference](#)

[CallResult Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResultException Property

Provide exception data to the caller if desired

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public Exception Exception { get; set; }
```

Property Value

Type: [Exception](#)

## ► See Also

[Reference](#)

[CallResult Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResultSuccess Property

Set to True if the function succeeds, otherwise False

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

C#

Copy

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

Property Value

Type: [Boolean](#)

## ► See Also

[Reference](#)

[CallResult Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResult Methods

The [CallResult](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">AsTuple</a>	Creates a Tuple from the properties
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
--------------------------	---

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

[Top](#)

## ▲ See Also

Reference

[CallResult Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResultAsTuple Method

Creates a Tuple from the properties

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public Tuple<bool, Exception> AsTuple()
```

Return Value

Type: [TupleBoolean, Exception](#)

A Tuple containing the property values

## ► Remarks

This is essentially a hard-coded Tuple; reflection is not used to obtain the properties. As a result, properties of child classes will not be in the returned Tuple.

## ► See Also

Reference

[CallResult Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# CallResult<TResult> Class

Generic class to be used as a return value. CallResult encapsulates a success/failure, an exception and a return value.

## ▪ Inheritance Hierarchy

```
SystemObject WCallResult  
WCallResultTResult
```

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ▪ Syntax

```
C#  
public class CallResult<TResult> : CallResult  
Copy
```

### Type Parameters

*TResult*

The type to be used for the Result member (the return value of the function)

The [CallResultTResult](#) type exposes the following members.

## ▪ Constructors

	Name	Description
≡	<a href="#">CallResultTResult</a>	Default constructor
≡	<a href="#">CallResultTResult(Boolean)</a>	Constructor

---

		accepting an initial Success value
≡	<a href="#">CallResultTResult(Boolean, TResult)</a>	Constructor accepting an initial Success value and an initial Result value
≡	<a href="#">CallResultTResult(Boolean, TResult, Exception)</a>	Constructor accepting an initial Success value, an initial Result value and an initial Exception value

---

[Top](#)

## Properties

	Name	Description
 	<a href="#">Empty</a>	Provides a new instance of an uninitialized CallResult<TResult>
	<a href="#">Exception</a>	Provide exception data to the caller if desired (Inherited from <a href="#">CallResult</a> .)
	<a href="#">Result</a>	The return value
	<a href="#">Success</a>	Set to True if the function succeeds, otherwise False (Inherited from <a href="#">CallResult</a> .)

---

[Top](#)

## ◀ Methods

	Name	Description
≡💡	<a href="#">AsTuple</a>	Creates a Tuple from the properties (Inherited from <a href="#">CallResult</a> .)
≡💡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by

[MonitorExtensions.](#))

	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

[Top](#)

## ▲ See Also

[Reference](#)

[W Namespace](#)



## Tungsten

W<sup>74</sup>

# CallResultTResult Constructor

## ► Overload List

	Name	Description
≡	<a href="#">CallResultTResult</a>	Default constructor
≡	<a href="#">CallResultTResult(Boolean)</a>	Constructor accepting an initial Success value
≡	<a href="#">CallResultTResult(Boolean, TResult)</a>	Constructor accepting an initial Success value and an initial Result value
≡	<a href="#">CallResultTResult(Boolean, TResult, Exception)</a>	Constructor accepting an initial Success value, an initial Result value and an initial Exception value

[Top](#)

## ► See Also

[Reference](#)

[CallResultTResult Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# CallResultTResult Constructor

Default constructor

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public CallResult()
```

## ► See Also

Reference

[CallResultTResult Class](#)

[CallResultTResult Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResultTResult Constructor (Boolean)

Constructor accepting an initial Success value

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

```
C#  
public CallResult(  
    bool success  
)
```

[Copy](#)

## Parameters

*success*

Type: [SystemBoolean](#)

The initial Success value

## ► See Also

Reference

[CallResultTResult Class](#)

[CallResultTResult Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# CallResult<TResult> Constructor (Boolean, TResult)

Constructor accepting an initial Success value and an initial Result value

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public CallResult(  
    bool success,  
    TResult result  
)
```

## Parameters

*success*

Type: [System.Boolean](#)

The initial Success value

*result*

Type: [TResult](#)

The initial Result value

## ► See Also

[Reference](#)

[CallResult<TResult> Class](#)

[CallResult<TResult> Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResult<*TResult*> Constructor (Boolean, *TResult*, Exception)

Constructor accepting an initial Success value, an initial Result value and an initial Exception value

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public CallResult(  
    bool success,  
    TResult result,  
    Exception e  
)
```

## Parameters

*success*

Type: [SystemBoolean](#)

The initial value for Success

*result*

Type: [TResult](#)

The initial Result value

*e*

Type: [SystemException](#)

An exception object, if an exception occurred

## ► See Also

Reference

[CallResultTResult Class](#)

[CallResultTResult Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResultTResult Properties

The [CallResultTResult](#) generic type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Empty</a>	Provides a new instance of an uninitialized CallResult<TResult>
	<a href="#">Exception</a>	Provide exception data to the caller if desired (Inherited from <a href="#">CallResult</a> .)
	<a href="#">Result</a>	The return value
	<a href="#">Success</a>	Set to True if the function succeeds, otherwise False (Inherited from <a href="#">CallResult</a> .)

[Top](#)

## ► See Also

[Reference](#)

[CallResultTResult Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# CallResult<TResult>.Empty Property

Provides a new instance of an uninitialized CallResult<TResult>

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

```
C#  
public static CallResult<TResult> Empty { get; }
```

[Copy](#)

Property Value

Type: [CallResult<TResult>](#)

## ► See Also

Reference

[CallResult<TResult> Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResult{TResult}Result Property

The return value

**Namespace:** [W](#)

**Assembly:** Tungsten.CallResult (in Tungsten.CallResult.dll) Version: 2.0.0

## ► Syntax

```
C#  
public TResult Result { get; set; }
```

[Copy](#)

Property Value

Type: [TResult](#)

## ► See Also

Reference

[CallResult{TResult} Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CallResultTResult Methods

The [CallResultTResult](#) generic type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">AsTuple</a>	Creates a Tuple from the properties (Inherited from <a href="#">CallResult</a> .)
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)



## ToString

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

# Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

[Top](#)

## ▲ See Also

Reference

[CallResultTResult Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ConsoleStringExtensions Class

Extension methods related to the Console

## ► Inheritance Hierarchy

[SystemObject](#) [WConsoleStringExtensions](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.Console (in Tungsten.Console.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public static class ConsoleStringExtensions
```

The [ConsoleStringExtensions](#) type exposes the following members.

## ► Methods

	Name	Description
 	<a href="#">WriteFullConsoleLine</a>	Writes text to the console. Columns which the text doesn't overwrite are filled with the specified padding character.
 	<a href="#">WriteToConsole</a>	Writes text to the console at the specified location (x,y)

[Top](#)

## ▲ See Also

Reference

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ConsoleStringExtensions Methods

The [ConsoleStringExtensions](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">WriteFullConsoleLine</a>	Writes text to the console. Columns which the text doesn't overwrite are filled with the specified padding character.
 <a href="#">WriteToConsole</a>	Writes text to the console at the specified location (x,y)

[Top](#)

## ▪ See Also

[Reference](#)

[ConsoleStringExtensions Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# ConsoleStringExtensionsWriteFullConsoleLine Method

Writes text to the console. Columns which the text doesn't overwrite are filled with the specified padding character.

**Namespace:** [W](#)

**Assembly:** Tungsten.Console (in Tungsten.Console.dll) Version: 2.0.0

## ► Syntax

```
C#  
public static void WriteFullConsoleLine(  
    this string message,  
    int verticalOffset = -1,  
    char paddingChar = ''  
)
```

[Copy](#)

## Parameters

*message*

Type: [System.String](#)

The text to write to the console

*verticalOffset* **(Optional)**

Type: [System.Int32](#)

The line (or row) on which to write the text. 0 is the top of the Console.

*paddingChar* **(Optional)**

Type: [System.Char](#)

The padding character used to fill the unused portion of the line

## Usage Note

In Visual Basic and C#, you can call this method as an instance

method on any object of type [String](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

[Reference](#)

[ConsoleStringExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ConsoleStringExtensionsWriteToCol Method

Writes text to the console at the specified location (x,y)

**Namespace:** [W](#)

**Assembly:** Tungsten.Console (in Tungsten.Console.dll) Version: 2.0.0

## ► Syntax

```
C#  
public static void WriteToConsole(  
    this string message,  
    int x,  
    int y  
)
```

[Copy](#)

## Parameters

*message*

Type: [SystemString](#)

The text to write to the console

*x*

Type: [SystemInt32](#)

The column on which to start writing text

*y*

Type: [SystemInt32](#)

The row on which to start writing text

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [String](#). When you use instance method syntax to call this method, omit the first parameter. For more

information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► Remarks

The Console origin (0,0) is top-left

## ► See Also

Reference

[ConsoleStringExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender Class

Wraps the functionality of delegate, event and RaiseXXX into a single class

## ► Inheritance Hierarchy

[SystemObject](#) [WEventTemplateTSender](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public class EventTemplate<TSender>
```

Type Parameters

*TSender*

The object raising the event

The [EventTemplateTSender](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">EventTemplateTSender</a>	Initializes a new instance of the <a href="#">EventTemplateTSender</a> class

[Top](#)

## ◀ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
 <a href="#">Raise</a>	Raises the template event
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Events

Name	Description
 <a href="#">OnRaised</a>	The template event

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ◀ See Also

Reference

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender Constructor

Initializes a new instance of the [EventTemplateTSender](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public EventTemplate()
```

## ► See Also

[Reference](#)

[EventTemplateTSender Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender Methods

The [EventTemplateTSender](#) generic type exposes the following members.

## ▪ Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">Raise</a>	Raises the template event

<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
--------------------------	---

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

[Top](#)

## ▲ See Also

Reference

[EventTemplateTSender Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSenderRaise Method

Raises the template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public void Raise(  
    TSender sender  
)
```

[Copy](#)

## Parameters

*sender*

Type: [TSender](#)

The object which raised this event

## ► See Also

### Reference

[EventTemplateTSender Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# EventTemplateTSender Events

The [EventTemplateTSender](#) generic type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">OnRaised</a>	The template event

[Top](#)

## ▪ See Also

Reference

[EventTemplateTSender Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSenderOnRaisedEvent

The template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public event Action<TSender> OnRaised
```

Value

Type: [SystemAction<TSender>](#)

## ► See Also

Reference

[EventTemplateTSender Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplate $<TSender, Arg1>$

## Class

Wraps the functionality of delegate, event and RaiseXXX into a single class

### ► Inheritance Hierarchy

[SystemObject](#) [WEventTemplate \$<TSender, Arg1>\$](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

### ► Syntax

C#

Copy

```
public class EventTemplate<TSender, Arg1>
```

#### Type Parameters

*TSender*

The object raising the event

*Arg1*

The first argument

The [EventTemplate \$<TSender, Arg1>\$](#)  type exposes the following members.

### ► Constructors

	Name	Description
≡	<a href="#">EventTemplate<math>&lt;TSender, Arg1&gt;</math></a>	Initializes a new

**Arg1** instance of the [EventTemplateTSender](#), **Arg1** class

[Top](#)

## Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">Raise</a>	Raises the template event
	<a href="#">ToString</a>	Returns a string that represents the current object.

(Inherited from [Object](#).)

[Top](#)

## Events

Name	Description
 <a href="#">OnRaised</a>	The template event

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ◀ See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1 Constructor

Initializes a new instance of the [EventTemplateTSender, Arg1](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public EventTemplate()
```

## ► See Also

Reference

[EventTemplateTSender, Arg1 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1 Methods

The [EventTemplateTSender, Arg1](#) generic type exposes the following members.

## Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

	Raise	Raises the template event
≡	ToString	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
≡ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
≡ <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
≡ <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
≡ <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by

[MonitorExtensions. \)](#)

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

Reference

[EventTemplateTSender, Arg1 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1Raise Method

Raises the template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public void Raise(  
    TSender sender,  
    Arg1 arg1  
)
```

### Parameters

*sender*

Type: [TSender](#)

The object which raised this event

*arg1*

Type: [Arg1](#)

The first argument

## ► See Also

[Reference](#)

[EventTemplateTSender, Arg1 Class](#)

[W Namespace](#)



## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1 Events

The [EventTemplateTSender, Arg1](#) generic type exposes the following members.

## Events

	Name	Description
	<a href="#">OnRaised</a>	The template event

[Top](#)

## See Also

[Reference](#)

[EventTemplateTSender, Arg1 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplate $T$ Sender, Arg1OnRaised Event

The template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public event Action<TSender, Arg1> OnRaised
```

Value

Type: [SystemAction \$T\$ Sender, Arg1](#)

## ► See Also

Reference

[EventTemplate \$T\$ Sender, Arg1 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplate $<TSender, Arg1, Arg2>$ Class

Wraps the functionality of delegate, event and RaiseXXX into a single class

## ► Inheritance Hierarchy

[SystemObject](#) [WEventTemplate \$<TSender, Arg1, Arg2>\$](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public class EventTemplate<TSender, Arg1, Arg2>
```

### Type Parameters

*TSender*

The object raising the event

*Arg1*

The first argument

*Arg2*

The second argument

The [EventTemplate \$<TSender, Arg1, Arg2>\$](#)  type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">EventTemplateTSender, Arg1, Arg2</a>	Initializes a new instance of the <code>EventTemplateTSender, Arg1, Arg2</code> class

[Top](#)

## ◀ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
 <a href="#">Raise</a>	Raises the template event



## ToString

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## Events

	Name	Description
	<a href="#">OnRaised</a>	The template event

[Top](#)

## Extension Methods

	Name	Description
	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by

[MonitorExtensions.](#))

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )

## [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1, Arg2 Constructor

Initializes a new instance of the [EventTemplateTSender, Arg1, Arg2](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public EventTemplate()
```

## ► See Also

Reference

[EventTemplateTSender, Arg1, Arg2 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1, Arg2 Methods

The [EventTemplateTSender, Arg1, Arg2](#) generic type exposes the following members.

## Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

	Raise	Raises the template event
≡	ToString	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
≡ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
≡ <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
≡ <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
≡ <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets it's IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ◀ See Also

Reference

[EventTemplateTSender, Arg1, Arg2 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplate $T$ Sender, Arg1, Arg2Raise Method

Raises the template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public void Raise(
    TSender sender,
    Arg1 arg1,
    Arg2 arg2
)
```

### Parameters

*sender*

Type: [TSender](#)

The object which raised this event

*arg1*

Type: [Arg1](#)

The first argument

*arg2*

Type: [Arg2](#)

The second argument

## ► See Also

Reference

EventTemplateTSender, Arg1, Arg2 Class  
W Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1, Arg2 Events

The [EventTemplateTSender, Arg1, Arg2](#) generic type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">OnRaised</a>	The template event

[Top](#)

## ▪ See Also

[Reference](#)

[EventTemplateTSender, Arg1, Arg2 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1, Arg2OnRaised Event

The template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public event Action<TSender, Arg1, Arg2> OnRaised
```

Value

Type: [SystemActionTSender, Arg1, Arg2](#)

## ► See Also

Reference

[EventTemplateTSender, Arg1, Arg2 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplate $<TSender, Arg1, Arg2, Arg3>$ Class

Wraps the functionality of delegate, event and RaiseXXX into a single class

## ► Inheritance Hierarchy

[SystemObject](#) [WEventTemplate \$<TSender, Arg1, Arg2, Arg3>\$](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public class EventTemplate<TSender, Arg1, Arg2, /
```

### Type Parameters

*TSender*

The object raising the event

*Arg1*

The first argument

*Arg2*

The second argument

*Arg3*

The third argument

The [EventTemplate \$<TSender, Arg1, Arg2, Arg3>\$](#)  type exposes the following members.

## ► Constructors

	Name	Description
≡♪	<a href="#">EventTemplateTSender, Arg1, Arg2, Arg3</a>	Initializes a new instance of the <a href="#">EventTemplateTSender, Arg1, Arg2, Arg3</a> class

[Top](#)

## ► Methods

	Name	Description
≡♪	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡♪	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡♪	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> .

(Inherited from [Object](#).)

 <a href="#">Raise</a>	Raises the template event
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Events

Name	Description
 <a href="#">OnRaised</a>	The template event

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1, Arg2, Arg3 Constructor

Initializes a new instance of the [EventTemplateTSender, Arg1, Arg2, Arg3](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

```
public EventTemplate()
```

[Copy](#)

## ► See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1, Arg2, Arg3 Methods

The [EventTemplateTSender, Arg1, Arg2, Arg3](#) generic type exposes the following members.

## Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

	Raise	Raises the template event
≡	ToString	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
≡ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
≡ <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
≡ <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
≡ <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplate<TSender, Arg1, Arg2, Arg3>.Raise Method

Raises the template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public void Raise(  
    TSender sender,  
    Arg1 arg1,  
    Arg2 arg2,  
    Arg3 arg3  
)
```

[Copy](#)

## Parameters

*sender*

Type: [TSender](#)

The object which raised this event

*arg1*

Type: [Arg1](#)

The first argument

*arg2*

Type: [Arg2](#)

The second argument

*arg3*

Type: [Arg3](#)

The third argument

## ▲ See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# `EventTemplateTSender, Arg1, Arg2, Arg3` Events

The [EventTemplateTSender, Arg1, Arg2, Arg3](#) generic type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">OnRaised</a>	The template event

[Top](#)

## ▪ See Also

[Reference](#)

[EventTemplateTSender, Arg1, Arg2, Arg3 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1, Arg2, Arg3OnRaised Event

The template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public event Action<TSender, Arg1, Arg2, Arg3> Or
```

Value

Type: [SystemActionTSender, Arg1, Arg2, Arg3](#)

## ► See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *EventTemplateTSender, Arg1, Arg2, Arg3, Arg4* Class

Wraps the functionality of delegate, event and RaiseXXX into a single class

## ► Inheritance Hierarchy

[SystemObject](#) *WEventTemplateTSender, Arg1, Arg2, Arg3, Arg4*

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public class EventTemplate<TSender, Arg1, Arg2, />
```

### Type Parameters

*TSender*

The object raising the event

*Arg1*

The first argument

*Arg2*

The second argument

*Arg3*

The third argument

*Arg4*

The fourth argument

The [EventTemplateTSender, Arg1, Arg2, Arg3, Arg4](#) type exposes the

following members.

## ► Constructors

	Name	Description
≡	<a href="#">EventTemplateTSender, Arg1, Arg2, Arg3, Arg4</a>	Initializes a new instance of the <code>EventTemplateTSender, Arg1, Arg2, Arg3, Arg4</code> class

[Top](#)

## ► Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)



[MemberwiseClone](#) Creates a shallow copy of the current [Object](#).  
(Inherited from [Object](#).)



[Raise](#) Raises the template event



[ToString](#) Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## Events

	Name	Description
	<a href="#">OnRaised</a>	The template event

[Top](#)

## Extension Methods

	Name	Description
	<a href="#">AsTTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by

[PropertyHostExtensions.\)](#)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	Scans each field and

## [MarkAsClean](#)

property of type [IProperty](#)  
and sets it's `IsDirty` flag to  
`false`

(Defined by  
[PropertyHostExtensions](#).)



## [Unlock](#)

Performs a Monitor  
unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *EventTemplateTSender, Arg1, Arg2, Arg3, Arg4* Constructor

Initializes a new instance of the [EventTemplateTSender, Arg1, Arg2, Arg3, Arg4](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

```
public EventTemplate()
```

Copy

## ► See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *EventTemplateTSender, Arg1, Arg2, Arg3, Arg4* Methods

The [EventTemplateTSender, Arg1, Arg2, Arg3, Arg4](#) generic type exposes the following members.

## ▲ Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

	Raise	Raises the template event
≡	ToString	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
≡ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
≡ <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
≡ <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
≡ <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplate<TSender, Arg1, Arg2, Arg3, Arg4>.Raise Method

Raises the template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public void Raise(
    TSender sender,
    Arg1 arg1,
    Arg2 arg2,
    Arg3 arg3,
    Arg4 arg4
)
```

### Parameters

*sender*

Type: [TSender](#)

The object which raised this event

*arg1*

Type: [Arg1](#)

The first argument

*arg2*

Type: [Arg2](#)

The second argument

*arg3*

Type: [Arg3](#)

The third argument

*arg4*

Type: [Arg4](#)

The fourth argument

## ► See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *EventTemplateTSender, Arg1, Arg2, Arg3, Arg4 Events*

The [EventTemplateTSender, Arg1, Arg2, Arg3, Arg4](#) generic type exposes the following members.

## ► Events

	Name	Description
	<a href="#">OnRaised</a>	The template event

[Top](#)

## ► See Also

[Reference](#)

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1, Arg2, Arg3, Arg4OnRaised Event

The template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public event Action<TSender, Arg1, Arg2, Arg3, Ar
```

Value

Type: [SystemActionTSender, Arg1, Arg2, Arg3, Arg4](#)

## ► See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplate<TSender, Arg1, Arg2, Arg3, Arg4, Arg5 Class

Wraps the functionality of delegate, event and RaiseXXX into a single class

## ► Inheritance Hierarchy

[SystemObject](#) [WEventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)  
Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public class EventTemplate<TSender, Arg1, Arg2, /
```

### Type Parameters

*TSender*

The object raising the event

*Arg1*

The first argument

*Arg2*

The second argument

*Arg3*

The third argument

*Arg4*

The fourth argument

*Arg5*

## The fifth argument

The `EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5` type exposes the following members.

## ▪ Constructors

	Name	Description
≡♪	<a href="#">EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5</a>	Initializes a new instance of the <code>EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5</code> class

[Top](#)

## ▪ Methods

	Name	Description
≡♪	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code> .)
≡♪	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code> .)
≡♪	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <code>Object</code> .)
≡♪	<a href="#">GetType</a>	Gets the <code>Type</code> of the current

instance.

(Inherited from [Object](#).)



### [MemberwiseClone](#)

Creates a shallow copy of the current [Object](#).  
(Inherited from [Object](#).)



### [Raise](#)

Raises the template event



### [ToString](#)

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## Events

	Name	Description
	<a href="#">OnRaised</a>	The template event

[Top](#)

## Extension Methods

	Name	Description
	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has

IOwnedProperty  
members  
(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IP <i>roperty</i> (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock

(Defined by  
[MonitorExtensions](#).)

---



### [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets it's `IsDirty` flag to false

(Defined by  
[PropertyHostExtensions](#).)

---



### [Unlock](#)

Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# `EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5` Constructor

Initializes a new instance of the `EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5` class

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public EventTemplate()
```

## ► See Also

[Reference](#)

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5* Methods

The [EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5](#) generic type exposes the following members.

## ▲ Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

	Raise	Raises the template event
≡	ToString	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
≡ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
≡ <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
≡ <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
≡ <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets it's IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5 Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplate<TSender, Arg1, Arg2, Arg3, Arg4, Arg5>.Raise Method

Raises the template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public void Raise(
    TSender sender,
    Arg1 arg1,
    Arg2 arg2,
    Arg3 arg3,
    Arg4 arg4,
    Arg5 arg5
)
```

### Parameters

*sender*

Type: [TSender](#)

The object which raised this event

*arg1*

Type: [Arg1](#)

The first argument

*arg2*

Type: [Arg2](#)

The second argument

*arg3*

Type: [Arg3](#)

The third argument

*arg4*

Type: [Arg4](#)

The fourth argument

*arg5*

Type: [Arg5](#)

The fifth argument

## ◀ See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5 Class](#)  
[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5* Events

The [EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5](#) generic type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">OnRaised</a>	The template event

[Top](#)

## ▪ See Also

[Reference](#)

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5 Class](#)  
[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5OnRaised Event

The template event

**Namespace:** [W](#)

**Assembly:** Tungsten.EventTemplate (in Tungsten.EventTemplate.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public event Action<TSender, Arg1, Arg2, Arg3, Ar
```

Value

Type: [SystemActionTSender, Arg1, Arg2, Arg3, Arg4, Arg5](#)

## ► See Also

Reference

[EventTemplateTSender, Arg1, Arg2, Arg3, Arg4, Arg5 Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# FromExtensions Class

Extensions which convert objects of one type to another

## ► Inheritance Hierarchy

[SystemObject](#) [WFromExtensions](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.From (in Tungsten.From.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public static class FromExtensions
```

The [FromExtensions](#) type exposes the following members.

## ► Methods

Name	Description
  <a href="#">FromBase64(Byte)</a>	Converts a Base64 encoded byte array back to a normal byte array
  <a href="#">FromBase64(String)</a>	Converts a Base64 encoded string back to a normal string
  <a href="#">FromBase64(Byte, Encoding)</a>	Converts a Base64 encoded byte array back to a normal byte array
  <a href="#">FromBase64(String, Encoding)</a>	Converts a Base64 encoded string back to a normal string
  <a href="#">FromCompressed</a>	Decompresses the byte array using

[Top](#)

## ◀ See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# FromExtensions Methods

The [FromExtensions](#) type exposes the following members.

## ▪ Methods

Name	Description
  <a href="#">FromBase64(Byte)</a>	Converts a Base64 encoded byte array back to a normal byte array
  <a href="#">FromBase64(String)</a>	Converts a Base64 encoded string back to a normal string
  <a href="#">FromBase64(Byte, Encoding)</a>	Converts a Base64 encoded byte array back to a normal byte array
  <a href="#">FromBase64(String, Encoding)</a>	Converts a Base64 encoded string back to a normal string
  <a href="#">FromCompressed</a>	Decompresses the byte array using System.IO.Compression.DeflateStream

[Top](#)

## ▪ See Also

[Reference](#)

[FromExtensions Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# FromExtensionsFromBase64 Method

## Overload List

	Name	Description
≡  	<a href="#">FromBase64(Byte)</a>	Converts a Base64 encoded byte array back to a normal byte array
≡  	<a href="#">FromBase64(String)</a>	Converts a Base64 encoded string back to a normal string
≡  	<a href="#">FromBase64(Byte, Encoding)</a>	Converts a Base64 encoded byte array back to a normal byte array
≡  	<a href="#">FromBase64(String, Encoding)</a>	Converts a Base64 encoded string back to a normal string

[Top](#)

## See Also

[Reference](#)

[FromExtensions Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# FromExtensionsFromBase64 Method (Byte)

Converts a Base64 encoded byte array back to a normal byte array

**Namespace:** [W](#)

**Assembly:** Tungsten.From (in Tungsten.From.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public static string FromBase64(  
    this byte[] this  
)
```

## Parameters

*this*

Type: [SystemByte](#)

The Base64 encoded byte array to convert

## Return Value

Type: [String](#)

A non-encoded string

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

**Reference**

[FromExtensions Class](#)

[FromBase64 Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# FromExtensionsFromBase64 Method (String)

Converts a Base64 encoded string back to a normal string

**Namespace:** [W](#)

**Assembly:** Tungsten.From (in Tungsten.From.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static string FromBase64(  
    this string this  
)
```

[Copy](#)

## Parameters

*this*

Type: [System.String](#)

The Base64 encoded string to convert

## Return Value

Type: [String](#)

A non-encoded string

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [String](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

**Reference**

[FromExtensions Class](#)

[FromBase64 Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# FromExtensionsFromBase64 Method (Byte, Encoding)

Converts a Base64 encoded byte array back to a normal byte array

**Namespace:** [W](#)

**Assembly:** Tungsten.From (in Tungsten.From.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public static string FromBase64(
    this byte[] this,
    Encoding encoding
)
```

## Parameters

*this*

Type: [SystemByte](#)

The Base64 encoded byte array to convert

*encoding*

Type: [System.TextEncoding](#)

The encoding to use

## Return Value

Type: [String](#)

A non-encoded string

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension](#)

[Methods \(C# Programming Guide\)](#).

## ► See Also

[Reference](#)

[FromExtensions Class](#)

[FromBase64 Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# FromExtensionsFromBase64 Method (String, Encoding)

Converts a Base64 encoded string back to a normal string

**Namespace:** [W](#)

**Assembly:** Tungsten.From (in Tungsten.From.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static string FromBase64(  
    this string this,  
    Encoding encoding  
)
```

[Copy](#)

## Parameters

*this*

Type: [System.String](#)

The Base64 encoded string to convert

*encoding*

Type: [System.TextEncoding](#)

The encoding to use

## Return Value

Type: [String](#)

A non-encoded string

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [String](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension](#)

[Methods \(C# Programming Guide\)](#).

## See Also

[Reference](#)

[FromExtensions Class](#)

[FromBase64 Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# FromExtensionsFromCompressed Method

Decompresses the byte array using  
System.IO.Compression.DeflateStream

**Namespace:** [W](#)

**Assembly:** Tungsten.From (in Tungsten.From.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static byte[] FromCompressed(  
    this byte[] bytes  
)
```

## Parameters

*bytes*

Type: [SystemByte](#)

The byte array containing compressed data

## Return Value

Type: [Byte](#)

A byte array of the decompressed data

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type . When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

Reference

[FromExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IOwnedProperty Interface

Used by `PropertyHostMethods.InitializeProperties` to find properties on which to set the owner. This interface is not used by self-owned properties.

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public interface IOwnedProperty
```

The `IOwnedProperty` type exposes the following members.

## ► Methods

	Name	Description
≡	<a href="#">SetOwner</a>	Sets the property owner to the specified value

[Top](#)

## ► See Also

[Reference](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# IOwnedProperty Methods

The [IOwnedProperty](#) type exposes the following members.

## ▪ Methods

	Name	Description
	<a href="#">SetOwner</a>	Sets the property owner to the specified value

[Top](#)

## ▪ See Also

[Reference](#)

[IOwnedProperty Interface](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IOwnedPropertySetOwner

## Method

Sets the property owner to the specified value

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

### ► Syntax

```
C#  
void SetOwner(  
    Object owner  
)
```

[Copy](#)

### Parameters

*owner*

Type: [SystemObject](#)

The new property owner

### ► See Also

Reference

[IOwnedProperty Interface](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# IProperty Interface

The base interface which Property must support

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public interface IProperty
```

The [IProperty](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">IsDirty</a>	True if the property's value has changed since initialization or since the last call to <a href="#">MarkAsClean</a>

[Top](#)

## ► See Also

[Reference](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# IProperty Properties

The [IProperty](#) type exposes the following members.

## ► Properties

Name	Description
 <a href="#">IsDirty</a>	True if the property's value has changed since initialization or since the last call to <a href="#">MarkAsClean</a>

[Top](#)

## ► See Also

[Reference](#)

[IProperty Interface](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IPropertyIsDirty Property

True if the property's value has changed since initialization or since the last call to `MarkAsClean`

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

```
bool IsDirty { get; set; }
```

[Copy](#)

Property Value

Type: [Boolean](#)

## ► See Also

Reference

[IProperty Interface](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IProperty<TValue> Interface

The base interface which Property must support

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public interface IProperty<TValue> : IProperty
```

Type Parameters

*TValue*

The type of value for the property

The [IProperty<TValue>](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">IsDirty</a>	True if the property's value has changed since initialization or since the last call to <a href="#">MarkAsClean</a> (Inherited from <a href="#">IProperty</a> .)
	<a href="#">Value</a>	The value of the property

[Top](#)

## ► See Also

## Reference

### [W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IProperty TValue Properties

The [IProperty TValue](#) generic type exposes the following members.

## ► Properties

Name	Description
 <a href="#">IsDirty</a>	True if the property's value has changed since initialization or since the last call to <a href="#">MarkAsClean</a> (Inherited from <a href="#">IProperty</a> .)
 <a href="#">Value</a>	The value of the property

[Top](#)

## ► See Also

[Reference](#)

[IProperty TValue Interface](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IProperty TValue Value Property

The value of the property

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
TValue Value { get; set; }
```

Property Value

Type: [TValue](#)

## ► See Also

Reference

[IProperty TValue Interface](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LockableTValue Class

Extends LockableSlim with ValueChangedDelegate notification

## ► Inheritance Hierarchy

```
SystemObject W.Threading.LockersReaderWriterLockerTValue
  WLockableSlimTValue
    WLockableTValue
      WPropertySlimTValue
```

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C#
public class Lockable<TValue> : LockableSlim<TValue>, IDisposable
```

### Type Parameters

*TValue*

The data Type to be used

The [LockableTValue](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">LockableTValue</a>	Constructs a new

## Lockable< TValue >

≡	<a href="#">LockableTValue(ActionObject, TValue, TValue)</a>	Constructs a new Lockable< TValue >
≡	<a href="#">LockableTValue(TValue)</a>	Constructs a new Lockable< TValue >
≡	<a href="#">LockableTValue(TValue, ActionObject, TValue, TValue)</a>	Constructs a new Lockable< TValue >

[Top](#)

## Properties

Name	Description
 <a href="#">Value</a>	Get or Set the value (Inherited from <a href="#">LockableSlimTValue</a> .)

[Top](#)

## Methods

Name	Description
≡ <a href="#">Dispose</a>	Disposes the Lockable resources
≡ <a href="#">Equals</a>	Determines whether object is equal to the (Inherited from <a href="#">Object</a> )
≡ <a href="#">Finalize</a>	Allows an object to free resources and perform operations before it is garbage collected. (Inherited from <a href="#">Object</a> )

≡	<a href="#">GetHashCode</a>	Serves as the default hash function for this type. (Inherited from <a href="#">Object</a> )
≡	<a href="#">GetState</a>	Retrieves the internal state of a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> )
💡	<a href="#">GetValue</a>	Gets the underlying value. (Inherited from <a href="#">LockBase</a> )
💡	<a href="#">InformWaiters</a>	Informs those who are waiting on a <a href="#">WaitForChanged</a> that something has changed.
≡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action function on a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value on a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action function asynchronously on a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value asynchronously on a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )

	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> )
	<a href="#">OnValueChanged</a>	Calls <a href="#">RaiseValueChanged</a> to raise the <a href="#">ValueChanged</a> event.
	<a href="#">RaiseValueChanged</a>	Raises the <a href="#">ValueChanged</a> event.
	<a href="#">SetState</a>	Sets the internal state of the <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
	<a href="#">SetValue</a>	Sets the value and raises the <a href="#">ValueChanged</a> event. (Overrides <a href="#">LockableSlimTValue.SetValue</a> )
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> )
	<a href="#">WaitForValueChanged</a>	Allows the caller to block until a <a href="#">ValueChanged</a> event occurs.

[Top](#)

## Events

	Name	Description
	<a href="#">ValueChanged</a>	Raised when the value has changed.

[Top](#)

## Fields

Name	Description
 <a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from <a href="#">ReaderWriterLockerTState</a> .)
 <a href="#">State</a>	The internal state (Inherited from <a href="#">ReaderWriterLockerTState</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a

Monitor lock  
(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by

[PropertyHostExtensions.\)](#)

---



## [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions.](#))

---

[Top](#)

## ► See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LockableTValue Constructor

## ▪ Overload List

Name	Description
<a href="#">LockableTValue</a>	Constructs a new Lockable<TValue>
<a href="#">LockableTValue(ActionObject, TValue, TValue)</a>	Constructs a new Lockable<TValue>
<a href="#">LockableTValue(TValue)</a>	Constructs a new Lockable<TValue>
<a href="#">LockableTValue(TValue, ActionObject, TValue, TValue)</a>	Constructs a new Lockable<TValue>

[Top](#)

## ▪ See Also

[Reference](#)

[LockableTValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# Lockable TValue Constructor

Constructs a new Lockable<TValue>

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public Lockable()
```

## ► See Also

[Reference](#)

[Lockable TValue Class](#)

[Lockable TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable<TValue> Constructor (Action<Object, TValue, TValue>, TValue)

Constructs a new Lockable<TValue>

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public Lockable(
    Action<Object, TValue, TValue> onValueChanged)
{}
```

### Parameters

*onValueChanged*

Type: [SystemActionObject, TValue, TValue](#)

The Action to call when the value has changed

## ► See Also

Reference

[LockableTValue Class](#)

[LockableTValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# Lockable<TValue> Constructor (TValue)

Constructs a new Lockable<TValue>

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C#  
public Lockable(  
    TValue initialValue  
)
```

[Copy](#)

## Parameters

*initialValue*

Type: [TValue](#)

The initial value

## ► See Also

### Reference

[Lockable<TValue> Class](#)

[Lockable<TValue> Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# Lockable<TValue> Constructor (TValue, ActionObject, TValue, TValue)

Constructs a new Lockable<TValue>

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public Lockable(
    TValue initialValue,
    Action<Object, TValue, TValue> onValueChanged)

```

### Parameters

*initialValue*

Type: [TValue](#)

[Missing <param name="initialValue"/> documentation for  
"M:W.Lockable`1.#ctor(`0,System.Action{System.Object,`0,`0})"]

*onValueChanged*

Type: [SystemActionObject](#), [TValue](#), [TValue](#)

[Missing <param name="onValueChanged"/> documentation for  
"M:W.Lockable`1.#ctor(`0,System.Action{System.Object,`0,`0})"]

## ► See Also

[Reference](#)

[Lockable TValue Class](#)

[Lockable TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable TValue Properties

The [LockableTValue](#) generic type exposes the following members.

## ► Properties

Name	Description
 <a href="#">Value</a>	Get or Set the value (Inherited from <a href="#">LockableSlimTValue</a> .)

[Top](#)

## ► See Also

Reference

[LockableTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable TValue Methods

The [Lockable TValue](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disposes the Lockable resources
<a href="#">Equals</a>	Determines whether object is equal to the (Inherited from Object)
<a href="#">Finalize</a>	Allows an object to release unmanaged resources and perform other operations before it is garbage collected. (Inherited from Object)
<a href="#">GetHashCode</a>	Serves as the default (Inherited from Object)
<a href="#">GetState</a>	Retrieves the internal state of a ReaderWriterLock (Inherited from ReaderWriterLocker)
<a href="#">GetType</a>	Gets the Type of the (Inherited from Object)
<a href="#">GetValue</a>	Gets the underlying value (Inherited from Lockable)

	<a href="#">InformWaiters</a>	Informs those who a WaitForChanged tha changed
≡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action f ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function . ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action f ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function . ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<a href="#">MemberwiseClone</a>	Creates a shallow cc Object. (Inherited from <a href="#">Object</a> )
💡	<a href="#">OnValueChanged</a>	Calls RaiseValueChanged event
💡	<a href="#">RaiseValueChanged</a>	Raises the ValueCh
≡	<a href="#">SetState</a>	Sets the internal stat ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )



## SetValue

Sets the value and raises the ValueChanged event.  
(Overrides [LockableSlimTValue.SetValue\(T\)](#))



## ToString

Returns a string that represents the current object.  
(Inherited from [Object](#))



## WaitForValueChanged

Allows the caller to block until changes

[Top](#)

## Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> )

[MonitorExtensions.](#))

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )

## [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[LockableTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable TValue Dispose Method

Disposes the Lockable and releases resources

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements

[IDisposableDispose](#)

## ► See Also

[Reference](#)

[Lockable TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable TValue InformWaiters Method

Informs those who are waiting on WaitForChanged that the value has changed

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
protected virtual void InformWaiters()
```

## ► See Also

[Reference](#)

[LockableTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable TValueOnValueChanged Method

Calls RaiseValueChanged to raise the ValueChanged event

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C#  
protected virtual void OnValueChanged(  
    Object sender,  
    TValue oldValue,  
    TValue newValue  
)
```

[Copy](#)

## Parameters

*sender*

Type: [SystemObject](#)

The object initiating the change

*oldValue*

Type: [TValue](#)

The previous value

*newValue*

Type: [TValue](#)

The current value

## ► See Also

Reference

Lockable TValue Class  
W Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable TValueRaiseValueChanged Method

Raises the ValueChanged event

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
protected void RaiseValueChanged(
    Object sender,
    TValue oldValue,
    TValue newValue
)
```

### Parameters

*sender*

Type: [SystemObject](#)

[Missing <param name="sender"/> documentation for  
"M:W.Lockable`1.RaiseValueChanged(System.Object,`0,`0)"]

*oldValue*

Type: [TValue](#)

The previous value

*newValue*

Type: [TValue](#)

The current value

## ► See Also

Reference

[LockableTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LockableTValueSetValue Method

Sets the value and raises the ValueChanged event

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
protected override void SetValue(  
    TValue newValue  
)
```

## Parameters

*newValue*

Type: [TValue](#)

The new value

## ► See Also

Reference

[LockableTValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# Lockable TValue WaitForValueChange Method

Allows the caller to block until Value changes

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public bool WaitForValueChanged(  
    int msTimeout = -1  
)
```

### Parameters

#### *msTimeout (Optional)*

Type: [SystemInt32](#)

The number of milliseconds to wait for the value to change

### Return Value

Type: [Boolean](#)

True if the value changed within the specified timeout period, otherwise False

## ► See Also

### Reference

[LockableTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable TValue Events

The [Lockable TValue](#) generic type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">ValueChanged</a>	Raised when the value has changed

[Top](#)

## ▪ See Also

Reference

[Lockable TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable TValue ValueChanged Event

Raised when the value has changed

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public event ValueChangedDelegate<TValue> ValueC
```

Value

Type: [WValueChangedDelegate TValue](#)

## ► See Also

[Reference](#)

[Lockable TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Lockable TValue Fields

The [Lockable TValue](#) generic type exposes the following members.

## ▪ Fields

Name	Description
 <a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from <a href="#">ReaderWriterLockerTState</a> .)
 <a href="#">State</a>	The internal state (Inherited from <a href="#">ReaderWriterLockerTState</a> .)

[Top](#)

## ▪ See Also

[Reference](#)

[Lockable TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# LockableSlim TValue Class

Uses ReaderWriterLock to provide thread-safe access to an underlying value

## ► Inheritance Hierarchy

```
SystemObject W.Threading.LockersReaderWriterLocker TValue
WLockableSlimTValue
WLockableTValue
```

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public class LockableSlim<TValue> : ReaderWriterI
```

### Type Parameters

*TValue*

The Type of value

The [LockableSlimTValue](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">LockableSlimTValue</a>	Constructs a new LockableSlim with a default initial

value

---

	<a href="#">LockableSlimTValue(TValue)</a>	Constructs a new LockableSlim assigning an initial value
---	--	--

---

[Top](#)

## ► Properties

Name	Description
	<a href="#">Value</a> Get or Set the value

---

[Top](#)

## ► Methods

Name	Description
	<a href="#">Dispose</a> Disposes the ReaderWriterLocker releases resources (Inherited from <a href="#">ReaderWriterLocker</a> )
	<a href="#">Equals</a> Determines whether specified object is equal to the current object. (Inherited from <a href="#">Object</a> )
	<a href="#">Finalize</a> Allows an object to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> )

---

≡	<a href="#">GetHashCode</a>	Serves as the default function. (Inherited from <a href="#">Object</a> )
≡	<a href="#">GetState</a>	Retrieves the internal state from within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the instance. (Inherited from <a href="#">Object</a> )
≡	<a href="#">GetValue</a>	Gets the underlying value.
≡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action function within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action function asynchronously within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value asynchronously within a <a href="#">ReaderWriterLockSlim</a> .

(Inherited from  
[ReaderWriterLocker](#))

🔑	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> )
≡	<a href="#">SetState</a>	Sets the internal state within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
🔑	<a href="#">SetValue</a>	Sets the underlying value.
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> )

[Top](#)

## Fields

	Name	Description
🔑	<a href="#">Locker</a>	The <a href="#">ReaderWriterLocker</a> used to access the State. (Inherited from <a href="#">ReaderWriterLockerTState</a> .)
🔑	<a href="#">State</a>	The internal state. (Inherited from <a href="#">ReaderWriterLockerTState</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ► Remarks

Can be overridden to provide additional functionality

## ► See Also

## Reference

### [W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LockableSlim TValue Constructor

## ▪ Overload List

Name	Description
 <a href="#">LockableSlimTValue</a>	Constructs a new LockableSlim with a default initial value
 <a href="#">LockableSlimTValue(TValue)</a>	Constructs a new LockableSlim assigning an initial value

[Top](#)

## ▪ See Also

Reference

[LockableSlimTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LockableSlim TValue Constructor

Constructs a new LockableSlim with a default initial value

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public LockableSlim()
```

## ► See Also

Reference

[LockableSlim TValue Class](#)

[LockableSlim TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LockableSlimTValue Constructor (*TValue*)

Constructs a new LockableSlim assigning an initial value

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C#  
public LockableSlim(  
    TValue initialValue  
)
```

[Copy](#)

## Parameters

*initialValue*

Type: [TValue](#)

The initial value to assign

## ► See Also

### Reference

[LockableSlimTValue Class](#)

[LockableSlimTValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# LockableSlim TValue Properties

The [LockableSlim TValue](#) generic type exposes the following members.

## ► Properties

Name	Description
 <a href="#">Value</a>	Get or Set the value

[Top](#)

## ► See Also

[Reference](#)

[LockableSlim TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LockableSlim TValue Value Property

Get or Set the value

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public TValue Value { get; set; }
```

Return Value

Type: [TValue](#)

The current value

## ► See Also

[Reference](#)

[LockableSlimTValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# LockableSlim TValue Methods

The [LockableSlim TValue](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disposes the ReaderWriterLocker releases resources (Inherited from <a href="#">ReaderWriterLocker</a> )
<a href="#">Equals</a>	Determines whether specified object is equal to the current object. (Inherited from <a href="#">Object</a> )
<a href="#">Finalize</a>	Allows an object to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> )
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> )
<a href="#">GetState</a>	Retrieves the internal state from within a ReaderWriterLockSlim. (Inherited from <a href="#">ReaderWriterLocker</a> )

---

≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the instance. (Inherited from <a href="#">Object</a> )
💡	<a href="#">GetValue</a>	Gets the underlying value.
≡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action function within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action function within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> )
≡	<a href="#">SetState</a>	Sets the internal state within a <a href="#">ReaderWriterLockSlim</a> .

(Inherited from  
[ReaderWriterLocker](#))

 	<a href="#">SetValue</a>	Sets the underlying value.
 	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> )

[Top](#)

## ◀ Extension Methods

Name	Description
 	<a href="#">AsTTType</a> Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 	<a href="#">InitializeProperties</a> Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 	<a href="#">InLock(Action)</a> Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ◀ See Also

[Reference](#)

[LockableSlim TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LockableSlim TValueGetValue Method

Gets the underlying value

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
protected virtual TValue GetValue()
```

Return Value

Type: *TValue*

[Missing <returns> documentation for "M:W.LockableSlim`1.GetValue"]

## ► See Also

Reference

[LockableSlimTValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# LockableSlimTValueSetValue Method

Sets the underlying value

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C#  
protected virtual void SetValue(  
    TValue value  
)
```

[Copy](#)

## Parameters

*value*

Type: [TValue](#)

The new value

## ► See Also

Reference

[LockableSlimTValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# LockableSlim TValue Fields

The [LockableSlim TValue](#) generic type exposes the following members.

## ► Fields

	Name	Description
💡	<a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from <a href="#">ReaderWriterLockerTState.</a> )
💡	<a href="#">State</a>	The internal state (Inherited from <a href="#">ReaderWriterLockerTState.</a> )

[Top](#)

## ► See Also

[Reference](#)

[LockableSlim TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# Property TValue Class

A Property with no owner (self-owned)

## ► Inheritance Hierarchy

```
SystemObject W.Threading.LockersReaderWriterLocker TValue
  WLockableSlim TValue
    WLockable TValue
      WPropertySlim TValue
        WPropertyBaseProperty TValue, TValue
          WProperty TValue
```

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public class Property<TValue> : PropertyBase<Prop
```

Type Parameters

*TValue*

The type of the property value

The [Property TValue](#) type exposes the following members.

## ► Constructors

Name	Description
<a href="#">Property TValue</a>	Initializes a new

instance of the  
[Property TValue](#)  
class

---

	<a href="#">Property TValue(ActionObject, TValue, TValue)</a>	Initializes a new instance of the <a href="#">Property TValue</a> class
	<a href="#">Property TValue(TValue)</a>	Initializes a new instance of the <a href="#">Property TValue</a> class
	<a href="#">Property TValue(TValue, ActionObject, TValue, TValue)</a>	Initializes a new instance of the <a href="#">Property TValue</a> class

---

[Top](#)

## Properties

	Name	Description
	<a href="#">DefaultValue</a>	Allows the programmer to assign a default value which can be reset via the <a href="#">ResetToDefaultValue</a> method. This value does not have to be the initial value. (Inherited from <a href="#">PropertyBaseTOwner</a> , <a href="#">TValue</a> .)
	<a href="#">IsDirty</a>	True if Value has changed since initialization or since the last call to <a href="#">MarkAsClean</a> (Inherited from <a href="#">PropertyBaseTOwner</a> , <a href="#">TValue</a> .)



### Owner

The property owner  
(Inherited from  
[PropertyBaseTOwner, TValue.](#))



### Value

Get or Set the value  
(Inherited from  
[LockableSlim TValue.](#))

[Top](#)

## ◀ Methods

Name	Description
<a href="#">Dispose</a>	Disposes the LockableSlim TValue class. This method releases resources (Inherited from <a href="#">Lockable TValue.</a> )
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> )
<a href="#">Finalize</a>	Allows an object to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> )
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> )
<a href="#">GetState</a>	Retrieves the internal state from within a <a href="#">ReaderWriterLockSlim</a> .

(Inherited from  
[ReaderWriterLocker](#))

---

≡ [GetType](#) Gets the [Type](#) of the instance.  
(Inherited from [Object](#))

---

≡ [GetValue](#) Gets the underlying value.  
(Inherited from [LockableSlim TValue](#))

---

≡ [InformWaiters](#) Informs those who are waiting on [WaitForCondition](#) that the value has changed.  
(Inherited from [Lockable TValue](#).)

---

≡ [InLock\(LockTypeEnum, ActionTState\)](#) Executes an action function within a [ReaderWriterLockSlim](#).  
(Inherited from [ReaderWriterLocker](#))

---

≡ [InLockTValue\(LockTypeEnum, FuncTState, TValue\)](#) Executes a function that returns a value within a [ReaderWriterLockSlim](#).  
(Inherited from [ReaderWriterLocker](#))

---

≡ [InLockAsync\(LockTypeEnum, ActionTState\)](#) Executes an action function asynchronously within a [ReaderWriterLockSlim](#).  
(Inherited from [ReaderWriterLocker](#))

---

≡ [InLockAsyncTValue\(LockTypeEnum, FuncTState, TValue\)](#) Executes a function that returns a value asynchronously within a [ReaderWriterLockSlim](#).

(Inherited from  
[ReaderWriterLocker](#))

 <a href="#">LoadValue</a>	Sets Value without raising notification events (Inherited from <a href="#">PropertyBaseTOwner&lt; TValue&gt;</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> )
 <a href="#">OnPropertyChanged</a>	Calls <a href="#">RaisePropertyChanged</a> to raise the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">PropertyBaseTOwner&lt; TValue&gt;</a> .)
 <a href="#">OnPropertyChanging</a>	Calls <a href="#">RaisePropertyChanging</a> to raise the <a href="#">PropertyChanging</a> event (Inherited from <a href="#">PropertyBaseTOwner&lt; TValue&gt;</a> .)
 <a href="#">OnValueChanged</a>	Calls <a href="#">RaiseValueChanged</a> to raise the <a href="#">ValueChanged</a> event (Inherited from <a href="#">PropertyBaseTOwner&lt; TValue&gt;</a> .)
 <a href="#">RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">PropertySlimTValue</a> .)

---

💡	<a href="#">RaiseOnPropertyChanging</a>	Raises the <code>PropertyChanging</code> event (Inherited from <code>PropertySlim TValue</code> .)
💡	<a href="#">RaiseValueChanged</a>	Raises the <code>ValueChanged</code> event (Inherited from <code>Lockable TValue</code> .)
💡	<a href="#">ResetDefaultValue</a>	Resets the Value to the value provided by <code>DefaultValue</code> (Inherited from <code>PropertyBase TValue</code> .)
💡	<a href="#">SetState</a>	Sets the internal state within a <code>ReaderWriterLockSlim</code> (Inherited from <code>ReaderWriterLocker</code> .)
💡	<a href="#">SetValue</a>	Calls <code>OnPropertyChanged</code> on assignment (Inherited from <code>PropertySlim TValue</code> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <code>Object</code> .)
💡	<a href="#">WaitForValuechanged</a>	Allows the caller to block until Value changes (Inherited from <code>Lockable TValue</code> .)

---

[Top](#)

## Events

	Name	Description
⚡	<a href="#">PropertyChanged</a>	(Inherited from <a href="#">PropertySlim TValue</a> .)
⚡	<a href="#">PropertyChanging</a>	(Inherited from <a href="#">PropertySlim TValue</a> .)
⚡	<a href="#">ValueChanged</a>	Raised when the value has changed (Inherited from <a href="#">Lockable TValue</a> .)

[Top](#)

## Operators

	Name	Description
⇄ 	<a href="#">(TValue to Property TValue)</a>	Implicit conversion from TValue to Property< TValue >
⇄ 	<a href="#">(Property TValue to TValue)</a>	Implicit conversion from Property< TValue > to TValue

[Top](#)

## Fields

	Name	Description
🔑	<a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from

[ReaderWriterLockerTState.\)](#)



### State

The internal state  
(Inherited from  
[ReaderWriterLockerTState.\)](#)

[Top](#)

## Extension Methods

Name	Description
<a href="#">AsTTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

Reference

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property TValue Constructor

## ► Overload List

	Name	Description
≡	<a href="#">Property TValue</a>	Initializes a new instance of the <a href="#">Property TValue</a> class
≡	<a href="#">Property TValue(ActionObject, TValue, TValue)</a>	Initializes a new instance of the <a href="#">Property TValue</a> class
≡	<a href="#">Property TValue(TValue)</a>	Initializes a new instance of the <a href="#">Property TValue</a> class
≡	<a href="#">Property TValue(TValue, ActionObject, TValue, TValue)</a>	Initializes a new instance of the <a href="#">Property TValue</a> class

[Top](#)

## ► See Also

Reference

[Property TValue Class](#)

[W Namespace](#)



## Tungsten

W<sup>74</sup>

# PropertyTValue Constructor

Initializes a new instance of the [PropertyTValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public Property()
```

## ► See Also

[Reference](#)

[PropertyTValue Class](#)

[PropertyTValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyTValue Constructor (ActionObject, TValue, TValue)

Initializes a new instance of the [PropertyTValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public Property(
    Action<Object, TValue, TValue> onValueChanged)

```

### Parameters

*onValueChanged*

Type: [SystemActionObject, TValue, TValue](#)

[Missing <param name="onValueChanged"/> documentation for "M:W.Property`1.#ctor(System.Action{System.Object,`0,`0})"]

## ► See Also

### Reference

[PropertyTValue Class](#)

[PropertyTValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyTValue Constructor (*TValue*)

Initializes a new instance of the [PropertyTValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public Property(  
    TValue defaultValue  
)
```

[Copy](#)

## Parameters

*defaultValue*

Type: [TValue](#)

[Missing <param name="defaultValue"/> documentation for  
"M:W.Property`1.#ctor(`0)"]

## ► See Also

[Reference](#)

[PropertyTValue Class](#)

[PropertyTValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# Property TValue Constructor (TValue, ActionObject, TValue, TValue)

Initializes a new instance of the [Property TValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public Property(
    TValue defaultValue,
    Action<Object, TValue, TValue> onValueChanged)

```

## Parameters

*defaultValue*

Type: [TValue](#)

[Missing <param name="defaultValue"/> documentation for  
"M:W.Property`1.#ctor(`0,System.Action{System.Object,`0,`0})"]

*onValueChanged*

Type: [SystemActionObject](#), [TValue](#), [TValue](#)

[Missing <param name="onValueChanged"/> documentation for  
"M:W.Property`1.#ctor(`0,System.Action{System.Object,`0,`0})"]

## ► See Also

**Reference**

[PropertyTValue Class](#)

[PropertyTValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property TValue Properties

The [Property TValue](#) generic type exposes the following members.

## Properties

	Name	Description
	<a href="#">DefaultValue</a>	Allows the programmer to assign a default value which can be reset via the <code>ResetToDefaultValue</code> method. This value does not have to be the initial value. (Inherited from <a href="#">PropertyBaseTOwner, TValue</a> .)
	<a href="#">IsDirty</a>	True if Value has changed since initialization or since the last call to <code>MarkAsClean</code> (Inherited from <a href="#">PropertyBaseTOwner, TValue</a> .)
	<a href="#">Owner</a>	The property owner (Inherited from <a href="#">PropertyBaseTOwner, TValue</a> .)
	<a href="#">Value</a>	Get or Set the value (Inherited from <a href="#">LockableSlim TValue</a> .)

[Top](#)

## See Also

[Reference](#)  
[Property TValue Class](#)

## [W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property TValue Methods

The [Property TValue](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disposes the Lockable TValue releases resources (Inherited from <a href="#">Lockable TValue</a> .)
<a href="#">Equals</a>	Determines whether specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetState</a>	Retrieves the internal state from within a ReaderWriterLockSlim. (Inherited from <a href="#">ReaderWriterLocker</a> .)

≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the instance. (Inherited from <a href="#">Object</a> )
💡	<a href="#">GetValue</a>	Gets the underlying value. (Inherited from <a href="#">LockableSlim TValue</a> )
💡	<a href="#">InformWaiters</a>	Informs those who are waiting on <a href="#">WaitForCondition</a> that the value has changed. (Inherited from <a href="#">Lockable TValue</a> .)
≡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action function within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action function asynchronously within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value asynchronously within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )

≡	<a href="#">LoadValue</a>	Sets Value without raising notification events (Inherited from <a href="#">PropertyBaseTOwner&lt;TValue&gt;</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnPropertyChanged</a>	Calls <a href="#">RaisePropertyChanged</a> to raise the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">PropertyBaseTOwner&lt;TValue&gt;</a> .)
💡	<a href="#">OnPropertyChanging</a>	Calls <a href="#">RaisePropertyChanging</a> to raise the <a href="#">PropertyChanging</a> event (Inherited from <a href="#">PropertyBaseTOwner&lt;TValue&gt;</a> .)
💡	<a href="#">OnValueChanged</a>	Calls <a href="#">RaiseValueChanged</a> to raise the <a href="#">ValueChanged</a> event (Inherited from <a href="#">PropertyBaseTOwner&lt;TValue&gt;</a> .)
💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">PropertySlimTValue</a> .)
💡	<a href="#">RaiseOnPropertyChanging</a>	Raises the <a href="#">PropertyChanging</a> event

(Inherited from [PropertySlim TValue](#).)

 <a href="#">RaiseValueChanged</a>	Raises the ValueChanged event (Inherited from <a href="#">Lockable TValue</a> .)
 <a href="#">ResetDefaultValue</a>	Resets the Value to its default value provided by DefaultValue (Inherited from <a href="#">PropertyBase TValue</a> .)
 <a href="#">SetState</a>	Sets the internal state within a ReaderWriterLockSlim (Inherited from <a href="#">ReaderWriterLocker</a> .)
 <a href="#">SetValue</a>	Calls OnPropertyChanged on assignment (Inherited from <a href="#">PropertySlim TValue</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">WaitForValueChanged</a>	Allows the caller to block until Value changes (Inherited from <a href="#">Lockable TValue</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

[Reference](#)

[PropertyValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property TValue Events

The [Property TValue](#) generic type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">PropertyChanged</a>	(Inherited from <a href="#">PropertySlim TValue</a> .)
 <a href="#">PropertyChanging</a>	(Inherited from <a href="#">PropertySlim TValue</a> .)
 <a href="#">ValueChanged</a>	Raised when the value has changed (Inherited from <a href="#">Lockable TValue</a> .)

[Top](#)

## ▪ See Also

Reference

[Property TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyTValue Type Conversions

## ► Operators

	Name	Description
	(TValue to PropertyTValue)	Implicit conversion from TValue to Property<TValue>
	(PropertyTValue to TValue)	Implicit conversion from Property<TValue> to TValue

[Top](#)

## ► See Also

Reference

[PropertyTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyTValue Conversion Operators

## ▪ Overload List

	Name	Description
	<a href="#">(TValue to PropertyTValue)</a>	Implicit conversion from TValue to Property<TValue>
	<a href="#">(PropertyTValue to TValue)</a>	Implicit conversion from Property<TValue> to TValue

[Top](#)

## ▪ See Also

Reference

[PropertyTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property TValue Conversion (TValue to Property TValue)

Implicit conversion from TValue to Property<TValue>

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static implicit operator Property<TValue>
    TValue value
)
```

### Parameters

*value*

Type: [TValue](#)

The value from which to create a new Property<TValue>

### Return Value

Type: [Property TValue](#)

[Missing <returns> documentation for  
"M:W.Property`1.op\_Implicit(`0)~W.Property{`0}"]

## ► See Also

Reference

[Property TValue Class](#)

Overload

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyTValue Conversion (PropertyTValue to TValue)

Implicit conversion from Property<TValue> to TValue

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static implicit operator TValue (
    Property<TValue> property
)
```

### Parameters

*property*

Type: [WPropertyTValue](#)

The Property<TValue> from which to obtain the value

### Return Value

Type: [TValue](#)

[Missing <returns> documentation for  
"M:W.Property`1.op\_Implicit(W.Property{`0})~`0"]

## ► See Also

### Reference

[PropertyTValue Class](#)

[Overload](#)

[W Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property TValue Fields

The [Property TValue](#) generic type exposes the following members.

## ► Fields

Name	Description
 <a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from <a href="#">ReaderWriterLockerTState.</a> )
 <a href="#">State</a>	The internal state (Inherited from <a href="#">ReaderWriterLockerTState.</a> )

[Top](#)

## ► See Also

[Reference](#)

[Property TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# Property`T`Owner, `T`Value Class

A generic Property with an owner

## ► Inheritance Hierarchy

```
SystemObject W.Threading.LockersReaderWriterLockerTValue
  WLockableSlimTValue
    WLockableTValue
      WPropertySlimTValue
        WPropertyBaseTOwner, TValue
          WPropertyTOwner, TValue
```

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public class Property<TOwner, TValue> : Property
  IOwnedProperty
where TOwner : class
```

### Type Parameters

`T`Owner

The type of owner

`T`Value

The type of the property value

The `PropertyTOwner, TValue` type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PropertyTOwner, TValue</a>	Constructs a new Property
≡	<a href="#">PropertyTOwner, TValue(ActionObject, TValue, TValue)</a>	Initializes a new instance of the <b>PropertyTOwner, TValue</b> class
≡	<a href="#">PropertyTOwner, TValue(TOwner)</a>	Constructs a new Property
≡	<a href="#">PropertyTOwner, TValue(TValue)</a>	Constructs a new Property
≡	<a href="#">PropertyTOwner, TValue(TOwner, ActionObject, TValue, TValue)</a>	Initializes a new instance of the <b>PropertyTOwner, TValue</b> class
≡	<a href="#">PropertyTOwner, TValue(TOwner, TValue)</a>	Constructs a new Property
≡	<a href="#">PropertyTOwner, TValue(TValue, ActionObject, TValue, TValue)</a>	Initializes a new instance of the <b>PropertyTOwner, TValue</b> class
≡	<a href="#">PropertyTOwner, TValue(TOwner, TValue, ActionObject, TValue, TValue)</a>	Initializes a new instance of the <b>PropertyTOwner, TValue</b> class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">DefaultValue</a>	Allows the programmer to assign a default value which can be reset via the <code>ResetToDefaultValue</code> method. This value does not have to be the initial value. (Inherited from <a href="#">PropertyBaseTOwner, TValue.</a> )
	<a href="#">IsDirty</a>	True if Value has changed since initialization or since the last call to <code>MarkAsClean</code> (Inherited from <a href="#">PropertyBaseTOwner, TValue.</a> )
	<a href="#">Owner</a>	The property owner (Inherited from <a href="#">PropertyBaseTOwner, TValue.</a> )
	<a href="#">Value</a>	Get or Set the value (Inherited from <a href="#">LockableSlimTValue.</a> )

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Dispose</a>	Disposes the Lockable releases resources (Inherited from <a href="#">LockableTValue.</a> )
	<a href="#">Equals</a>	Determines whether

specified object is **executed** by the current object.  
(Inherited from [Object](#))

💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> )
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> )
💡	<a href="#">GetState</a>	Retrieves the internal state from within a <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the instance. (Inherited from <a href="#">Object</a> )
💡	<a href="#">GetValue</a>	Gets the underlying value. (Inherited from <a href="#">LockableSlim TValue</a> )
💡	<a href="#">InformWaiters</a>	Informs those who are waiting on <a href="#">WaitForCondition</a> that the value has changed. (Inherited from <a href="#">Lockable TValue</a> .)
💡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action function within a <a href="#">ReaderWriterLockSlim</a> .

		(Inherited from <a href="#">ReaderWriterLocker</a> )
 <a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function within a ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )	
 <a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action f within a ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )	
 <a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function within a ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )	
 <a href="#">LoadValue</a>	Sets Value without r notification events (Inherited from <a href="#">PropertyBaseTOwner</a> )	
 <a href="#">MemberwiseClone</a>	Creates a shallow cc the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> )	
 <a href="#">OnPropertyChanged</a>	Calls <a href="#">RaisePropertyChanged</a> to raise the PropertyChanged event (Inherited from <a href="#">PropertyBaseTOwner</a> )	

💡	<a href="#">OnPropertyChanging</a>	Calls <a href="#">RaisePropertyChanged</a> to raise the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">PropertyBaseTValue</a> .)
💡	<a href="#">OnValueChanged</a>	Calls <a href="#">RaiseValueChanged</a> to raise the <a href="#">ValueChanged</a> event (Inherited from <a href="#">PropertyBaseTValue</a> .)
💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">PropertySlimTValue</a> .)
💡	<a href="#">RaiseOnPropertyChanging</a>	Raises the <a href="#">PropertyChanging</a> event (Inherited from <a href="#">PropertySlimTValue</a> .)
💡	<a href="#">RaiseValueChanged</a>	Raises the <a href="#">ValueChanged</a> event (Inherited from <a href="#">LockableTValue</a> .)
💡	<a href="#">ResetDefaultValue</a>	Resets the Value to the value provided by <a href="#">DefaultValue</a> (Inherited from <a href="#">PropertyBaseTValue</a> .)
💡	<a href="#">SetState</a>	Sets the internal state within a

---

		ReaderWriterLockSlim (Inherited from <a href="#">ReaderWriterLocker</a> )
 <a href="#">SetValue</a>		Calls <a href="#">OnPropertyChanged</a> on assignment (Inherited from <a href="#">PropertySlimTValue</a> .)
 <a href="#">ToString</a>		Returns a string that represents the current object. (Inherited from <a href="#">Object</a> )
 <a href="#">WaitForValueChanged</a>		Allows the caller to block until Value changes (Inherited from <a href="#">LockableTValue</a> .)

---

[Top](#)

## Events

	Name	Description
 <a href="#">PropertyChanged</a>		(Inherited from <a href="#">PropertySlimTValue</a> .)
 <a href="#">PropertyChanging</a>		(Inherited from <a href="#">PropertySlimTValue</a> .)
 <a href="#">ValueChanged</a>		Raised when the value has changed (Inherited from <a href="#">LockableTValue</a> .)

---

[Top](#)

## Operators

	Name	Description
 <b>S</b>	<a href="#">(TValue to Property&lt;TOwner, TValue&gt;)</a>	Implicit conversion from TValue to Property<TOwner, TValue>
 <b>S</b>	<a href="#">(Property&lt;TOwner, TValue&gt; to TValue)</a>	Implicit conversion from Property<TOwner, TValue> to TValue

[Top](#)

## Fields

	Name	Description
	<a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from <a href="#">ReaderWriterLockerTState</a> .)
	<a href="#">State</a>	The internal state (Inherited from <a href="#">ReaderWriterLockerTState</a> .)

[Top](#)

## Extension Methods

	Name	Description
	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)

---

	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

---

[IsDirty](#) Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---

 [Lock](#) Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

 [MarkAsClean](#) Scans each field and property of type IProperty and sets its IsDirty flag to false  
(Defined by [PropertyHostExtensions](#).)

---

 [Unlock](#) Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

[Top](#)

## ▲ See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyTOwner, TValue Constructor

## ▪ Overload List

Name	Description
<a href="#">PropertyTOwner, TValue</a>	Constructs a new Property
<a href="#">PropertyTOwner, TValue(ActionObject, TValue, TValue)</a>	Initializes a new instance of the <a href="#">PropertyTOwner, TValue</a> class
<a href="#">PropertyTOwner, TValue(TOwner)</a>	Constructs a new Property
<a href="#">PropertyTOwner, TValue(TValue)</a>	Constructs a new Property
<a href="#">PropertyTOwner, TValue(TOwner, ActionObject, TValue, TValue)</a>	Initializes a new instance of the <a href="#">PropertyTOwner, TValue</a> class
<a href="#">PropertyTOwner, TValue(TOwner, TValue)</a>	Constructs a new Property
<a href="#">PropertyTOwner, TValue(TValue, ActionObject, TValue, TValue)</a>	Initializes a new instance of the <a href="#">PropertyTOwner, TValue</a> class

[PropertyTOwner](#),  
[TValue\(TOwner, TValue, ActionObject, TValue, TValue\)](#)

Initializes a new instance of the [PropertyTOwner](#), [TValue](#) class

---

[Top](#)

## ▲ See Also

Reference

[PropertyTOwner, TValue Class](#)  
[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyT<sub>Owner</sub>, TValue Constructor

Constructs a new Property

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

```
public Property()
```

Copy

## ► See Also

Reference

[PropertyT<sub>Owner</sub>, TValue Class](#)

[PropertyT<sub>Owner</sub>, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyTOwner, TValue Constructor (ActionObject, TValue, TValue)

Initializes a new instance of the [PropertyTOwner, TValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public Property(
    Action<Object, TValue, TValue> onValueChanged)

```

### Parameters

*onValueChanged*

Type: [SystemActionObject, TValue, TValue](#)

[Missing <param name="onValueChanged"/> documentation for  
"M:W.Property`2.#ctor(System.Action{System.Object,`1,`1})"]

## ► See Also

[Reference](#)

[PropertyTOwner, TValue Class](#)

[PropertyTOwner, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property $T$ Owner, TValue Constructor ( $T$ Owner)

Constructs a new Property

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public Property(  
    TOwner owner  
)
```

[Copy](#)

## Parameters

*owner*

Type: [TOwner](#)

The owner of the property

## ► See Also

### Reference

[Property \$T\$ Owner, TValue Class](#)

[Property \$T\$ Owner, TValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyT<sub>Owner</sub>, TValue Constructor (TValue)

Constructs a new Property

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public Property(  
    TValue defaultValue  
)
```

[Copy](#)

## Parameters

*defaultValue*

Type: [TValue](#)

The default and initial value of the property

## ► See Also

### Reference

[PropertyT<sub>Owner</sub>, TValue Class](#)

[PropertyT<sub>Owner</sub>, TValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# Property`T`Owner, `T`Value Constructor (`T`Owner, ActionObject, `T`Value, `T`Value)

Initializes a new instance of the [Property`T`Owner, `T`Value](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public Property(
    TOwner owner,
    Action<Object, TValue, TValue> onValueChanged)

```

## Parameters

*owner*

Type: `T`Owner

[Missing <param name="owner"/> documentation for  
"M:W.Property`2.#ctor(`0,System.Action{System.Object,`1,`1})"]

*onValueChanged*

Type: SystemActionObject, `T`Value, `T`Value

[Missing <param name="onValueChanged"/> documentation for  
"M:W.Property`2.#ctor(`0,System.Action{System.Object,`1,`1})"]

## ► See Also

## Reference

[PropertyTOwner, TValue Class](#)

[PropertyTOwner, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property $T$ Owner, $T$ Value Constructor ( $T$ Owner, $T$ Value)

Constructs a new Property

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public Property(
    TOwner owner,
    TValue defaultValue
)
```

### Parameters

*owner*

Type: [TOwner](#)

The owner of the property

*defaultValue*

Type: [TValue](#)

The default and initial value of the property

## ► See Also

[Reference](#)

[Property \$T\$ Owner,  \$T\$ Value Class](#)

[Property \$T\$ Owner,  \$T\$ Value Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyTOwner, TValue Constructor (TValue, ActionObject, TValue, TValue)

Initializes a new instance of the [PropertyTOwner, TValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public Property(
    TValue defaultValue,
    Action<Object, TValue, TValue> onValueChanged)

```

### Parameters

*defaultValue*

Type: [TValue](#)

[Missing <param name="defaultValue"/> documentation for  
"M:W.Property`2.#ctor(`1,System.Action{System.Object,`1,`1})"]

*onValueChanged*

Type: [SystemActionObject, TValue, TValue](#)

[Missing <param name="onValueChanged"/> documentation for  
"M:W.Property`2.#ctor(`1,System.Action{System.Object,`1,`1})"]

## ► See Also

## Reference

[PropertyTOwner, TValue Class](#)

[PropertyTOwner, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property`T`Owner, `T`Value Constructor (`T`Owner, `T`Value, ActionObject, `T`Value, `T`Value)

Initializes a new instance of the [Property`T`Owner, `T`Value](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public Property(
    TOwner owner,
    TValue defaultValue,
    Action<Object, TValue, TValue> onValueChanged)

```

### Parameters

*owner*

Type: [TOwner](#)

[Missing <param name="owner"/> documentation for  
"M:W.Property`2.#ctor(`0,`1,System.Action{System.Object,`1,`1})"]

*defaultValue*

Type: [TValue](#)

[Missing <param name="defaultValue"/> documentation for  
"M:W.Property`2.#ctor(`0,`1,System.Action{System.Object,`1,`1})"]

*onValueChanged*

Type: [SystemActionObject, TValue, TValue](#)

[Missing <param name="onValueChanged"/> documentation for

"M:W.Property`2.#ctor(`0,`1,System.Action{System.Object,`1,`1})"]

## ► See Also

[Reference](#)

[PropertyTOwner, TValue Class](#)

[PropertyTOwner, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyTOwner, TValue Properties

The [PropertyTOwner, TValue](#) generic type exposes the following members.

## Properties

	Name	Description
	<a href="#">DefaultValue</a>	Allows the programmer to assign a default value which can be reset via the <a href="#">ResetToDefaultValue</a> method. This value does not have to be the initial value. (Inherited from <a href="#">PropertyBaseTOwner, TValue</a> .)
	<a href="#">IsDirty</a>	True if Value has changed since initialization or since the last call to <a href="#">MarkAsClean</a> (Inherited from <a href="#">PropertyBaseTOwner, TValue</a> .)
	<a href="#">Owner</a>	The property owner (Inherited from <a href="#">PropertyBaseTOwner, TValue</a> .)
	<a href="#">Value</a>	Get or Set the value (Inherited from <a href="#">LockableSlimTValue</a> .)

[Top](#)

## ▲ See Also

Reference

[PropertyTOwner, TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyT`Owner`, TValue Methods

The `PropertyTOwner, TValue` generic type exposes the following members.

## Methods

Name	Description
<code>Dispose</code>	Disposes the LockableTValue object and releases resources. (Inherited from <code>LockableTValue</code> .)
<code>Equals</code>	Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code> .)
<code>Finalize</code>	Allows an object to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code> .)
<code>GetHashCode</code>	Serves as the default hash function. (Inherited from <code>Object</code> .)
<code>GetState</code>	Retrieves the internal state from within a <code>ReaderWriterLockSlim</code> object.

(Inherited from  
[ReaderWriterLocker](#))

---

≡  
[GetType](#)

Gets the [Type](#) of the instance.  
(Inherited from [Object](#))

---

≡  
[GetValue](#)

Gets the underlying value.  
(Inherited from [LockableSlim TValue](#))

---

≡  
[InformWaiters](#)

Informs those who are waiting on [WaitForCondition](#) that the value has changed.  
(Inherited from [Lockable TValue](#).)

---

≡  
[InLock\(LockTypeEnum, ActionTState\)](#)

Executes an action function within a [ReaderWriterLockSlim](#).  
(Inherited from [ReaderWriterLocker](#))

---

≡  
[InLockTValue\(LockTypeEnum, FuncTState, TValue\)](#)

Executes a function within a [ReaderWriterLockSlim](#).  
(Inherited from [ReaderWriterLocker](#))

---

≡  
[InLockAsync\(LockTypeEnum, ActionTState\)](#)

Executes an action function within a [ReaderWriterLockSlim](#).  
(Inherited from [ReaderWriterLocker](#))

---

≡  
[InLockAsyncTValue\(LockTypeEnum, FuncTState, TValue\)](#)

Executes a function within a [ReaderWriterLockSlim](#).

(Inherited from  
[ReaderWriterLocker](#))

 <a href="#">LoadValue</a>	Sets Value without raising notification events (Inherited from <a href="#">PropertyBaseTOwner</a> <a href="#">TValue</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> )
 <a href="#">OnPropertyChanged</a>	Calls <a href="#">RaisePropertyChanged</a> to raise the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">PropertyBaseTOwner</a> <a href="#">TValue</a> .)
 <a href="#">OnPropertyChanging</a>	Calls <a href="#">RaisePropertyChanging</a> to raise the <a href="#">PropertyChanging</a> event (Inherited from <a href="#">PropertyBaseTOwner</a> <a href="#">TValue</a> .)
 <a href="#">OnValueChanged</a>	Calls <a href="#">RaiseValueChanged</a> to raise the <a href="#">ValueChanged</a> event (Inherited from <a href="#">PropertyBaseTOwner</a> <a href="#">TValue</a> .)
 <a href="#">RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">PropertySlimTValue</a> .)

---

💡	<a href="#">RaiseOnPropertyChanging</a>	Raises the <code>PropertyChanging</code> event (Inherited from <code>PropertySlim TValue</code> .)
💡	<a href="#">RaiseValueChanged</a>	Raises the <code>ValueChanged</code> event (Inherited from <code>Lockable TValue</code> .)
💡	<a href="#">ResetDefaultValue</a>	Resets the Value to its default value provided by <code>DefaultValue</code> (Inherited from <code>PropertyBase TValue</code> .)
💡	<a href="#">SetState</a>	Sets the internal state within a <code>ReaderWriterLockSlim</code> (Inherited from <code>ReaderWriterLocker</code> .)
💡	<a href="#">SetValue</a>	Calls <code>OnPropertyChanged</code> on assignment (Inherited from <code>PropertySlim TValue</code> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <code>Object</code> .)
💡	<a href="#">WaitForValuechanged</a>	Allows the caller to block until Value changes (Inherited from <code>Lockable TValue</code> .)

---

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

---

 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## See Also

[Reference](#)

[PropertyTOwner, TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyT<sub>Owner</sub>, TValue Events

The [PropertyT<sub>Owner</sub>, TValue](#) generic type exposes the following members.

## ► Events

	Name	Description
⚡	<a href="#">PropertyChanged</a>	(Inherited from <a href="#">PropertySlimTValue</a> .)
⚡	<a href="#">PropertyChanging</a>	(Inherited from <a href="#">PropertySlimTValue</a> .)
⚡	<a href="#">ValueChanged</a>	Raised when the value has changed (Inherited from <a href="#">LockableTValue</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PropertyT<sub>Owner</sub>, TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyTOwner, TValue Type Conversions

## Operators

	Name	Description
	(TValue to PropertyTOwner, TValue)	Implicit conversion from TValue to Property<TOwner, TValue>
	(PropertyTOwner, TValue to TValue)	Implicit conversion from Property<TOwner, TValue> to TValue

[Top](#)

## See Also

[Reference](#)

[PropertyTOwner, TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyTOwner, TValue Conversion Operators

## ▪ Overload List

	Name	Description
	(TValue to PropertyTOwner, TValue)	Implicit conversion from TValue to Property<TOwner, TValue>
	(PropertyTOwner, TValue to TValue)	Implicit conversion from Property<TOwner, TValue> to TValue

[Top](#)

## ▪ See Also

[Reference](#)

[PropertyTOwner, TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property`TOwner, TValue` Conversion (`TValue` to Property`TOwner, TValue`)

Implicit conversion from `TValue` to `Property<TOwner, TValue>`

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static implicit operator Property<TOwner,
    TValue value
)
```

### Parameters

`value`

Type: [TValue](#)

The value from which to create a new `Property<TOwner, TValue>`

### Return Value

Type: [Property`TOwner, TValue`](#)

[Missing <returns> documentation for  
"M:W.Property`2.op\_Implicit(`1)~W.Property{`0,`1}"]

## ► See Also

Reference

PropertyTOwner, TValue Class  
Overload  
W Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Property $T$ Owner, TValue Conversion (Property $T$ Owner, TValue to TValue)

Implicit conversion from Property<TOwner, TValue> to TValue

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static implicit operator TValue (
    Property<TOwner, TValue> property
)
```

### Parameters

*property*

Type: [WProperty \$T\$ Owner, TValue](#)

The Property<TOwner, TValue> from which to obtain the value

### Return Value

Type: [TValue](#)

[Missing <returns> documentation for  
"M:W.Property`2.op\_Implicit(W.Property{`0,`1})~`1"]

## ► See Also

Reference

[Property \$T\$ Owner, TValue Class](#)

## Overload W Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyTOwner, TValue Fields

The [PropertyTOwner, TValue](#) generic type exposes the following members.

## ► Fields

	Name	Description
	<a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from <a href="#">ReaderWriterLockerTState</a> .)
	<a href="#">State</a>	The internal state (Inherited from <a href="#">ReaderWriterLockerTState</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PropertyTOwner, TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBase<TOwner, TValue> Class

[Missing <summary> documentation for "T:W.PropertyBase`2"]

## ► Inheritance Hierarchy

```
SystemObject W.Threading.LockersReaderWriterLocker TValue
  WLockableSlim TValue
  WLockable TValue
  WPropertySlim TValue
  WPropertyBaseTOwner, TValue
    WPropertyTValue
    WPropertyTOwner, TValue
```

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public abstract class PropertyBase<TOwner, TValue>
  IProperty<TValue>, IProperty, IDisposable
```

### Type Parameters

*TOwner*

[Missing <typeparam name="TOwner"/> documentation for  
"T:W.PropertyBase`2"]

*TValue*

[Missing <typeparam name="TValue"/> documentation for  
"T:W.PropertyBase`2"]

The `PropertyBaseTOwner, TValue` type exposes the following members.

## ► Constructors

Name	Description
<code>PropertyBaseTOwner, TValue</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(ActionObject, TValue, TValue)</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(TOwner)</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(TValue)</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(TOwner, ActionObject, TValue, TValue)</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(TOwner, TValue)</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class

 <a href="#">PropertyBaseTOwner, TValue(TValue, ActionObject, TValue, TValue)</a>	Initializes a new instance of the <b>PropertyBaseTOwner, TValue</b> class
 <a href="#">PropertyBaseTOwner, TValue(TOwner, TValue, ActionObject, TValue, TValue)</a>	Initializes a new instance of the <b>PropertyBaseTOwner, TValue</b> class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">DefaultValue</a>	Allows the programmer to assign a default value which can be reset via the <code>ResetToDefaultValue</code> method. This value does not have to be the initial value.
	<a href="#">IsDirty</a>	True if Value has changed since initialization or since the last call to <code>MarkAsClean</code>
	<a href="#">Owner</a>	The property owner
	<a href="#">Value</a>	Get or Set the value (Inherited from <a href="#">LockableSlimTValue</a> .)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Dispose</a>	Disposes the Lockal

(Inherited from [LockBase](#))

 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> )
 <a href="#">Finalize</a>	Allows an object to try to perform other cleanup before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> )
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> )
 <a href="#">GetState</a>	Retrieves the internal state of the <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLockBase</a> )
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> )
 <a href="#">GetValue</a>	Gets the underlying value. (Inherited from <a href="#">LockBase</a> )
 <a href="#">InformWaiters</a>	Informs those who are waiting on the lock that <a href="#">WaitForChanged</a> has been triggered. (Inherited from <a href="#">LockBase</a> )
 <a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action function on the lock. This method is called by <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLockBase</a> )
 <a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value on the lock. This method is called by <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLockBase</a> )
 <a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action function on the lock asynchronously. This method is called by <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLockBase</a> )

(Inherited from [Read](#))

≡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function ReaderWriterLockSlim.
≡	<a href="#">LoadValue</a>	Sets Value without raising events.
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy.
💡	<a href="#">OnPropertyChanged</a>	Calls <a href="#">RaisePropertyChanged</a> event.
💡	<a href="#">OnPropertyChanging</a>	Calls <a href="#">RaisePropertyChanging</a> event.
💡	<a href="#">OnValueChanged</a>	Calls <a href="#">RaiseValueChanged</a> event.
💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event.
💡	<a href="#">RaiseOnPropertyChanging</a>	Raises the <a href="#">PropertyChanging</a> event.
💡	<a href="#">RaiseValueChanged</a>	Raises the <a href="#">ValueChanged</a> event.
≡	<a href="#">ResetDefaultValue</a>	Resets the Value to its DefaultValue.

	<a href="#">SetState</a>	Sets the internal state. <a href="#">ReaderWriterLockSlim</a> (Inherited from <a href="#">ReaderWriterLockSlim</a> )
💡	<a href="#">SetValue</a>	Calls <a href="#">OnPropertyChanged</a> . (Inherited from <a href="#">PropertyChangedBase</a> )
📝	<a href="#">ToString</a>	Returns a string that represents this object. (Inherited from <a href="#">Object</a> )
⌚	<a href="#">WaitForValueChanged</a>	Allows the caller to block until the value changes. (Inherited from <a href="#">LockableTValue</a> )

[Top](#)

## Events

	Name	Description
⚡	<a href="#">PropertyChanged</a>	(Inherited from <a href="#">PropertySlimTValue</a> .)
⚡	<a href="#">PropertyChanging</a>	(Inherited from <a href="#">PropertySlimTValue</a> .)
⚡	<a href="#">ValueChanged</a>	Raised when the value has changed. (Inherited from <a href="#">LockableTValue</a> .)

[Top](#)

## Fields

	Name	Description
💡	<a href="#">Locker</a>	The ReaderWriterLocker used to access

the State  
(Inherited from  
[ReaderWriterLockerTState](#).)



### [State](#)

The internal state  
(Inherited from  
[ReaderWriterLockerTState](#).)

[Top](#)

## ► Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in

a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
---	--

 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
---	--

 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
--	--

 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
--	--

 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
---	--

 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
--	--

[Top](#)

## ◀ See Also

Reference

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# **PropertyBaseTOwner, TValue** Constructor

## ▪ Overload List

Name	Description
<code>PropertyBaseTOwner, TValue</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(ActionObject, TValue, TValue)</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(TOwner)</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(TValue)</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(TOwner, ActionObject, TValue, TValue)</code>	Initializes a new instance of the <code>PropertyBaseTOwner, TValue</code> class
<code>PropertyBaseTOwner, TValue(TOwner, TValue)</code>	Initializes a new instance of the

[PropertyBaseTOwner, TValue](#) class



[PropertyBaseTOwner, TValue\(TValue, ActionObject, TValue, TValue\)](#)

Initializes a new instance of the [PropertyBaseTOwner, TValue](#) class



[PropertyBaseTOwner, TValue\(TOwner, TValue, ActionObject, TValue, TValue\)](#)

Initializes a new instance of the [PropertyBaseTOwner, TValue](#) class

[Top](#)

## ► See Also

[Reference](#)

[PropertyBaseTOwner, TValue Class](#)  
[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValue Constructor

Initializes a new instance of the [PropertyBaseTOwner, TValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public PropertyBase()
```

## ► See Also

Reference

[PropertyBaseTOwner, TValue Class](#)

[PropertyBaseTOwner, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValue Constructor (ActionObject, TValue, TValue)

Initializes a new instance of the [PropertyBaseTOwner, TValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public PropertyBase(
    Action<Object, TValue, TValue> onValueChanged)

```

### Parameters

*onValueChanged*

Type: [SystemActionObject, TValue, TValue](#)

[Missing <param name="onValueChanged"/> documentation for "M:W.PropertyBase`2.#ctor(System.Action{System.Object,`1,`1})"]

## ► See Also

[Reference](#)

[PropertyBaseTOwner, TValue Class](#)

[PropertyBaseTOwner, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValue Constructor (TOwner)

Initializes a new instance of the [PropertyBaseTOwner, TValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public PropertyBase(  
    TOwner owner  
)
```

[Copy](#)

## Parameters

*owner*

Type: [TOwner](#)

[Missing <param name="owner"/> documentation for  
"M:W.PropertyBase`2.#ctor(0)"]

## ► See Also

Reference

[PropertyBaseTOwner, TValue Class](#)

[PropertyBaseTOwner, TValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValue Constructor (TValue)

Initializes a new instance of the [PropertyBaseTOwner, TValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public PropertyBase(  
    TValue defaultValue  
)
```

[Copy](#)

## Parameters

*defaultValue*

Type: [TValue](#)

[Missing <param name="defaultValue"/> documentation for  
"M:W.PropertyBase`2.#ctor(1)"]

## ► See Also

[Reference](#)

[PropertyBaseTOwner, TValue Class](#)

[PropertyBaseTOwner, TValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBase<TOwner, TValue> Constructor (TOwner, Action<Object, TValue, TValue>, TValue)

Initializes a new instance of the [PropertyBase<TOwner, TValue>](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public PropertyBase(
    TOwner owner,
    Action<Object, TValue, TValue> onValueChanged)

```

## Parameters

*owner*

Type: [TOwner](#)

[Missing <param name="owner"/> documentation for  
"M:W.PropertyBase`2.#ctor(`0,System.Action{System.Object,`1,`1})"]

*onValueChanged*

Type: [SystemAction<Object, TValue, TValue>](#)

[Missing <param name="onValueChanged"/> documentation for  
"M:W.PropertyBase`2.#ctor(`0,System.Action{System.Object,`1,`1})"]

## ► See Also

## Reference

[PropertyBaseTOwner, TValue Class](#)

[PropertyBaseTOwner, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValue Constructor (TOwner, TValue)

Initializes a new instance of the [PropertyBaseTOwner, TValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public PropertyBase(  
    TOwner owner,  
    TValue defaultValue  
)
```

[Copy](#)

### Parameters

*owner*

Type: [TOwner](#)

[Missing <param name="owner"/> documentation for  
"M:W.PropertyBase`2.#ctor(`0,`1)"]

*defaultValue*

Type: [TValue](#)

[Missing <param name="defaultValue"/> documentation for  
"M:W.PropertyBase`2.#ctor(`0,`1)"]

## ► See Also

[Reference](#)

[PropertyBaseTOwner, TValue Class](#)

[PropertyBaseTOwner, TValue Overload](#)

## [W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValue Constructor (TValue, ActionObject, TValue, TValue)

Initializes a new instance of the [PropertyBaseTOwner, TValue](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public PropertyBase(
    TValue defaultValue,
    Action<Object, TValue, TValue> onValueChanged)

```

### Parameters

*defaultValue*

Type: [TValue](#)

[Missing <param name="defaultValue"/> documentation for  
"M:W.PropertyBase`2.#ctor(`1,System.Action{System.Object,`1,`1})"]

*onValueChanged*

Type: [SystemActionObject, TValue, TValue](#)

[Missing <param name="onValueChanged"/> documentation for  
"M:W.PropertyBase`2.#ctor(`1,System.Action{System.Object,`1,`1})"]

## ► See Also

## Reference

[PropertyBaseTOwner, TValue Class](#)

[PropertyBaseTOwner, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBase<TOwner, TValue> Constructor (TOwner, TValue, Action<Object, TValue, TValue>, TValue)

Initializes a new instance of the [PropertyBase<TOwner, TValue>](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public PropertyBase(
    TOwner owner,
    TValue defaultValue,
    Action<Object, TValue, TValue> onValueChanged,
)
◀ ▶
```

### Parameters

*owner*

Type: [TOwner](#)

[Missing <param name="owner"/> documentation for  
"M:W.PropertyBase`2.#ctor(`0,`1,`2,`3)"]

*defaultValue*

Type: [TValue](#)

[Missing <param name="defaultValue"/> documentation for  
"M:W.PropertyBase`2.#ctor(`0,`1,`2,`3)"]

*onValueChanged*

Type: [SystemAction<Object, TValue, TValue>](#)

[Missing <param name="onValueChanged"/> documentation for

"M:W.PropertyBase`2.#ctor(`0,`1,`System.Action{`System.Object,`1,`1})"]

## See Also

[Reference](#)

[PropertyBaseTOwner, TValue Class](#)

[PropertyBaseTOwner, TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValue Properties

The [PropertyBaseTOwner, TValue](#) generic type exposes the following members.

## Properties

	Name	Description
	<a href="#">DefaultValue</a>	Allows the programmer to assign a default value which can be reset via the <code>ResetToDefaultValue</code> method. This value does not have to be the initial value.
	<a href="#">IsDirty</a>	True if Value has changed since initialization or since the last call to <code>MarkAsClean</code>
	<a href="#">Owner</a>	The property owner
	<a href="#">Value</a>	Get or Set the value (Inherited from <a href="#">LockableSlimTValue</a> .)

[Top](#)

## See Also

### Reference

[PropertyBaseTOwner, TValue Class](#)  
[W Namespace](#)



## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValueDefaultValue Property

Allows the programmer to assign a default value which can be reset via the ResetToDefaultValue method. This value does not have to be the initial value.

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

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

Property Value

Type: [TValue](#)

## ► See Also

Reference

[PropertyBaseTOwner, TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValueIsDirty Property

True if Value has changed since initialization or since the last call to MarkAsClean

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

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

Property Value

Type: [Boolean](#)

Implements

[IPropertyIsDirty](#)

## ► See Also

Reference

[PropertyBaseTOwner, TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValueOwner Property

The property owner

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public TOwner Owner { get; protected set; }
```

Property Value

Type: [TOwner](#)

## ► See Also

Reference

[PropertyBaseTOwner, TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBase<TOwner, TValue> Methods

The [PropertyBase<TOwner, TValue>](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disposes the <a href="#">Lockable</a> interface. (Inherited from <a href="#">Lockable</a> )
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> )
<a href="#">Finalize</a>	Allows an object to try to perform other cleanup before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> )
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> )
<a href="#">GetState</a>	Retrieves the internal state of the <a href="#">ReaderWriterLockSlim</a> object. (Inherited from <a href="#">ReaderWriterLockSlim</a> )
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> )
<a href="#">GetValue</a>	Gets the underlying value. (Inherited from <a href="#">Lockable</a> )

💡	<a href="#">InformWaiters</a>	Informs those who a WaitForChanged tha (Inherited from <a href="#">LockableTValueOnV</a> )
💡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action f ReaderWriterLockSli (Inherited from <a href="#">Read</a> )
💡	<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function ReaderWriterLockSli (Inherited from <a href="#">Read</a> )
💡	<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action f ReaderWriterLockSli (Inherited from <a href="#">Read</a> )
💡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function ReaderWriterLockSli (Inherited from <a href="#">Read</a> )
💡	<a href="#">LoadValue</a>	Sets Value without r
💡	<a href="#">MemberwiseClone</a>	Creates a shallow cc (Inherited from <a href="#">Object</a> )
💡	<a href="#">OnPropertyChanged</a>	Calls RaisePropertyC PropertyChanged ev (Overrides <a href="#">PropertySlimTValueOnP</a> )
💡	<a href="#">OnPropertyChanging</a>	Calls RaisePropertyC PropertyChanging ev (Overrides <a href="#">PropertySlimTValueOnP</a> )
💡	<a href="#">OnValueChanged</a>	Calls RaiseValueCh ValueChanged event (Overrides <a href="#">LockableTValueOnV</a> )

`TValue, TValue).`

💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <code>PropertyChanged</code> event. (Inherited from <code>PropertyChangedBase</code> )
💡	<a href="#">RaiseOnPropertyChanging</a>	Raises the <code>PropertyChanging</code> event. (Inherited from <code>PropertyChangedBase</code> )
💡	<a href="#">RaiseValueChanged</a>	Raises the <code>ValueChanged</code> event. (Inherited from <code>LockableObject</code> )
💡	<a href="#">ResetDefaultValue</a>	Resets the value to its <code>DefaultValue</code> .
💡	<a href="#">SetState</a>	Sets the internal state of the <code>ReaderWriterLockSlim</code> . (Inherited from <code>ReaderWriterLockSlim</code> )
💡	<a href="#">SetValue</a>	Calls <code>OnPropertyChanged</code> and <code>OnValueChanged</code> . (Inherited from <code>PropertyChangedBase</code> )
💡	<a href="#">ToString</a>	Returns a string that identifies the object. (Inherited from <code>Object</code> )
💡	<a href="#">WaitForValueChanged</a>	Allows the caller to block until the value changes. (Inherited from <code>LockableObject</code> )

[Top](#)

## ► Extension Methods

Name	Description
💡 	<a href="#">AsTType</a> Use Generic syntax for the <code>as</code> operator. (Defined by <a href="#">AsExtensions</a> .)

	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of

each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---

 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## ▲ See Also

Reference

[PropertyBaseTOwner, TValue Class](#)  
[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValueLoadValue Method

Sets Value without raising notification events

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public void LoadValue(  
    TValue value  
)
```

[Copy](#)

## Parameters

*value*

Type: [TValue](#)

[Missing <param name="value"/> documentation for  
"M:W.PropertyBase`2.LoadValue(`1)"]

## ► See Also

[Reference](#)

[PropertyBaseTOwner, TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValueOnPropertyChanged Method

Calls RaisePropertyChanged to raise the PropertyChanged event

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
protected override void OnPropertyChanged(
    string propertyName = null
)
```

### Parameters

*propertyName* **(Optional)**

Type: [System.String](#)

The name of the caller (the property which changed)

## ► See Also

### Reference

[PropertyBaseTOwner, TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValueOnPropertyChanging Method

Calls RaisePropertyChanging to raise the PropertyChanging event

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
protected override void OnPropertyChanged(
    string propertyName = null
)
```

### Parameters

*propertyName* **(Optional)**

Type: [System.String](#)

The name of the caller (the property which changed)

## ► See Also

### Reference

[PropertyBaseTOwner, TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBase<TOwner>. TValueOnValueChanged Method

Calls RaiseValueChanged to raise the ValueChanged event

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
protected override void OnValueChanged(  
    Object sender,  
    TValue oldValue,  
    TValue newValue  
)
```

[Copy](#)

## Parameters

*sender*

Type: [SystemObject](#)

The property owner

*oldValue*

Type: [TValue](#)

The previous value

*newValue*

Type: [TValue](#)

The current value

## ► See Also

Reference

PropertyBaseTOwner, TValue Class  
W Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValueResetDefaultValue Method

Resets the Value to the value provided by DefaultValue

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public void ResetDefaultValue(  
    bool raise  
)
```

### Parameters

*raise*

Type: [SystemBoolean](#)

If True, raise OnPropertyChanged event and call the OnValueChanged callback

## ► See Also

[Reference](#)

[PropertyBaseTOwner, TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValue Events

The [PropertyBaseTOwner, TValue](#) generic type exposes the following members.

## Events

	Name	Description
⚡	<a href="#">PropertyChanged</a>	(Inherited from <a href="#">PropertySlimTValue</a> .)
⚡	<a href="#">PropertyChanging</a>	(Inherited from <a href="#">PropertySlimTValue</a> .)
⚡	<a href="#">ValueChanged</a>	Raised when the value has changed (Inherited from <a href="#">LockableTValue</a> .)

[Top](#)

## See Also

[Reference](#)

[PropertyBaseTOwner, TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyBaseTOwner, TValue Fields

The [PropertyBaseTOwner, TValue](#) generic type exposes the following members.

## Fields

	Name	Description
	<a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from <a href="#">ReaderWriterLockerTState</a> .)
	<a href="#">State</a>	The internal state (Inherited from <a href="#">ReaderWriterLockerTState</a> .)

[Top](#)

## See Also

[Reference](#)

[PropertyBaseTOwner, TValue Class](#)  
[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyChangedNotifier Class

This is a base class for supporting INotifyPropertyChanged

## ► Inheritance Hierarchy

```
SystemObject WPropertyChangedNotifier  
WPropertyHostNotifier
```

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public abstract class PropertyChangedNotifier : ]  
[< >]
```

The [PropertyChangedNotifier](#) type exposes the following members.

## ► Constructors

	Name	Description
💡	<a href="#">PropertyChangedNotifier</a>	Initializes a new instance of the <a href="#">PropertyChangedNotifier</a> class

[Top](#)

## ► Methods

Name	Description
 <a href="#">Equals</a>	<p>Determines whether the specified object is equal to the current object.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
 <a href="#">Finalize</a>	<p>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
 <a href="#">GetHashCode</a>	<p>Serves as the default hash function.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
 <a href="#">GetType</a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
 <a href="#">GetValue</a>	<p>Override this method to provide Get functionality</p>
 <a href="#">MemberwiseClone</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
 <a href="#">OnPropertyChanged</a>	<p>Calls <a href="#">RaisePropertyChanged</a> to raise the <a href="#">PropertyChanged</a> event</p>
 <a href="#">RaiseOnPropertyChanged</a>	<p>Raises the <a href="#">PropertyChanged</a></p>

---

		event
💡	<a href="#">SetValue</a>	Calls <code>OnPropertyChanged</code> . This method does not make assignments. Override this method to make assignments.
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## Events

	Name	Description
⚡	<a href="#">PropertyChanged</a>	Raised when a property changes

---

[Top](#)

## Extension Methods

	Name	Description
⬇️ ⚡	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
⬇️	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class

which has  
IOwnedProperty  
members  
(Defined by  
[PropertyHostExtensions](#).)

---

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock

(Defined by  
[MonitorExtensions](#).)

---



### [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets it's `IsDirty` flag to false

(Defined by  
[PropertyHostExtensions](#).)

---



### [Unlock](#)

Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyChangedNotifier Constructor

Initializes a new instance of the [PropertyChangedNotifier](#) class

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected PropertyChangedNotifier()
```

## ► See Also

[Reference](#)

[PropertyChangedNotifier Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyChangedNotifier Methods

The [PropertyChangedNotifier](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetValue</a>	Override this method to provide Get functionality

	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#"> OnPropertyChanged</a>	Calls <a href="#">RaisePropertyChanged</a> to raise the <a href="#">PropertyChanged</a> event
	<a href="#"> RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event
	<a href="#"> SetValue</a>	Calls <a href="#"> OnPropertyChanged</a> . This method does not make assignments. Override this method to make assignments.
	<a href="#"> ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#"> InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method

should be called in the constructor of any class which has `IOwnedProperty` members  
(Defined by [PropertyHostExtensions](#).)

---

	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the <code>IsDirty</code> value of each field and property of type <code>IProperty</code> (Defined by <a href="#">PropertyHostExtensions</a> .)

 Lock	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 MarkAsClean	Scans each field and property of type <code>IProperty</code> and sets its <code>IsDirty</code> flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 Unlock	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## ► See Also

[Reference](#)

[PropertyChangedNotifier Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyChangedNotifier.GetValue Method

Override this method to provide Get functionality

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected virtual Object GetValue()
```

Return Value

Type: [Object](#)

Unless overridden, this function will always return null

## ► See Also

Reference

[PropertyChangedNotifier Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyChangedNotifierOnProperty Method

Calls `RaisePropertyChanged` to raise the `PropertyChanged` event

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected virtual void OnPropertyChanged(  
    string propertyName = null  
)
```

### Parameters

*propertyName* (**Optional**)

Type: [SystemString](#)

The name of the caller (the property which changed)

## ► See Also

[Reference](#)

[PropertyChangedNotifier Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyChangedNotifier.RaiseOnPropertyChanged Method

Raises the PropertyChanged event

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
protected virtual void RaisePropertyChanged(  
    Object sender,  
    string propertyName  
)
```

[Copy](#)

### Parameters

*sender*

Type: [SystemObject](#)

The sender is the owner of the property

*propertyName*

Type: [SystemString](#)

The name of the caller (the property which changed)

## ► See Also

[Reference](#)

[PropertyChangedNotifier Class](#)

[W Namespace](#)



## Tungsten

W<sup>74</sup>

# PropertyChangedNotifier.SetValue Method

Calls `OnPropertyChanged`. This method does not make assignments. Override this method to make assignments.

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
protected virtual void SetValue(  
    Object value,  
    string propertyName = "")  
)
```

[Copy](#)

## Parameters

*value*

Type: [SystemObject](#)

The new value

*propertyName* (Optional)

Type: [SystemString](#)

The name of the caller (the property being set)

## ► See Also

[Reference](#)

[PropertyChangedNotifier Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyChangedNotifier Events

The [PropertyChangedNotifier](#) type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">PropertyChanged</a>	Raised when a property changes

[Top](#)

## ▪ See Also

[Reference](#)

[PropertyChangedNotifier Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyChangedNotifier.PropertyChangedEvent

Raised when a property changes

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public event PropertyChangedEventHandler PropertyChanged
```

Value

Type: [System.ComponentModel.PropertyChangedEventHandler](#)

Implements

[INotifyPropertyChanged](#)

## ► See Also

Reference

[PropertyChangedNotifier Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyHost Class

Provides a base class to automate the IsDirty, MarkAsClean and InitializeProperties functionality Note that this class does not support INotifyPropertyChanged and is not intended to host owned properties (though nothing prevents you from doing so)

## ► Inheritance Hierarchy

[SystemObject](#) [WPropertyHost](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class PropertyHost
```

The [PropertyHost](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PropertyHost</a>	Calls PropertyHostExtensions.InitializeProperties so you don't have to

[Top](#)

## ► Properties

	Name	Description
	<a href="#">IsDirty</a>	Finds all Properties and checks their IsDirty flag

[Top](#)

## ▲ Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MarkAsClean</a>	Uses reflection to find all Properties and mark them as clean (call <code>Property.MarkAsClean()</code> )
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> .

(Inherited from [Object](#).)

 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
--	---

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ◀ See Also

Reference

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHost Constructor

Calls PropertyHostExtensions.InitializeProperties so you don't have to

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public PropertyHost()
```

## ► See Also

Reference

[PropertyHost Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHost Properties

The [PropertyHost](#) type exposes the following members.

## ► Properties

Name	Description
 <a href="#">IsDirty</a>	Finds all Properties and checks their IsDirty flag

[Top](#)

## ► See Also

[Reference](#)

[PropertyHost Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostIsDirty Property

Finds all Properties and checks their IsDirty flag

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public bool IsDirty { get; }
```

Return Value

Type: [Boolean](#)

True if any Property's IsDirty flag is true. Otherwise false.

## ► See Also

Reference

[PropertyHost Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHost Methods

The [PropertyHost](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MarkAsClean</a>	Uses reflection to find all Properties and mark them as clean (call <code>Property.MarkAsClean()</code> )
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> .

(Inherited from [Object](#).)

 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
--	---

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

Reference

[PropertyHost Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostMarkAsClean Method

Uses reflection to find all Properties and mark them as clean (call `Property.MarkAsClean()`)

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

```
public void MarkAsClean()
```

[Copy](#)

## ► See Also

[Reference](#)

[PropertyHost Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyHostExtensions Class

Extension methods related to W.PropertyHost or any class which has multiple W.Property members

## ► Inheritance Hierarchy

[SystemObject](#) [WPropertyHostExtensions](#)

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static class PropertyHostExtensions
```

The [PropertyHostExtensions](#) type exposes the following members.

## ► Methods

	Name	Description
 	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members
 	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty



## MarkAsClean

Scans each field and property of type `IProperty` and sets it's `IsDirty` flag to false

---

[Top](#)

## See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostExtensions Methods

The [PropertyHostExtensions](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members
 	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type <a href="#">IProperty</a>
 	<a href="#">MarkAsClean</a>	Scans each field and property of type <a href="#">IProperty</a> and sets it's IsDirty flag to false

[Top](#)

## ▪ See Also

[Reference](#)

[PropertyHostExtensions Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyHostExtensionsInitializeProperties Method

Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members.

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static void InitializeProperties(  
    this Object this  
)
```

### Parameters

*this*

Type: [System.Object](#)

The object on which to find and initialize properties

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

Reference

[PropertyHostExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostExtensionsIsDirty Method

Scans the IsDirty value of each field and property of type IProperty

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public static bool IsDirty(  
    this Object this  
)
```

[Copy](#)

### Parameters

*this*

Type: [SystemObject](#)

The object on which to inspect for dirty properties

### Return Value

Type: [Boolean](#)

True if any IProperty member's IsDirty value is true, otherwise false

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ▲ See Also

Reference

[PropertyHostExtensions Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostExtensionsMarkAsClean Method

Scans each field and property of type `IProperty` and sets its `IsDirty` flag to false

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static void MarkAsClean(  
    this Object this  
)
```

## Parameters

*this*

Type: [System.Object](#)

The object on which to mark all properties as clean

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

### Reference

## PropertyHostExtensions Class W Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostNotifier Class

Provides a base class to automate the IsDirty, MarkAsClean and InitializeProperties functionality Note that this class inherits PropertyChangedNotifier for INotifyPropertyChanged support

## ► Inheritance Hierarchy

SystemObject WPropertyChangedNotifier  
WPropertyChangedNotifier

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public class PropertyHostNotifier : PropertyChangedNotifi
◀ ▶
```

The [PropertyHostNotifier](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PropertyHostNotifier</a>	Calls PropertyHostMethods.InitializeProperties so you don't have to

[Top](#)

## ► Properties

Name	Description
 <a href="#">IsDirty</a>	Finds all Properties and checks their IsDirty flag

[Top](#)

## ◀ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetValue</a>	Override this method to provide Get functionality (Inherited from <a href="#">PropertyChangedNotifier</a> .)
 <a href="#">MarkAsClean</a>	Uses reflection to find all

		Properties and mark them as clean (call <code>Property.MarkAsClean()</code> )
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <code>Object</code> . (Inherited from <code>Object</code> .)
💡	<a href="#">OnPropertyChanged</a>	Calls <code>RaisePropertyChanged</code> to raise the <code>PropertyChanged</code> event (Inherited from <code>PropertyChangedNotifier</code> .)
💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <code>PropertyChanged</code> event (Inherited from <code>PropertyChangedNotifier</code> .)
💡	<a href="#">SetValue</a>	Calls <code>OnPropertyChanged</code> . This method does not make assignments. Override this method to make assignments. (Inherited from <code>PropertyChangedNotifier</code> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <code>Object</code> .)

[Top](#)

## Events

Name	Description
⚡	

[PropertyChanged](#) Raised when a property changes  
(Inherited from [PropertyChangedNotifier](#).)

[Top](#)

## ► Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

---

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## ▲ See Also

Reference

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostNotifier Constructor

Calls PropertyHostMethods.InitializeProperties so you don't have to

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public PropertyHostNotifier()
```

## ► See Also

Reference

[PropertyHostNotifier Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostNotifier Properties

The [PropertyHostNotifier](#) type exposes the following members.

## ► Properties

Name	Description
 <a href="#">IsDirty</a>	Finds all Properties and checks their IsDirty flag

[Top](#)

## ► See Also

Reference

[PropertyHostNotifier Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostNotifierIsDirty Property

Finds all Properties and checks their IsDirty flag

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public bool IsDirty { get; }
```

Return Value

Type: [Boolean](#)

True if any Property's IsDirty flag is true. Otherwise false.

## ► See Also

[Reference](#)

[PropertyHostNotifier Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyHostNotifier Methods

The [PropertyHostNotifier](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
<a href="#">GetValue</a>	Override this method to provide Get functionality (Inherited from <a href="#">PropertyChangedNotifier</a> .)
<a href="#">MarkAsClean</a>	Uses reflection to find all

		Properties and mark them as clean (call <code>Property.MarkAsClean()</code> )
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <code>Object</code> . (Inherited from <code>Object</code> .)
💡	<a href="#">OnPropertyChanged</a>	Calls <code>RaisePropertyChanged</code> to raise the <code>PropertyChanged</code> event (Inherited from <code>PropertyChangedNotifier</code> .)
💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <code>PropertyChanged</code> event (Inherited from <code>PropertyChangedNotifier</code> .)
💡	<a href="#">SetValue</a>	Calls <code>OnPropertyChanged</code> . This method does not make assignments. Override this method to make assignments. (Inherited from <code>PropertyChangedNotifier</code> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <code>Object</code> .)

[Top](#)

## Extension Methods

Name	Description
💡	

	<b>AsTType</b>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
▼	<b>InitializeProperties</b>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
▼	<b>InLock(Action)</b>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
▼	<b>InLockTType(FuncTType)</b>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
▼	<b>InLockAsync(Action)</b>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
▼	<b>InLockAsyncTType(FuncTType)</b>	Overloaded. Asynchronously performs the action in a Monitor

`lock`  
(Defined by  
[MonitorExtensions](#).)

---

 [IsDirty](#) Scans the `IsDirty` value of each field and property of type `IProperty`  
(Defined by  
[PropertyHostExtensions](#).)

---

 [Lock](#) Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

 [MarkAsClean](#) Scans each field and property of type `IProperty` and sets its `IsDirty` flag to false  
(Defined by  
[PropertyHostExtensions](#).)

---

 [Unlock](#) Performs a Monitor unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[PropertyHostNotifier Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertyHostNotifierMarkAsClean Method

Uses reflection to find all Properties and mark them as clean (call Property.MarkAsClean())

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

```
public void MarkAsClean()
```

[Copy](#)

## ► See Also

Reference

[PropertyHostNotifier Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHostNotifier Events

The [PropertyHostNotifier](#) type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">PropertyChanged</a>	Raised when a property changes (Inherited from <a href="#">PropertyChangedNotifier</a> .)

[Top](#)

## ▪ See Also

[Reference](#)

[PropertyHostNotifier Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Class

PropertySlim extends W.Lockable by adding support for  
INotifyPropertyChanged

## ► Inheritance Hierarchy

```
SystemObject W.Threading.LockersReaderWriterLocker TValue
WLockableSlim TValue
WLockable TValue
WPropertySlim TValue
WPropertyBase TOwner, TValue
```

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public class PropertySlim<TValue> : Lockable<TValue>
    INotifyPropertyChanging, INotifyPropertyChanged
```

### Type Parameters

*TValue*

The Type of Value

The [PropertySlim TValue](#) type exposes the following members.

## ► Constructors

Name	Description
------	-------------

	<a href="#">PropertySlim TValue</a>	Construct a new PropertySlim with a default initial value
	<a href="#">PropertySlim TValue(ActionObject, TValue, TValue)</a>	Concstrcut a new PropertySlim
	<a href="#">PropertySlim TValue(TValue)</a>	Concstrcut a new PropertySlim
	<a href="#">PropertySlim TValue(TValue, ActionObject, TValue, TValue)</a>	Concstrcut a new PropertySlim

[Top](#)

## Properties

Name	Description
 <a href="#">Value</a>	Get or Set the value (Inherited from <a href="#">LockableSlim TValue</a> .)

[Top](#)

## Methods

Name	Description
 <a href="#">Dispose</a>	Disposes the Lockak releases resources (Inherited from <a href="#">Lock</a> )
 <a href="#">Equals</a>	Determines whether

object is equal to the object.  
(Inherited from [Object](#))

💡	<a href="#">Finalize</a>	Allows an object to tidy up resources and perform cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> )
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> )
💡	<a href="#">GetState</a>	Retrieves the internal state within a ReaderWriterLockSlim instance. (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the instance. (Inherited from <a href="#">Object</a> )
💡	<a href="#">GetValue</a>	Gets the underlying value. (Inherited from <a href="#">LockableSlim TValue</a> )
💡	<a href="#">InformWaiters</a>	Informs those who are waiting on a <a href="#">WaitForChanged</a> that the value has changed. (Inherited from <a href="#">LockableSlim</a> )
💡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action function within a lock. (Inherited from <a href="#">ReaderWriterLockSlim</a> )
💡	<a href="#">InLockTValue(LockTypeEnum, ActionTValue)</a>	Executes a function that returns a value within a lock. (Inherited from <a href="#">ReaderWriterLocker</a> )

	<code>FuncTState, TValue)</code>	ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<code>InLockAsync(LockTypeEnum, ActionTState)</code>	Executes an action f ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )
≡	<code>InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</code>	Executes a function t ReaderWriterLockSli (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<code>MemberwiseClone</code>	Creates a shallow cc current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> )
💡	<code>OnPropertyChanged</code>	Calls <code>RaisePropertyC</code> raise the <code>PropertyChanged</code> event
💡	<code>OnPropertyChanging</code>	Calls <code>RaisePropertyC</code> raise the <code>PropertyChanging</code> event
💡	<code>OnValueChanged</code>	Calls <code>RaiseValueChange</code> the <code>ValueChanged</code> event (Inherited from <a href="#">LockBase</a> )
💡	<code>RaiseOnPropertyChanged</code>	Raises the <code>PropertyChanged</code> event
💡	<code>RaiseOnPropertyChanging</code>	Raises the <code>PropertyChanging</code> event
💡	<code>RaiseValueChanged</code>	Raises the <code>ValueChanged</code> event (Inherited from <a href="#">LockBase</a> )
≡	<code>SetState</code>	Sets the internal stat a ReaderWriterLockSli

---

		(Inherited from <a href="#">ReaderWriterLocker</a> )
 <a href="#">SetValue</a>	Calls <a href="#">OnPropertyChanged</a> when the value is assigned (Overrides <a href="#">LockableTValue.SetValue</a> )	
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> )	
 <a href="#">WaitForValueChanged</a>	Allows the caller to be notified when value changes (Inherited from <a href="#">LockableTValue</a> )	

---

[Top](#)

## Events

	Name	Description
 <a href="#">PropertyChanged</a>		
 <a href="#">PropertyChanging</a>		
 <a href="#">ValueChanged</a>	Raised when the value has changed (Inherited from <a href="#">LockableTValue</a> .)	

---

[Top](#)

## Operators

	Name	Description
 <a href="#">(TValue to S)</a>		Implicit conversion from

---

	<a href="#">PropertySlim TValue</a> )	TValue to PropertySlim< TValue >
 	<a href="#">(PropertySlim TValue to TValue)</a>	Implicit conversion from PropertySlim< TValue > to TValue

---

[Top](#)

## Fields

	Name	Description
 	<a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from <a href="#">ReaderWriterLockerTState.</a> )
 	<a href="#">State</a>	The internal state (Inherited from <a href="#">ReaderWriterLockerTState.</a> )

---

[Top](#)

## Extension Methods

	Name	Description
 	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner" This method should be called in the

constructor of any class  
which has  
`IOwnedProperty`  
members  
(Defined by  
[PropertyHostExtensions](#).)

---

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the <code>IsDirty</code> value of each field and property of type <code>IProperty</code> (Defined by <a href="#">PropertyHostExtensions</a> .)



## Lock

Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)



## MarkAsClean

Scans each field and  
property of type IProperty  
and sets it's IsDirty flag to  
false  
(Defined by  
[PropertyHostExtensions](#).)



## Unlock

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Constructor

## ► Overload List

	Name	Description
≡	<a href="#">PropertySlimTValue</a>	Construct a new PropertySlim with a default initial value
≡	<a href="#">PropertySlimTValue(ActionObject, TValue, TValue)</a>	Concstruct a new PropertySlim
≡	<a href="#">PropertySlimTValue(TValue)</a>	Concstruct a new PropertySlim
≡	<a href="#">PropertySlimTValue(TValue, ActionObject, TValue, TValue)</a>	Concstruct a new PropertySlim

[Top](#)

## ► See Also

Reference

[PropertySlimTValue Class](#)  
[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Constructor

Construct a new PropertySlim with a default initial value

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public PropertySlim()
```

## ► See Also

Reference

[PropertySlim TValue Class](#)

[PropertySlim TValue Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Constructor (ActionObject, TValue, TValue)

Construct a new PropertySlim

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public PropertySlim(
    Action<Object, TValue, TValue> onChanged)
{}
```

### Parameters

*onValueChanged*

Type: [SystemActionObject, TValue, TValue](#)

Called when the value is changed

## ► See Also

Reference

[PropertySlim TValue Class](#)

[PropertySlim TValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Constructor (TValue)

Construct a new PropertySlim

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
public PropertySlim(  
    TValue initialValue  
)
```

[Copy](#)

### Parameters

*initialValue*

Type: [TValue](#)

The initial value

## ► See Also

### Reference

[PropertySlimTValue Class](#)

[PropertySlimTValue Overload](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Constructor (TValue, ActionObject, TValue, TValue)

Concstruct a new PropertySlim

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public PropertySlim(
    TValue initialValue,
    Action<Object, TValue, TValue> onValueChanged)

```

## Parameters

*initialValue*

Type: [TValue](#)

The initial value

*onValueChanged*

Type: [SystemActionObject](#), [TValue](#), [TValue](#)

Called when the value is changed

## ► See Also

Reference

[PropertySlimTValue Class](#)

## PropertySlim TValue Overload W Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Properties

The [PropertySlim TValue](#) generic type exposes the following members.

## ► Properties

Name	Description
 <a href="#">Value</a>	Get or Set the value (Inherited from <a href="#">LockableSlim TValue</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PropertySlim TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Methods

The [PropertySlim TValue](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disposes the Lockable releases resources (Inherited from <a href="#">Lockable</a> )
<a href="#">Equals</a>	Determines whether object is equal to the object. (Inherited from <a href="#">Object</a> )
<a href="#">Finalize</a>	Allows an object to tidy up resources and perform cleanup operations to be reclaimed by garbage collection. (Inherited from <a href="#">Object</a> )
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> )
<a href="#">GetState</a>	Retrieves the internal state within a ReaderWriterLock. (Inherited from <a href="#">ReaderWriterLocker</a> )
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the instance. (Inherited from <a href="#">Object</a> )

---

💡	<a href="#">GetValue</a>	Gets the underlying value. (Inherited from <a href="#">LockableSlim TValue</a> )
💡	<a href="#">InformWaiters</a>	Informs those who are waiting on a <a href="#">WaitForChanged</a> that the value has changed. (Inherited from <a href="#">LockableSlim</a> )
💡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action function on a lock held by the current <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value on a lock held by the current <a href="#">ReaderWriterLockSlim</a> . (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action function on a lock held by the current <a href="#">ReaderWriterLockSlim</a> in a background thread. (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function that returns a value on a lock held by the current <a href="#">ReaderWriterLockSlim</a> in a background thread. (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> )
💡	<a href="#">OnPropertyChanged</a>	Calls <a href="#">RaisePropertyChanged</a> to raise the <a href="#">PropertyChanged</a> event.
💡	<a href="#">OnPropertyChanging</a>	Calls <a href="#">RaisePropertyChanging</a> to raise the <a href="#">PropertyChanging</a> event.

---

💡	<a href="#">OnValueChanged</a>	Calls <code>RaiseValueChange</code> when the <code>ValueChanged</code> event is raised. (Inherited from <a href="#">LockableTValueSetV</a> )
💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <code>PropertyChanged</code> event.
💡	<a href="#">RaiseOnPropertyChanging</a>	Raises the <code>PropertyChanging</code> event.
💡	<a href="#">RaiseValueChanged</a>	Raises the <code>ValueChanged</code> event. (Inherited from <a href="#">LockableTValueSetV</a> )
💡	<a href="#">SetState</a>	Sets the internal state under a <code>ReaderWriterLock</code> . (Inherited from <a href="#">ReaderWriterLocker</a> )
💡	<a href="#">SetValue</a>	Calls <code>OnPropertyChanged</code> when the value assignment occurs. (Overrides <a href="#">LockableTValueSetV.SetValue</a> )
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> )
💡	<a href="#">WaitForValueChanged</a>	Allows the caller to block until value changes occur. (Inherited from <a href="#">LockableTValueSetV</a> )

[Top](#)

## Extension Methods

Name	Description
💡  <a href="#">AsTType</a>	Use Generic syntax for the as operator.

(Defined by  
[AsExtensions](#).)

---

InitializeProperties

Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has [IOwnedProperty](#) members

(Defined by  
[PropertyHostExtensions](#).)

---

InLock(Action)

Overloaded.  
Performs the action in a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

InLockTType(FuncTType)

Overloaded.  
Performs the function in a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

InLockAsync(Action)

Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

InLockAsyncTType(FuncTType)

Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by

[MonitorExtensions.\)](#)



### [IsDirty](#)

Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions.](#))



### [Lock](#)

Performs a Monitor lock  
(Defined by [MonitorExtensions.](#))



### [MarkAsClean](#)

Scans each field and property of type IProperty and sets its IsDirty flag to false  
(Defined by [PropertyHostExtensions.](#))



### [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions.](#))

[Top](#)

## ▲ See Also

[Reference](#)

[PropertySlim TValue Class](#)

[W Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim TValueOnPropertyChanged Method

Calls RaisePropertyChanged to raise the PropertyChanged event

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected virtual void OnPropertyChanged(  
    string propertyName = null  
)
```

### Parameters

*propertyName* (**Optional**)

Type: [SystemString](#)

The name of the caller (the property which changed)

## ► See Also

[Reference](#)

[PropertySlimTValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertySlim TValueOnPropertyChanged Method

Calls RaisePropertyChanging to raise the PropertyChanging event

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected virtual void OnPropertyChanged(  
    string propertyName = null  
)
```

### Parameters

*propertyName* (**Optional**)

Type: [SystemString](#)

The name of the caller (the property which changed)

## ► See Also

[Reference](#)

[PropertySlimTValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertySlimTValueRaiseOnProperty Method

Raises the PropertyChanged event

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected void RaisePropertyChanged(  
    Object sender,  
    string propertyName  
)
```

## Parameters

*sender*

Type: [SystemObject](#)

The sender is the owner of the property

*propertyName*

Type: [SystemString](#)

The name of the caller (the property which changed)

## ► See Also

Reference

[PropertySlimTValue Class](#)

[W Namespace](#)



## Tungsten

W<sup>74</sup>

# PropertySlimTValueRaiseOnProperty Method

Raises the PropertyChanging event

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected void RaiseOnPropertyChanged(  
    Object sender,  
    string propertyName  
)
```

### Parameters

*sender*

Type: [SystemObject](#)

The sender is the owner of the property

*propertyName*

Type: [SystemString](#)

The name of the caller (the property which changing)

## ► See Also

Reference

[PropertySlimTValue Class](#)

[W Namespace](#)



## Tungsten

W<sup>74</sup>

# PropertySlim TValue SetValue Method

Calls OnPropertyChanged on assignment

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C#  
protected override void SetValue(  
    TValue value  
)
```

[Copy](#)

### Parameters

*value*

Type: [TValue](#)

The new value

## ► See Also

[Reference](#)

[PropertySlimTValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Events

The [PropertySlim TValue](#) generic type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">PropertyChanged</a>	
 <a href="#">PropertyChanging</a>	
 <a href="#">ValueChanged</a>	Raised when the value has changed (Inherited from <a href="#">Lockable TValue</a> .)

[Top](#)

## ▪ See Also

Reference

[PropertySlim TValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim`1.PropertyChanged Event

[Missing <summary> documentation for "E:W.PropertySlim`1.PropertyChanged"]

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public event PropertyChangedEventHandler PropertyChanged
```

Value

Type: [System.ComponentModel.PropertyChangedEventHandler](#)

Implements

[INotifyPropertyChanged](#)

## ► See Also

Reference

[PropertySlim`1 Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertySlim TValuePropertyChanging Event

[Missing <summary> documentation for "E:W.PropertySlim`1.PropertyChanging"]

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public event PropertyChangingEventHandler Propert
```

Value

Type: [System.ComponentModel.PropertyChangingEventHandler](#)

Implements

[INotifyPropertyChanging](#)[PropertyChanging](#)

## ► See Also

Reference

[PropertySlim TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Type Conversions

## Operators

	Name	Description
	<a href="#">(TValue to PropertySlimTValue)</a>	Implicit conversion from TValue to PropertySlim<TValue>
	<a href="#">(PropertySlimTValue to TValue)</a>	Implicit conversion from PropertySlim<TValue> to TValue

[Top](#)

## See Also

Reference

[PropertySlimTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Conversion Operators

## ▪ Overload List

	Name	Description
	<a href="#">(TValue to PropertySlimTValue)</a>	Implicit conversion from TValue to PropertySlim<TValue>
	<a href="#">(PropertySlimTValue to TValue)</a>	Implicit conversion from PropertySlim<TValue> to TValue

[Top](#)

## ▪ See Also

Reference

[PropertySlimTValue Class](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Conversion (TValue to PropertySlim TValue)

Implicit conversion from TValue to PropertySlim<TValue>

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static implicit operator PropertySlim<TValue>
    TValue value
)
```

### Parameters

*value*

Type: [TValue](#)

The value from which to create a new PropertySlim<TValue>

### Return Value

Type: [PropertySlim TValue](#)

[Missing <returns> documentation for  
"M:W.PropertySlim`1.op\_Implicit(`0)~W.PropertySlim{`0}"]

## ► See Also

Reference

[PropertySlim TValue Class](#)

[Overload](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim $TValue$ Conversion (PropertySlim $TValue$ to $TValue$ )

Implicit conversion from PropertySlim< $TValue$ > to  $TValue$

**Namespace:** [W](#)

**Assembly:** Tungsten.Property (in Tungsten.Property.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static implicit operator TValue (
    PropertySlim<TValue> property
)
```

### Parameters

*property*

Type: [WPropertySlim \$TValue\$](#)

The PropertySlim< $TValue$ > from which to obtain the value

### Return Value

Type: [TValue](#)

[Missing <returns> documentation for  
"M:W.PropertySlim`1.op\_Implicit(W.PropertySlim{`0})~`0"]

## ► See Also

### Reference

[PropertySlim \$TValue\$  Class](#)

[Overload](#)

[W Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertySlim TValue Fields

The [PropertySlim TValue](#) generic type exposes the following members.

## ► Fields

	Name	Description
💡	<a href="#">Locker</a>	The ReaderWriterLocker used to access the State (Inherited from <a href="#">ReaderWriterLockerTState.</a> )
💡	<a href="#">State</a>	The internal state (Inherited from <a href="#">ReaderWriterLockerTState.</a> )

[Top](#)

## ► See Also

[Reference](#)

[PropertySlim TValue Class](#)

[W Namespace](#)

## Tungsten

W<sup>74</sup>

# ValueChangedDelegate<TValue> Delegate

Raised when the value has changed

**Namespace:** [W](#)

**Assembly:** Tungsten.Lockable (in Tungsten.Lockable.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public delegate void ValueChangedDelegate<TValue>
    Object sender,
    TValue oldValue,
    TValue newValue
)
```

## Parameters

*sender*

Type: [SystemObject](#)

The object which raised the event

*oldValue*

Type: [TValue](#)

The old value

*newValue*

Type: [TValue](#)

The new value

## Type Parameters

*TValue*

[Missing <typeparam name="TValue"/> documentation for "T:W.ValueChangedDelegate`1"]

## ▲ See Also

[Reference](#)

[W Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# W.Domains Namespace

[Missing <summary> documentation for "N:W.Domains"]

## ► Classes

	Class	Description
	<a href="#">DomainLoader</a>	An AppDomain helper class which makes it easy to host reloadable AppDomains. Supports ShadowCopy.

## ► Interfaces

	Interface	Description
	<a href="#">IDomainLoader</a>	Defines the interface for a DomainLoader

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoader Class

An AppDomain helper class which makes it easy to host reloadable AppDomains. Supports ShadowCopy.

## ► Inheritance Hierarchy

[SystemObject](#) [W.DomainsDomainLoader](#)

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public class DomainLoader : IDomainLoader,  
    IDisposable
```

The [DomainLoader](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">DomainLoader(String, Boolean)</a>	Creates an AppDomain under the current AppDomain
≡	<a href="#">DomainLoader(String, String, Boolean)</a>	Creates an AppDomain under the current AppDomain

[Top](#)

## Properties

	Name	Description
	<a href="#">DomainName</a>	The name of the new AppDomain

[Top](#)

## Methods

	Name	Description
	<a href="#">Create(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.
	<a href="#">CreateTInterfaceType(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.
	<a href="#">Dispose</a>	Performs application-defined tasks

associated with freeing, releasing, or resetting unmanaged resources.

 <a href="#">DoCallback</a>	Executes an action in the context of the hosted AppDomain
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Execute(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
 <a href="#">ExecuteTResult(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
 <a href="#">ExecuteStaticMethod(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain

≡	<a href="#">ExecuteStaticMethodTResult(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain
≡	<a href="#">Finalize</a>	Destructs the DomainLoader instance. Calls Dispose. (Overrides <a href="#">ObjectFinalize.</a> )
≡	<a href="#">GetDataTData</a>	Gets the value stored in the current application domain for the specified name
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object.</a> )
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object.</a> )
≡	<a href="#">Load</a>	Loads the dlls into the new AppDomain
≡	<a href="#">MemberwiseClone</a>	Creates a

shallow copy of  
the current  
[Object](#).  
(Inherited from  
[Object](#).)

 <a href="#">SetData</a>	Sets the value of the specified application domain property
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Unload</a>	Unloads the AppDomain and deletes files in the cache folder. The cache folder is where dlls are copied, and run, when using shadow copying.

[Top](#)

## ► Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for

the as operator.  
(Defined by  
[AsExtensions](#).)

---

 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

---

 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets it's IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ▲ See Also

[Reference](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# DomainLoader Constructor

## ► Overload List

	Name	Description
≡	<a href="#">DomainLoader(String, Boolean)</a>	Creates an AppDomain under the current AppDomain
≡	<a href="#">DomainLoader(String, String, Boolean)</a>	Creates an AppDomain under the current AppDomain

[Top](#)

## ► See Also

### Reference

[DomainLoader Class](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# DomainLoader Constructor (String, Boolean)

Creates an AppDomain under the current AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public DomainLoader(
    string relativeSubFolderForDomain,
    bool useShadowCopy = false
)
```

### Parameters

*relativeSubFolderForDomain*

Type: [SystemString](#)

The relative path to the subfolder which will be the root folder for the new AppDomain

*useShadowCopy (Optional)*

Type: [SystemBoolean](#)

True to shadow copy files. This allows dlls to be added, removed or modified while the AppDomain is still loaded.

## ► See Also

[Reference](#)

[DomainLoader Class](#)

[DomainLoader Overload](#)

## W.Domains Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoader Constructor (String, String, Boolean)

Creates an AppDomain under the current AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public DomainLoader(
    string domainName,
    string relativeSubFolderForDomain,
    bool useShadowCopy = false
)
```

### Parameters

*domainName*

Type: [SystemString](#)

The name for the domain. If not assigned, or null or empty, a Guid is assigned.

*relativeSubFolderForDomain*

Type: [SystemString](#)

The relative path to the subfolder which will be the root folder for the new AppDomain

*useShadowCopy (Optional)*

Type: [SystemBoolean](#)

True to shadow copy files. This allows dlls to be added, removed or modified while the AppDomain is still loaded.

## ▲ See Also

Reference

[DomainLoader Class](#)

[DomainLoader Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoader Properties

The [DomainLoader](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">DomainName</a>	The name of the new AppDomain

[Top](#)

## ► See Also

[Reference](#)

[DomainLoader Class](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderDomainName Property

The name of the new AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public string DomainName { get; }
```

Property Value

Type: [String](#)

## ► See Also

Reference

[DomainLoader Class](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoader Methods

The [DomainLoader](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Create(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.
<a href="#">CreateTInterfaceType(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.
<a href="#">Dispose</a>	Performs application-defined tasks associated with freeing,

releasing, or  
resetting  
unmanaged  
resources.

≡	<a href="#">DoCallback</a>	Executes an action in the context of the hosted AppDomain
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Execute(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
≡	<a href="#">ExecuteTResult(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
≡	<a href="#">ExecuteStaticMethod(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain
≡	<a href="#">ExecuteStaticMethodTResult(String, String, Object)</a>	Executes a static method

on the  
specified type  
across the  
AppDomain

 <a href="#">Finalize</a>	Destructs the DomainLoader instance. Calls Dispose. (Overrides <a href="#">ObjectFinalize.</a> )
 <a href="#">GetDataTData</a>	Gets the value stored in the current application domain for the specified name
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">Load</a>	Loads the dlls into the new AppDomain
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current

[Object.](#)  
(Inherited from  
[Object.](#))

 <a href="#">SetData</a>	Sets the value of the specified application domain property
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object.</a> )
 <a href="#">Unload</a>	Unloads the AppDomain and deletes files in the cache folder. The cache folder is where dlls are copied, and run, when using shadow copying.

[Top](#)

## ► Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by

[AsExtensions.\)](#)

---

 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)



## IsDirty

Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)



## Lock

Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)



## MarkAsClean

Scans each field and property of type IProperty and sets it's IsDirty flag to false  
(Defined by [PropertyHostExtensions](#).)



## Unlock

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

[Top](#)

## See Also

### Reference

[DomainLoader Class](#)

[W.Domains Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderCreate Method

## ▪ Overload List

Name	Description
 <a href="#">CreateTInterfaceType(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.
 <a href="#">Create(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.

[Top](#)

## ▪ See Also

[Reference](#)

[DomainLoader Class](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# DomainLoaderCreate $T$ InterfaceType Method (String)

Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public TInterfaceType Create<TInterfaceType>(  
    string typeName  
)
```

### Parameters

*typeName*

Type: [System.String](#)

The name of the type which is to be instantiated

### Type Parameters

*TInterfaceType*

The handle to the class is automatically cast to the interfafce TInterfaceType

### Return Value

Type: [TInterfaceType](#)

A handle to the instantiated object. This value should be cast to an interface as only interfaces will work across AppDomains.

Implements  
[IDomainLoaderCreateTInterfaceType\(String\)](#)

## See Also

Reference

[DomainLoader Class](#)

[Create Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderCreate Method (String)

Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public Object Create(  
    string typeName  
)
```

### Parameters

*typeName*

Type: [System.String](#)

The name of the type which is to be instantiated

### Return Value

Type: [Object](#)

A handle to the instantiated object. This value should be cast to an interface as only interfaces will work across AppDomains.

### Implements

[IDomainLoaderCreate\(String\)](#)

## ► See Also

### Reference

DomainLoader Class  
Create Overload  
W.Domains Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderDispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

```
public void Dispose()
```

[Copy](#)

Implements

[IDomainLoaderDispose](#)

[IDisposableDispose](#)

## ► See Also

Reference

[DomainLoader Class](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# DomainLoaderDoCallback Method

Executes an action in the context of the hosted AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public void DoCallback(  
    Action action  
)
```

### Parameters

*action*

Type: [SystemAction](#)

The action to be called in the context of the hosted AppDomain

## ► See Also

Reference

[DomainLoader Class](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# DomainLoaderExecute Method

## ▪ Overload List

	Name	Description
≡	<a href="#">ExecuteTResult(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
≡	<a href="#">Execute(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.

[Top](#)

## ▪ See Also

### Reference

[DomainLoader Class](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# DomainLoaderExecuteTResult Method (String, String, Object)

Instantiates a class and calls a method exposed by it.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public TResult Execute<TResult>(  
    string typeName,  
    string methodName,  
    params Object[] args  
)
```

## Parameters

*typeName*

Type: [System.String](#)

The name of the type which exposes the static method

*methodName*

Type: [System.String](#)

The name of the static method

*args*

Type: [System.Object](#)

Any arguments to be passed to the static method

## Type Parameters

*TResult*

The result of the function call is cast to TResult

## Return Value

Type: *TResult*

The return value from the function, casted to TResult

## Implements

[IDomainLoaderExecuteTResult\(String, String, Object\)](#)

## See Also

### Reference

[DomainLoader Class](#)

[Execute Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderExecute Method (String, String, Object)

Instantiates a class and calls a method exposed by it.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public void Execute(
    string typeName,
    string methodName,
    params Object[] args
)
```

### Parameters

*typeName*

Type: [System.String](#)

The name of the type which exposes the static method

*methodName*

Type: [System.String](#)

The name of the static method

*args*

Type: [System.Object](#)

Any arguments to be passed to the static method

### Implements

[IDomainLoaderExecute\(String, String, Object\)](#)

## ▲ See Also

Reference

[DomainLoader Class](#)

[Execute Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderExecuteStaticMethod Method

## ▪ Overload List

Name	Description
 <a href="#">ExecuteStaticMethodTResult(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain
 <a href="#">ExecuteStaticMethod(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain

[Top](#)

## ▪ See Also

[Reference](#)

[DomainLoader Class](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderExecuteStaticMethod Method (String, String, Object)

Executes a static method on the specified type across the AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public TResult ExecuteStaticMethod<TResult>(
    string typeName,
    string staticMethodName,
    params Object[] args
)
```

### Parameters

*typeName*

Type: [SystemString](#)

The name of the type which exposes the static method

*staticMethodName*

Type: [SystemString](#)

The name of the static method

*args*

Type: [SystemObject](#)

Any parameters to be passed to the static method

### Type Parameters

*TResult*

The result of the function call is cast to TResult

## Return Value

Type: *TResult*

The return value from the function, casted to TResult.

## Implements

[IDomainLoader](#)[ExecuteStaticMethod](#)[TResult](#)([String](#), [String](#), [Object](#))

## See Also

### Reference

[DomainLoader Class](#)

[ExecuteStaticMethod Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderExecuteStaticMethod Method (String, String, Object)

Executes a static method on the specified type across the AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public void ExecuteStaticMethod(
    string typeName,
    string staticMethodName,
    params Object[] args
)
```

### Parameters

*typeName*

Type: [SystemString](#)

The name of the type which exposes the static method

*staticMethodName*

Type: [SystemString](#)

The name of the static method

*args*

Type: [SystemObject](#)

Any arguments to be passed to the static method

### Implements

[IDomainLoaderExecuteStaticMethod\(String, String, Object\)](#)

## ▲ See Also

### Reference

[DomainLoader Class](#)

[ExecuteStaticMethod Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderFinalize Method

Destructs the DomainLoader instance. Calls Dispose.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
protected override void Finalize()
```

Implements

[ObjectFinalize](#)

## ► See Also

[Reference](#)

[DomainLoader Class](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderGetData<TData> Method

Gets the value stored in the current application domain for the specified name

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public TData GetData<TData>(  
    string name  
)
```

## Parameters

*name*

Type: [System.String](#)

The name of a predefined or custom domain property

## Type Parameters

*TData*

The type of data to be returned

## Return Value

Type: [TData](#)

The data stored in the domain property as cast to T

## ► See Also

## Reference

[DomainLoader Class](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderLoad Method

Loads the dlls into the new AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public void Load()
```

Implements

[IDomainLoaderLoad](#)

## ► See Also

[Reference](#)

[DomainLoader Class](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DomainLoaderSetData Method

Sets the value of the specified application domain property

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public void SetData(  
    string name,  
    Object data  
)
```

## Parameters

*name*

Type: [SystemString](#)

The name of a predefined or custom domain property

*data*

Type: [SystemObject](#)

The value to be assigned to the domain property

## ► See Also

Reference

[DomainLoader Class](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# DomainLoaderUnload Method

Unloads the AppDomain and deletes files in the cache folder. The cache folder is where dlls are copied, and run, when using shadow copying.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public void Unload()
```

Implements

[IDomainLoaderUnload](#)

## ► See Also

Reference

[DomainLoader Class](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# IDomainLoader Interface

Defines the interface for a DomainLoader

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public interface IDomainLoader
```

The [IDomainLoader](#) type exposes the following members.

## ► Methods

Name	Description
 <a href="#">Create(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.
 <a href="#">CreateTInterfaceType(String)</a>	Instantiates a class and returns a

handle to it.  
This handle  
must be cast  
to an  
interface in  
order to work  
across  
AppDomains.

 <a href="#">Dispose</a>	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
 <a href="#">Execute(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
 <a href="#">ExecuteTResult(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
 <a href="#">ExecuteStaticMethod(String, String, Object)</a>	Executes a static method on the specified type across

## the AppDomain

---

 <a href="#">ExecuteStaticMethodTResult(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain
 <a href="#">Load</a>	Loads the dlls into the new AppDomain
 <a href="#">Unload</a>	Unloads the AppDomain and deletes files in the cache folder. The cache folder is where dlls are copied, and run, when using shadow copying.

---

[Top](#)

## See Also

[Reference](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoader Methods

The [IDomainLoader](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Create(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.
<a href="#">CreateTInterfaceType(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.
<a href="#">Dispose</a>	Performs application-defined tasks

associated with freeing, releasing, or resetting unmanaged resources.

≡	<a href="#">Execute(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
≡	<a href="#">ExecuteTResult(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
≡	<a href="#">ExecuteStaticMethod(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain
≡	<a href="#">ExecuteStaticMethodTResult(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain
≡	<a href="#">Load</a>	Loads the

dlls into the  
new  
AppDomain

---



## Unload

Unloads the AppDomain and deletes files in the cache folder. The cache folder is where dlls are copied, and run, when using shadow copying.

---

[Top](#)

## See Also

Reference

[IDomainLoader Interface](#)  
[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderCreate Method

## ▪ Overload List

Name	Description
 <a href="#">Create(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.
 <a href="#">CreateTInterfaceType(String)</a>	Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.

[Top](#)

## ▪ See Also

[Reference](#)

[IDomainLoader Interface](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# IDomainLoaderCreate Method (String)

Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
Object Create(  
    string typeName  
)
```

### Parameters

*typeName*

Type: [System.String](#)

The name of the type which is to be instantiated

### Return Value

Type: [Object](#)

A handle to the instantiated object. This value should be cast to an interface as only interfaces will work across AppDomains.

## ► See Also

[Reference](#)

[IDomainLoader Interface](#)

[Create Overload](#)

## W.Domains Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderCreate $T$ InterfaceType Method (String)

Instantiates a class and returns a handle to it. This handle must be cast to an interface in order to work across AppDomains.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
TInterfaceType Create<TInterfaceType>(  
    string typeName  
)
```

### Parameters

*typeName*

Type: [System.String](#)

The name of the type which is to be instantiated

### Type Parameters

*TInterfaceType*

The handle to the class is automatically cast to the interface  
*TInterfaceType*

### Return Value

Type: [TInterfaceType](#)

A handle to the instantiated object. This value should be cast to an interface as only interfaces will work across AppDomains.

## ▲ See Also

Reference

[IDomainLoader Interface](#)

[Create Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderDispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

```
void Dispose()
```

[Copy](#)

## ► See Also

Reference

[IDomainLoader Interface](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderExecute Method

## ▪ Overload List

	Name	Description
≡	<a href="#">ExecuteTResult(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.
≡	<a href="#">Execute(String, String, Object)</a>	Instantiates a class and calls a method exposed by it.

[Top](#)

## ▪ See Also

### Reference

[IDomainLoader Interface](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderExecute $TResult$ Method (String, String, Object)

Instantiates a class and calls a method exposed by it.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
TResult Execute<TResult>(
    string typeName,
    string methodName,
    params Object[] args
)
```

### Parameters

*typeName*

Type: [System.String](#)

The name of the type which exposes the static method

*methodName*

Type: [System.String](#)

The name of the static method

*args*

Type: [System.Object](#)

Any arguments to be passed to the static method

### Type Parameters

*TResult*

The result of the function call is cast to TResult

## Return Value

Type: *TResult*

The return value from the function, casted to TResult

## See Also

[Reference](#)

[IDomainLoader Interface](#)

[Execute Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderExecute Method (String, String, Object)

Instantiates a class and calls a method exposed by it.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

```
C#  
Copy  
void Execute(  
    string typeName,  
    string methodName,  
    params Object[] args  
)
```

### Parameters

*typeName*

Type: [System.String](#)

The name of the type which exposes the static method

*methodName*

Type: [System.String](#)

The name of the static method

*args*

Type: [System.Object](#)

Any arguments to be passed to the static method

## ► See Also

Reference

[IDomainLoader Interface](#)

[Execute Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderExecuteStaticMethod Method

## ► Overload List

Name	Description
 <a href="#">ExecuteStaticMethodTResult(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain
 <a href="#">ExecuteStaticMethod(String, String, Object)</a>	Executes a static method on the specified type across the AppDomain

[Top](#)

## ► See Also

Reference

[IDomainLoader Interface](#)  
[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderExecuteStaticMethod<TResult>.Method (String, String, Object)

Executes a static method on the specified type across the AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
 TResult ExecuteStaticMethod<TResult>(
    string typeName,
    string staticMethodName,
    params Object[] args
)
```

## Parameters

*typeName*

Type: [System.String](#)

The name of the type which exposes the static method

*staticMethodName*

Type: [System.String](#)

The name of the static method

*args*

Type: [System.Object](#)

Any parameters to be passed to the static method

## Type Parameters

*TResult*

The result of the function call is cast to TResult

## Return Value

Type: *TResult*

The return value from the function, casted to TResult.

## See Also

[Reference](#)

[IDomainLoader Interface](#)

[ExecuteStaticMethod Overload](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderExecuteStaticMethod Method (String, String, Object)

Executes a static method on the specified type across the AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
void ExecuteStaticMethod(
    string typeName,
    string staticMethodName,
    params Object[] args
)
```

### Parameters

*typeName*

Type: [System.String](#)

The name of the type which exposes the static method

*staticMethodName*

Type: [System.String](#)

The name of the static method

*args*

Type: [System.Object](#)

Any arguments to be passed to the static method

## ► See Also

[Reference](#)

[IDomainLoader Interface](#)  
[ExecuteStaticMethod Overload](#)  
[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderLoad Method

Loads the dlls into the new AppDomain

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

```
void Load()
```

[Copy](#)

## ► See Also

[Reference](#)

[IDomainLoader Interface](#)

[W.Domains Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IDomainLoaderUnload Method

Unloads the AppDomain and deletes files in the cache folder. The cache folder is where dlls are copied, and run, when using shadow copying.

**Namespace:** [W.Domains](#)

**Assembly:** Tungsten.Domains (in Tungsten.Domains.dll) Version: 2.0.0

## ► Syntax

C#

```
void Unload()
```

[Copy](#)

## ► See Also

[Reference](#)

[IDomainLoader Interface](#)

[W.Domains Namespace](#)

## Tungsten

W<sup>74</sup>

# W.Encryption Namespace

[Missing <summary> documentation for "N:W.Encryption"]

## ► Classes

Class	Description
 <a href="#">AssymmetricEncryption</a>	Facilitates two way (assymmetric) encryption RSA cryptography
 <a href="#">MD5</a>	Used to generate MD5 hashes and strings against them
 <a href="#">PasswordHash</a>	Salted password hashing with PBKDF2 Author: havoc AT defuse.ca <a href="http://www.defuse.ca/hashing-security.htm">www: http://www.defuse.ca/hashing-security.htm</a> Compatibility: .NET 3.0 and later.
 <a href="#">RSA</a>	Provides RSA encryption functionality
 <a href="#">RSAMethods</a>	Replaces RSA. This code was adapted from .NETStandard from an article published on CodeProject by Mathew John Schlauder. It is less complicated but works more closely with the initial RSA implementation. See: <a href="https://www.codeproject.com/Articles/1000000/Key-RSA-Encryption-in-C-NET">https://www.codeproject.com/Articles/1000000/Key-RSA-Encryption-in-C-NET</a>

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AssymmetricEncryption Class

Facilitates two way (assymmetric) encryption via RSA cryptography

## ► Inheritance Hierarchy

[SystemObject](#) [W.EncryptionAssymmetricEncryption](#)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public class AssymmetricEncryption
```

The [AssymmetricEncryption](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">AssymmetricEncryption</a>	Constructs a new TwoWayEncryption instance

[Top](#)

## ► Properties

	Name	Description
!	<a href="#">CurrentKeySize</a>	The KeySize the object was

created with

	<a href="#">LegalKeySizes</a>	The legal RSA key sizes supported by the platform
	<a href="#">PublicKey</a>	The local public key
	<a href="#">RemotePublicKey</a>	The remote's public key

[Top](#)

## Methods

	Name	Description
	<a href="#">Decrypt</a>	Decrypts data with the local private key
	<a href="#">Encrypt</a>	Encrypts data with the remote public key
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">ExchangeKeys</a>	Calls the function which completes the exchange and sets <code>RemotePublicKey</code> to the result. This function must be implemented by the developer and is contextual to his or her scenario. In all cases however, the return value must be the remote public key upon success, or null to specify a failure.
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it

is reclaimed by garbage collection.  
(Inherited from [Object](#).)

 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ► Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a>

members  
(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by

[MonitorExtensions.](#))



### [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets its `IsDirty` flag to `false`  
(Defined by [PropertyHostExtensions.](#))



### [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions.](#))

---

[Top](#)

## ► See Also

Reference

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AssymmetricEncryption Constructor

Constructs a new TwoWayEncryption instance

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C#  
public AssymmetricEncryption(  
    int keySize = 2048  
)
```

[Copy](#)

## Parameters

*keySize* **(Optional)**

Type: [SystemInt32](#)

The encryption key size

## ► See Also

### Reference

[AssymmetricEncryption Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# AssymmetricEncryption Properties

The [AssymmetricEncryption](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">CurrentKeySize</a>	The KeySize the object was created with
	<a href="#">LegalKeySizes</a>	The legal RSA key sizes supported by the platform
	<a href="#">PublicKey</a>	The local public key
	<a href="#">RemotePublicKey</a>	The remote's public key

[Top](#)

## See Also

[Reference](#)

[AssymmetricEncryption Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# AssymmetricEncryptionCurrentKeySize Property

The KeySize the object was created with

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public int CurrentKeySize { get; }
```

Property Value

Type: [Int32](#)

## ► See Also

Reference

[AssymmetricEncryption Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AssymmetricEncryptionLegalKeySizes Property

The legal RSA key sizes supported by the platform

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C#  
public KeySizes[] LegalKeySizes { get; }
```

Property Value

Type: [KeySizes](#)

## ► See Also

Reference

[AssymmetricEncryption Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AssymmetricEncryptionPublicKey Property

The local public key

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public RSAParameters PublicKey { get; }
```

Property Value

Type: [RSAParameters](#)

## ► See Also

Reference

[AssymmetricEncryption Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AssymmetricEncryptionRemotePublicKey Property

The remote's public key

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public Nullable<RSAParameters> RemotePublicKey {
```

Property Value

Type: [NullableRSAParameters](#)

## ► See Also

Reference

[AssymmetricEncryption Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# AssymmetricEncryption Methods

The [AssymmetricEncryption](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Decrypt</a>	Decrypts data with the local private key
≡	<a href="#">Encrypt</a>	Encrypts data with the remote public key
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ExchangeKeys</a>	Calls the function which completes the exchange and sets RemotePublicKey to the result. This function must be implemented by the developer and is contextual to his or her scenario. In all cases however, the return value must be the remote public key upon success, or null to specify a failure.
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)

	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

### Reference

[AssymmetricEncryption Class](#)  
[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AssymmetricEncryptionDecrypt Method

Decrypts data with the local private key

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C#  
public bool Decrypt(  
    ref byte[] bytes  
)
```

[Copy](#)

### Parameters

*bytes*

Type: [SystemByte](#)

The data to decrypt

### Return Value

Type: [Boolean](#)

The decrypted data

## ► See Also

[Reference](#)

[AssymmetricEncryption Class](#)

[W.Encryption Namespace](#)



## Tungsten

W<sup>74</sup>

# AssymmetricEncryptionEncrypt Method

Encrypts data with the remote public key

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C#  
public bool Encrypt(  
    ref byte[] bytes  
)
```

[Copy](#)

### Parameters

*bytes*

Type: [SystemByte](#)

The data to encrypt with the remote public key

### Return Value

Type: [Boolean](#)

[Missing <returns> documentation for  
"M:W.Encryption.AssymmetricEncryption.Encrypt(System.Byte[]@)"]

## ► See Also

Reference

[AssymmetricEncryption Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# AssymmetricEncryptionExchangeKeys Method

Calls the function which completes the exchange and sets RemotePublicKey to the result. This function must be implemented by the developer and is contextual to his or her scenario. In all cases however, the return value must be the remote public key upon success, or null to specify a failure.

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public bool ExchangeKeys(
    Func<RSAParameters, Nullable<RSAParameter>>
)
```

### Parameters

*del*

Type: [SystemFuncRSAParameters, NullableRSAParameters](#)  
The function to call

### Return Value

Type: [Boolean](#)

True if RemotePublicKey was assigned a non-null value, otherwise False

## ► See Also

## Reference

[AssymmetricEncryption Class](#)  
[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MD5 Class

Used to generate MD5 hashes and verify input strings against them

## ► Inheritance Hierarchy

[SystemObject](#) [W.EncryptionMD5](#)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class MD5
```

The [MD5](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">MD5</a>	Initializes a new instance of the <a href="#">MD5</a> class

[Top](#)

## ► Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to

		the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡 <b>S</b>	<a href="#">GetMd5Hash(String)</a>	Generates an MD5 hash of the input string
💡 <b>S</b>	<a href="#">GetMd5Hash(String, MD5)</a>	Generates an MD5 hash of the input string
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
💡 <b>S</b>	<a href="#">VerifyMd5Hash</a>	Verifies a hash against a string

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by

[MonitorExtensions.](#))

	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

[Top](#)

## ▲ See Also

[Reference](#)

[W.Encryption Namespace](#)



## Tungsten

W<sup>74</sup>

# MD5 Constructor

Initializes a new instance of the [MD5](#) class

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public MD5()
```

## ► See Also

[Reference](#)

[MD5 Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MD5 Methods

The [MD5](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡ S	<a href="#">GetMd5Hash(String)</a>	Generates an MD5 hash of the input string
≡ S	<a href="#">GetMd5Hash(String, MD5)</a>	Generates an MD5 hash of the input string
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of

the current [Object](#).  
(Inherited from [Object](#).)

 	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
 	<a href="#">VerifyMd5Hash</a>	Verifies a hash against a string

[Top](#)

## ▪ Extension Methods

Name	Description
  	<a href="#">AsTType</a> Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a> Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a> Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ◀ See Also

[Reference](#)

[MD5 Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MD5GetMd5Hash Method

## ▪ Overload List

	Name	Description
≡ 	<a href="#">GetMd5Hash(String)</a>	Generates an MD5 hash of the input string
≡ 	<a href="#">GetMd5Hash(String, MD5)</a>	Generates an MD5 hash of the input string

[Top](#)

## ▪ See Also

[Reference](#)

[MD5 Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MD5GetMd5Hash Method (String)

Generates an MD5 hash of the input string

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static string GetMd5Hash(  
    string input  
)
```

## Parameters

*input*

Type: [System.String](#)

An MD5 hash of this input will be created

## Return Value

Type: [String](#)

An MD5 hash of the inputted value

## ► See Also

[Reference](#)

[MD5 Class](#)

[GetMd5Hash Overload](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MD5GetMd5Hash Method (String, MD5)

Generates an MD5 hash of the input string

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static string GetMd5Hash(  
    string input,  
    MD5 md5  
)
```

## Parameters

*input*

Type: [System.String](#)

An MD5 hash of this input will be created

*md5*

Type: [System.Security.CryptographyMD5](#)

The previously allocated MD5 object to use

## Return Value

Type: [String](#)

An MD5 hash of the inputted value

## ► See Also

[Reference](#)

MD5 Class  
GetMd5Hash Overload  
W.Encryption Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MD5VerifyMd5Hash Method

Verifies a hash against a string

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C#  
public static bool VerifyMd5Hash(  
    string input,  
    string hash  
)
```

[Copy](#)

## Parameters

*input*

Type: [SystemString](#)

The string to verify

*hash*

Type: [SystemString](#)

The MD5 hash used in the verification

## Return Value

Type: [Boolean](#)

True if the input string is verified, otherwise False

## ► See Also

[Reference](#)

[MD5 Class](#)

[W.Encryption Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PasswordHash Class

Salted password hashing with PBKDF2-SHA1. Author: havoc AT defuse.ca www: <http://crackstation.net/hashing-security.htm>  
Compatibility: .NET 3.0 and later.

## ► Inheritance Hierarchy

[SystemObject](#) [W.EncryptionPasswordHash](#)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class PasswordHash
```

The [PasswordHash](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PasswordHash</a>	Initializes a new instance of the <a href="#">PasswordHash</a> class

[Top](#)

## ► Methods

	Name	Description
≡	<a href="#">S</a>	

	<a href="#">CreateHash</a>	Creates a salted PBKDF2 hash of the password.
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetLicenseInfo</a>	License information related to the author of PasswordHash (Taylor Hornby)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">ValidatePassword</a>	Validates a password given a hash of the correct one.

[Top](#)

## Fields

Name	Description
  <a href="#">HASH_BYTE_SIZE</a>	
  <a href="#">ITERATION_INDEX</a>	
  <a href="#">PBKDF2_INDEX</a>	
  <a href="#">PBKDF2_ITERATIONS</a>	
  <a href="#">SALT_BYTE_SIZE</a>	
  <a href="#">SALT_INDEX</a>	

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by

[PropertyHostExtensions.\)](#)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	Scans each field and

## [MarkAsClean](#)

property of type [IProperty](#)  
and sets it's `IsDirty` flag to  
`false`

(Defined by  
[PropertyHostExtensions](#).)



## [Unlock](#)

Performs a Monitor  
unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

Reference

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PasswordHash Constructor

Initializes a new instance of the [PasswordHash](#) class

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public PasswordHash()
```

## ► See Also

[Reference](#)

[PasswordHash Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PasswordHash Methods

The [PasswordHash](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">CreateHash</a>	Creates a salted PBKDF2 hash of the password.
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 	<a href="#">GetLicenseInfo</a>	License information related to the author of PasswordHash (Taylor Hornby)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)

	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
 	<a href="#">ValidatePassword</a>	Validates a password given a hash of the correct one.

[Top](#)

## Extension Methods

Name	Description
 	<a href="#">AsTType</a> Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a> Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a> Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

•	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
•	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
•	<a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

Reference

[PasswordHash Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PasswordHashCreateHash Method

Creates a salted PBKDF2 hash of the password.

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C#  
public static string CreateHash(  
    string password  
)
```

[Copy](#)

### Parameters

*password*

Type: [SystemString](#)

The password to hash.

### Return Value

Type: [String](#)

The hash of the password.

## ► See Also

[Reference](#)

[PasswordHash Class](#)

[W.Encryption Namespace](#)



## Tungsten

W<sup>74</sup>

# PasswordHashGetLicenseInfo Method

License information related to the author of PasswordHash (Taylor Hornby)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static string GetLicenseInfo()
```

Return Value

Type: [String](#)

License information related to the author of PasswordHash (Taylor Hornby)

## ► See Also

[Reference](#)

[PasswordHash Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# PasswordHashValidatePassword Method

Validates a password given a hash of the correct one.

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C#  
public static bool ValidatePassword(  
    string password,  
    string correctHash  
)
```

[Copy](#)

### Parameters

*password*

Type: [SystemString](#)

The password to check.

*correctHash*

Type: [SystemString](#)

A hash of the correct password.

### Return Value

Type: [Boolean](#)

True if the password is correct. False otherwise.

## ► See Also

[Reference](#)

PasswordHash Class  
W.Encryption Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PasswordHash Fields

The [PasswordHash](#) type exposes the following members.

## ▪ Fields

Name	Description
<a href="#"><b>HASH_BYTE_SIZE</b></a>	
<a href="#"><b>ITERATION_INDEX</b></a>	
<a href="#"><b>PBKDF2_INDEX</b></a>	
<a href="#"><b>PBKDF2_ITERATIONS</b></a>	
<a href="#"><b>SALT_BYTE_SIZE</b></a>	
<a href="#"><b>SALT_INDEX</b></a>	

[Top](#)

## ▪ See Also

Reference

[PasswordHash Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# PasswordHashHASH\_BYTE\_SIZE Field

[Missing <summary> documentation for "F:W.Encryption.PasswordHash.HASH\_BYTE\_SIZE"]

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public const int HASH_BYTE_SIZE = 24
```

Field Value

Type: [Int32](#)

## ► See Also

[Reference](#)

[PasswordHash Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# PasswordHashITERATION\_INDEX Field

[Missing <summary> documentation for "F:W.Encryption.PasswordHash.ITERATION\_INDEX"]

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public const int ITERATION_INDEX = 0
```

Field Value

Type: [Int32](#)

## ► See Also

[Reference](#)

[PasswordHash Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# PasswordHashPBKDF2\_INDEX Field

[Missing <summary> documentation for "F:W.Encryption.PasswordHash.PBKDF2\_INDEX"]

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public const int PBKDF2_INDEX = 2
```

Field Value

Type: [Int32](#)

## ► See Also

[Reference](#)

[PasswordHash Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# PasswordHashPBKDF2\_ITERATION Field

[Missing <summary> documentation for  
"F:W.Encryption.PasswordHash.PBKDF2\_ITERATIONS"]

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public const int PBKDF2_ITERATIONS = 1000
```

Field Value

Type: [Int32](#)

## ► See Also

[Reference](#)

[PasswordHash Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# PasswordHashSALT\_BYTEx\_SIZE Field

[Missing <summary> documentation for "F:W.Encryption.PasswordHash.SALT\_BYTEx\_SIZE"]

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public const int SALT_BYTEx_SIZE = 24
```

Field Value

Type: [Int32](#)

## ► See Also

[Reference](#)

[PasswordHash Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# PasswordHashSALT\_INDEX Field

[Missing <summary> documentation for "F:W.Encryption.PasswordHash.SALT\_INDEX"]

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public const int SALT_INDEX = 1
```

Field Value

Type: [Int32](#)

## ► See Also

Reference

[PasswordHash Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# RSA Class

Provides RSA encryption functionality

## ► Inheritance Hierarchy

[SystemObject](#) [W.EncryptionRSA](#)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class RSA : IDisposable
```

The [RSA](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">RSA</a>	Constructs a new RSA

[Top](#)

## ► Properties

	Name	Description
⊕	<a href="#">KeySize</a>	The encryption key size
⊕	<a href="#">LegalKeySizes</a>	Gets the key sizes that are

supported by the asymmetric algorithm

[Top](#)

## Methods

	Name	Description
≡	<a href="#">Decrypt(Byte)</a>	Decrypts a byte[] (previously encrypted with the Encrypt method)
≡	<a href="#">Decrypt(Byte, RSAParameters)</a>	Decrypts a byte[] (previously encrypted with the Encrypt method)
≡	<a href="#">Dispose</a>	Disposes the instance and releases resources
≡	<a href="#">Encrypt(Byte)</a>	Encrypts a byte[]
≡	<a href="#">Encrypt(Byte, RSAParameters)</a>	Encrypts a byte[]
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash

function.

(Inherited from [Object](#).)

 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Fields

Name	Description
 <a href="#">PrivateKey</a>	The private key used to decrypt data(do not share)
 <a href="#">PublicKey</a>	The public key used to encrypt data (should be shared)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)



## InitializeProperties

Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members  
(Defined by [PropertyHostExtensions](#).)



## InLock(Action)

Overloaded.  
Performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)



## InLockTType(FuncTType)

Overloaded.  
Performs the function in a Monitor lock  
(Defined by [MonitorExtensions](#).)



## InLockAsync(Action)

Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)



## InLockAsyncTType(FuncTType)

Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)



## IsDirty

Scans the IsDirty value of

each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)



### [Lock](#)

Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and property of type IProperty and sets its IsDirty flag to false  
(Defined by [PropertyHostExtensions](#).)



### [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

[Top](#)

## ▲ See Also

Reference

[W.Encryption Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSA Constructor

Constructs a new RSA

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public RSA(  
    int keySize  
)
```

## Parameters

*keySize*

Type: [System.Int32](#)

[Missing <param name="keySize"/> documentation for  
"M:W.Encryption.RSA.#ctor(System.Int32)"]

## ► See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# RSA Properties

The [RSA](#) type exposes the following members.

## ▪ Properties

	Name	Description
	<a href="#">KeySize</a>	The encryption key size
	<a href="#">LegalKeySizes</a>	Gets the key sizes that are supported by the asymmetric algorithm

[Top](#)

## ▪ See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAKeySize Property

The encryption key size

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public int KeySize { get; }
```

Property Value

Type: [Int32](#)

## ► See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSALegalKeySizes Property

Gets the key sizes that are supported by the asymmetric algorithm

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public KeySizes[] LegalKeySizes { get; }
```

Return Value

Type: [KeySizes](#)

An enumeration of the supported key sizes

## ► See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSA Methods

The [RSA](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Decrypt(Byte)</a>	Decrypts a byte[] (previously encrypted with the Encrypt method)
≡	<a href="#">Decrypt(Byte, RSAParameters)</a>	Decrypts a byte[] (previously encrypted with the Encrypt method)
≡	<a href="#">Dispose</a>	Disposes the instance and releases resources
≡	<a href="#">Encrypt(Byte)</a>	Encrypts a byte[]
≡	<a href="#">Encrypt(Byte, RSAParameters)</a>	Encrypts a byte[]
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)

 	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
 	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAEncrypt Method

## ▪ Overload List

	Name	Description
≡	<a href="#">Decrypt(Byte)</a>	Decrypts a byte[] (previously encrypted with the Encrypt method)
≡	<a href="#">Decrypt(Byte, RSAParameters)</a>	Decrypts a byte[] (previously encrypted with the Encrypt method)

[Top](#)

## ▪ See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAEncrypt Method (Byte)

Decrypts a byte[] (previously encrypted with the Encrypt method)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public byte[] Decrypt(  
    byte[] bytes  
)
```

### Parameters

*bytes*

Type: [SystemByte](#)

The encrypted byte[]

### Return Value

Type: [Byte](#)

A byte[] containing the decrypted value

## ► See Also

[Reference](#)

[RSA Class](#)

[Decrypt Overload](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# RSAEncrypt Method (Byte, RSAParameters)

Decrypts a byte[] (previously encrypted with the Encrypt method)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public byte[] Decrypt(
    byte[] bytes,
    RSAParameters privateKey
)
```

### Parameters

*bytes*

Type: [System.Byte](#)

The encrypted byte[]

*privateKey*

Type: [System.Security.Cryptography.RSAParameters](#)

The private key used to decrypt the byte[]

### Return Value

Type: [Byte](#)

A byte[] containing the decrypted value

## ► See Also

Reference

RSA Class  
Decrypt Overload  
W.Encryption Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSADispose Method

Disposes the instance and releases resources

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public virtual void Dispose()
```

Implements

[IDisposableDispose](#)

## ► See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAEncrypt Method

## ► Overload List

Name	Description
 <a href="#">Encrypt(Byte)</a>	Encrypts a byte[]
 <a href="#">Encrypt(Byte, RSAParameters)</a>	Encrypts a byte[]

[Top](#)

## ► See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAEncrypt Method (Byte)

Encrypts a byte[]

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public byte[] Encrypt(  
    byte[] bytes  
)
```

### Parameters

*bytes*

Type: [SystemByte](#)

The byte[] to be encrypted

### Return Value

Type: [Byte](#)

A byte[] containing the encrypted bytes

## ► See Also

[Reference](#)

[RSA Class](#)

[Encrypt Overload](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# RSAEncrypt Method (Byte, RSAParameters)

Encrypts a byte[]

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C#  
public byte[] Encrypt(  
    byte[] bytes,  
    RSAParameters publicKey  
)
```

[Copy](#)

### Parameters

*bytes*

Type: [System.Byte](#)

The bytes to encrypt

*publicKey*

Type: [System.Security.Cryptography.RSAParameters](#)

The public key used to encrypt the bytes

### Return Value

Type: [Byte](#)

A byte[] containing the encrypted bytes

## ► See Also

[Reference](#)

RSA Class  
Encrypt Overload  
W.Encryption Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSA Fields

The [RSA](#) type exposes the following members.

## ▪ Fields

	Name	Description
◆	<a href="#">PrivateKey</a>	The private key used to decrypt data(do not share)
◆	<a href="#">PublicKey</a>	The public key used to encrypt data (should be shared)

[Top](#)

## ▪ See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAPrivateKey Field

The private key used to decrypt data(do not share)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public RSAParameters PrivateKey
```

Field Value

Type: [RSAParameters](#)

## ► See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAPublicKey Field

The public key used to encrypt data (should be shared)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public RSAParameters PublicKey
```

Field Value

Type: [RSAParameters](#)

## ► See Also

[Reference](#)

[RSA Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAMethods Class

Replaces RSA. This code was adapted for NetStandard from an article published on CodeProject by Mathew John Schlabaugh in 2007. It is less complicated but works more often than my initial RSA implementation. See:

<https://www.codeproject.com/Articles/10877/Public-Key-RSA-Encryption-in-C-NET>

## ► Inheritance Hierarchy

[SystemObject](#) [W.EncryptionRSAMethods](#)

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static class RSAMethods
```

The [RSAMethods](#) type exposes the following members.

## ► Methods

	Name	Description
≡ ♀ S	<a href="#">CreateKeyPair</a>	Generates a public/private key pair
≡ ♀ S	<a href="#">Decrypt</a>	Decrypts a byte array which was previously encrypted with the Encrypt method
≡ ♀ S		

	<a href="#">DecryptAsync</a>	Asynchronously decrypts a string which was previously encrypted with the Encrypt method
  	<a href="#">Encrypt</a>	Encrypts a byte array using the specified keysize and public key
  	<a href="#">EncryptAsync</a>	Asynchronously encrypts a string using the specified keysize and public key
  	<a href="#">LegalKeySizes</a>	Returns an array containing the supported key sizes

[Top](#)

## ▲ See Also

[Reference](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAMethods Methods

The [RSAMethods](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡ 	<a href="#">CreateKeyPair</a>	Generates a public/private key pair
≡ 	<a href="#">Decrypt</a>	Decrypts a byte array which was previously encrypted with the Encrypt method
≡ 	<a href="#">DecryptAsync</a>	Asynchronously decrypts a string which was previously encrypted with the Encrypt method
≡ 	<a href="#">Encrypt</a>	Encrypts a byte array using the specified keysize and public key
≡ 	<a href="#">EncryptAsync</a>	Asynchronously encrypts a string using the specified keysize and public key
≡ 	<a href="#">LegalKeySizes</a>	Returns an array containing the supported key sizes

[Top](#)

## ▪ See Also

[Reference](#)

[RSAMethods Class](#)

## [W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAMethodsCreateKeyPair Method

Generates a public/private key pair

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static void CreateKeyPair(
    int keySize,
    out RSAParameters privateKey,
    out RSAParameters publicKey
)
```

### Parameters

*keySize*

Type: [System.Int32](#)

The key size to use when creating the public and private keys

*privateKey*

Type: [System.Security.Cryptography.RSAParameters](#)

The generated private key

*publicKey*

Type: [System.Security.Cryptography.RSAParameters](#)

The generated public key

### Return Value

Type:

A newly created PublicPrivateKeyPair containing the public and

private keys

## ► See Also

[Reference](#)

[RSAMethods Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAMethodsDecrypt Method

Decrypts a byte array which was previously encrypted with the Encrypt method

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static byte[] Decrypt(  
    byte[] bytes,  
    RSAParameters key  
)
```

## Parameters

*bytes*

Type: [SystemByte](#)

The encrypted data

*key*

Type: [System.Security.CryptographyRSAParameters](#)

The key to decrypt the data

## Return Value

Type: [Byte](#)

A byte array containing the decrypted value

## ► See Also

Reference

[RSAMethods Class](#)

## [W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAMethodsDecryptAsync Method

Asynchronously decrypts a string which was previously encrypted with the Encrypt method

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static Task<byte[]> DecryptAsync(  
    byte[] bytes,  
    RSAParameters key  
)
```

## Parameters

*bytes*

Type: [SystemByte](#)

The encrypted byte array

*key*

Type: [System.Security.CryptographyRSAParameters](#)

The key to decrypt the data

## Return Value

Type: [TaskByte](#)

A byte array containing the decrypted value

## ► See Also

Reference

[RSAMethods Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAMethodsEncrypt Method

Encrypts a byte array using the specified keysize and public key

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static byte[] Encrypt(  
    byte[] bytes,  
    RSAParameters key  
)
```

### Parameters

*bytes*

Type: [System.Byte](#)

The data to encrypt

*key*

Type: [System.Security.Cryptography.RSAParameters](#)

The key used to encrypt the data

### Return Value

Type: [Byte](#)

A byte array containing the encrypted data

## ► See Also

[Reference](#)

[RSAMethods Class](#)

[W.Encryption Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAMethodsEncryptAsync Method

Asynchronously encrypts a string using the specified keysize and public key

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static Task<byte[]> EncryptAsync(  
    byte[] bytes,  
    RSAParameters key  
)
```

### Parameters

*bytes*

Type: [SystemByte](#)

The data to encrypt

*key*

Type: [System.Security.CryptographyRSAParameters](#)

The key to encrypt the data

### Return Value

Type: [TaskByte](#)

A byte array containing the encrypted data

## ► See Also

Reference

[RSAMethods Class](#)

[W.Encryption Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RSAMethodsLegalKeySizes Method

Returns an array containing the supported key sizes

**Namespace:** [W.Encryption](#)

**Assembly:** Tungsten.Encryption (in Tungsten.Encryption.dll) Version: 2.0.2

## ► Syntax

```
C#  
public static KeySizes[] LegalKeySizes()
```

**Return Value**

Type: [KeySizes](#)

An array of supported key sizes

## ► See Also

[Reference](#)

[RSAMethods Class](#)

[W.Encryption Namespace](#)

## Tungsten

W<sup>74</sup>

# W.Firewall Namespace

[Missing <summary> documentation for "N:W.Firewall"]

## ► Classes

	Class	Description
	<a href="#">Rules</a>	Provides static methods to add, remove and check the existance of, Windows firewall rules

## ► Enumerations

	Enumeration	Description
	<a href="#">RulesEFirewallProfiles</a>	The firewall profile type
	<a href="#">RulesEFirewallRuleAction</a>	Firewall rule actions

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Rules Class

Provides static methods to add, remove and check the existance of, Windows firewall rules

## ► Inheritance Hierarchy

[SystemObject](#) [W.FirewallRules](#)

**Namespace:** [W.Firewall](#)

**Assembly:** Tungsten.Firewall (in Tungsten.Firewall.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public static class Rules
```

The [Rules](#) type exposes the following members.

## ► Methods

	Name	Description
≡  	<a href="#">Add</a>	Adds a rule to the firewall
≡  	<a href="#">Exists</a>	Checks if a particular rule exists
≡  	<a href="#">Remove</a>	Removes a firewall rule

[Top](#)

## ► See Also

[Reference](#)

## [W.Firewall Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Rules Methods

The [Rules](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡  	<a href="#">Add</a>	Adds a rule to the firewall
≡  	<a href="#">Exists</a>	Checks if a particular rule exists
≡  	<a href="#">Remove</a>	Removes a firewall rule

[Top](#)

## ▪ See Also

[Reference](#)

[Rules Class](#)

[W.Firewall Namespace](#)

## Tungsten

W<sup>74</sup>

# RulesAdd Method

Adds a rule to the firewall

**Namespace:** [W.Firewall](#)

**Assembly:** Tungsten.Firewall (in Tungsten.Firewall.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public static void Add(
    string ruleName,
    string ruleGroup,
    int protocol = 6,
    string localPorts = "80",
    RulesEFirewallRuleAction action = RulesEFirewallRuleAction.Allow,
    RulesEFirewallProfiles profiles = RulesEFirewallProfiles.Default)
)
```

## Parameters

*ruleName*

Type: [SystemString](#)

The name of the rule to add

*ruleGroup*

Type: [SystemString](#)

The group under which the rule is added

*protocol (Optional)*

Type: [SystemInt32](#)

The desired rule protocol

*localPorts (Optional)*

Type: [SystemString](#)

The desired rule port

*action (Optional)*

Type: [W.FirewallRulesEFirewallRuleAction](#)

The desired rule action, to allow or block communications  
*profiles (Optional)*

Type: [W.FirewallRulesEFirewallProfiles](#)

The desired rule profile

## ► See Also

[Reference](#)

[Rules Class](#)

[W.Firewall Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RulesExists Method

Checks if a particular rule exists

**Namespace:** [W.Firewall](#)

**Assembly:** Tungsten.Firewall (in Tungsten.Firewall.dll) Version: 2.0.0

## ► Syntax

```
C#  
public static bool Exists(  
    string ruleName  
)
```

[Copy](#)

### Parameters

*ruleName*

Type: [System.String](#)

The name of the rule to check

### Return Value

Type: [Boolean](#)

True if the rule exists, otherwise false

## ► See Also

[Reference](#)

[Rules Class](#)

[W.Firewall Namespace](#)

## Tungsten

W<sup>74</sup>

# RulesRemove Method

Removes a firewall rule

**Namespace:** [W.Firewall](#)

**Assembly:** Tungsten.Firewall (in Tungsten.Firewall.dll) Version: 2.0.0

## ► Syntax

```
C#  
public static void Remove(  
    string ruleName  
)
```

[Copy](#)

### Parameters

*ruleName*

Type: [System.String](#)

The name of the rule to remove

## ► See Also

[Reference](#)

[Rules Class](#)

[W.Firewall Namespace](#)

## Tungsten

W<sup>74</sup>

# RulesEFirewallProfiles Enumeration

The firewall profile type

**Namespace:** [W.Firewall](#)

**Assembly:** Tungsten.Firewall (in Tungsten.Firewall.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public enum EFirewallProfiles
```

## ► Members

Member name	Value	Description
Public	4	Public
Private	2	Private
Domain	1	Domain
All	2147483647	All

## ► See Also

Reference

[W.Firewall Namespace](#)

## Tungsten

W<sup>74</sup>

# RulesEFirewallRuleAction Enumeration

Firewall rule actions

**Namespace:** [W.Firewall](#)

**Assembly:** Tungsten.Firewall (in Tungsten.Firewall.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public enum EFirewallRuleAction
```

## ► Members

Member name	Value	Description
Allowed	1	Allow communications
Block	2	Block communications

## ► See Also

Reference

[W.Firewall Namespace](#)

## Tungsten

W<sup>74</sup>

# W.InterProcess Namespace

[Missing <summary> documentation for "N:W.InterProcess"]

## ► Classes

	Class	Description
	<a href="#">CopyData</a>	Sends and receives data via WM_COPYDATA
	<a href="#">CopyDataCopyDataStruct</a>	Helper class which converts byte arrays and COPYDATASTRUCTs
	<a href="#">CopyDataTMessage</a>	Sends and receives Generics via WM_COPYDATA
	<a href="#">CopyDataLogger</a>	Logs messages to the specified window via WM_COPYDATA messages

## ► Structures

	Structure	Description
	<a href="#">COPYDATASTRUCT</a>	

## Tungsten

W<sup>74</sup>

# CopyData Class

Sends and receives data via WM\_COPYDATA

## ► Inheritance Hierarchy

```
SystemObject SystemMarshalByRefObject
  System.Windows.FormsNativeWindow
    W.InterProcessCopyData
      W.InterProcessCopyDataTMessage
```

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public class CopyData : NativeWindow
```

The [CopyData](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">CopyData(String)</a>	Constructs a new CopyData instance which can only listen for messages
≡	<a href="#">CopyData(PredicateString, Boolean)</a>	Constructs a new CopyData instance
≡		

	<code>CopyData(IntPtr, PredicateString, Boolean)</code>	Constructs a new CopyData instance
≡	<code>CopyData(Form, PredicateString, Boolean)</code>	Constructs a new CopyData instance

[Top](#)

## Properties

Name	Description
 <a href="#">Handle</a>	Gets the handle for this window. (Inherited from <a href="#">NativeWindow</a> .)
 <a href="#">TargetWindows</a>	The windows which should receive messages

[Top](#)

## Methods

Name	Description
 <a href="#">AssignHandle</a>	Assigns a handle to this winc (Inherited from <a href="#">NativeWindow</a> )
 <a href="#">CreateHandle</a>	Creates a window and its han with the specified creation parameters. (Inherited from <a href="#">NativeWindow</a> )
 <a href="#">CreateObjRef</a>	Creates an object that contai the relevant information requ generate a proxy used to communicate with a remote c (Inherited from <a href="#">MarshalByRefObject</a> .)

---

≡♥	<a href="#">DefWndProc</a>	Invokes the default window procedure associated with the window. (Inherited from <a href="#">NativeWindow</a> )
≡♥	<a href="#">DestroyHandle</a>	Destroys the window and its handle. (Inherited from <a href="#">NativeWindow</a> )
≡♥ S	<a href="#">Enable</a>	Enables a window to receive WM_COPYDATA messages
≡♥	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡♥	<a href="#">Finalize</a>	Releases the resources associated with this window. (Inherited from <a href="#">NativeWindow</a> )
≡♥	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡♥	<a href="#">GetLifetimeService</a>	Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from <a href="#">MarshalByRefObject</a> .)
≡♥	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡♥	<a href="#">InitializeLifetimeService</a>	Obtains a lifetime service object to control the lifetime policy for this instance.

---

		(Inherited from <a href="#">MarshalByRefObject</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone(Boolean)</a>	Creates a shallow copy of the current <a href="#">MarshalByRefObject</a> (Inherited from <a href="#">MarshalByRefObject</a> .)
💡	<a href="#">OnHandleChange</a>	Specifies a notification method is called when the handle for window is changed. (Inherited from <a href="#">NativeWindow</a> )
💡	<a href="#">OnThreadException</a>	When overridden in a derived class, manages an unhandled thread exception. (Inherited from <a href="#">NativeWindow</a> )
💡	<a href="#">OnWM_COPYDATA</a>	Called when the window receives WM_COPYDATA message
💡	<a href="#">RaiseBytesReceived</a>	Raises the BytesReceived event
💡	<a href="#">RaiseError</a>	Raises the Error event
💡	<a href="#">RefreshTargets</a>	Refreshes the list of target windows.
💡	<a href="#">ReleaseHandle</a>	Releases the handle associated with this window. (Inherited from <a href="#">NativeWindow</a> )
💡	<a href="#">Send(Byte)</a>	Sends a message via WM_COPYDATA
💡 S		

	<a href="#">Send(IntPtr, Byte, IntPtr)</a>	Sends a byte array from the current window to windows matching filter predicate
≡ 	<a href="#">Send(IntPtr, Byte, IntPtr, ActionWin32Exception)</a>	Sends a byte array from the current window to windows matching filter predicate
≡ 	<a href="#">Send(IntPtr, Byte, PredicateString, ActionWin32Exception)</a>	Sends a byte array from the current window to windows matching filter predicate
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">WndProc</a>	The window procedure (Overrides <a href="#">NativeWindow.WndProc(Message)</a> )

[Top](#)

## Events

	Name	Description
⚡	<a href="#">BytesReceived</a>	Raised when a message has been received
⚡	<a href="#">Error</a>	Raised when an error occurs while sending or receiving messages

[Top](#)

## Extension Methods

Name	Description

---

 	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs

the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
---	-------------------------	---

---

	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
---	----------------------	---

---

	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
---	-----------------------------	---

---

	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
---	------------------------	---

[Top](#)

## ▲ See Also

[Reference](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyData Constructor

## ▪ Overload List

Name	Description
 <a href="#">CopyData(String)</a>	Constructs a new CopyData instance which can only listen for messages
 <a href="#">CopyData(PredicateString, Boolean)</a>	Constructs a new CopyData instance
 <a href="#">CopyData(IntPtr, PredicateString, Boolean)</a>	Constructs a new CopyData instance
 <a href="#">CopyData(Form, PredicateString, Boolean)</a>	Constructs a new CopyData instance

[Top](#)

## ▪ See Also

[Reference](#)

[CopyData Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyData Constructor (String)

Constructs a new CopyData instance which can only listen for messages

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public CopyData(  
    string windowText  
)
```

### Parameters

*windowText*

Type: [System.String](#)

The Window Text for the underlying NativeWindow

## ► See Also

[Reference](#)

[CopyData Class](#)

[CopyData Overload](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyData Constructor (PredicateString, Boolean)

Constructs a new CopyData instance

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public CopyData(  
    Predicate<string> filter,  
    bool findAll  
)
```

[Copy](#)

## Parameters

*filter*

Type: [System.PredicateString](#)

The predicate used to filter target windows by Window Text

*findAll*

Type: [System.Boolean](#)

If True, multiple windows can be targeted, otherwise only the first window found will be targeted

## ► See Also

[Reference](#)

[CopyData Class](#)

[CopyData Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyData Constructor (IntPtr, PredicateString, Boolean)

Constructs a new CopyData instance

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public CopyData(  
    IntPtr hSourceWnd,  
    Predicate<string> filter,  
    bool findAll  
)
```

[Copy](#)

## Parameters

*hSourceWnd*

Type: [System.IntPtr](#)

The window handle which will send and receive messages

*filter*

Type: [System.PredicateString](#)

The predicate used to filter target windows by Window Text

*findAll*

Type: [System.Boolean](#)

If True, multiple windows can be targeted, otherwise only the first window found will be targeted

## ► See Also

**Reference**

[CopyData Class](#)

[CopyData Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyData Constructor (Form, PredicateString, Boolean)

Constructs a new CopyData instance

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public CopyData(
    Form parent,
    Predicate<string> filter,
    bool findAll
)
```

## Parameters

*parent*

Type: [System.Windows.FormsForm](#)

The Form which will send and receive messages

*filter*

Type: [System.PredicateString](#)

The predicate used to filter target windows by Window Text

*findAll*

Type: [SystemBoolean](#)

If True, multiple windows can be targeted, otherwise only the first window found will be targeted

## ► See Also

**Reference**

[CopyData Class](#)

[CopyData Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyData Properties

The [CopyData](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">Handle</a>	Gets the handle for this window. (Inherited from <a href="#">NativeWindow</a> .)
	<a href="#">TargetWindows</a>	The windows which should receive messages

[Top](#)

## See Also

Reference

[CopyData Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTargetWindows Property

The windows which should receive messages

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
protected IntPtr[] TargetWindows { get; }
```

Property Value

Type: [IntPtr](#)

## ► See Also

Reference

[CopyData Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyData Methods

The [CopyData](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">AssignHandle</a>	Assigns a handle to this window. (Inherited from <a href="#">NativeWindow</a> .)
<a href="#">CreateHandle</a>	Creates a window and its handle with the specified creation parameters. (Inherited from <a href="#">NativeWindow</a> .)
<a href="#">CreateObjRef</a>	Creates an object that contains the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <a href="#">MarshalByRefObject</a> .)
<a href="#">DefWndProc</a>	Invokes the default window procedure associated with this window. (Inherited from <a href="#">NativeWindow</a> .)
<a href="#">DestroyHandle</a>	Destroys the window and its handle. (Inherited from <a href="#">NativeWindow</a> .)
<a href="#">Enable</a>	Enables a window to receive WM_COPYDATA messages.

	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Releases the resources associated with this window. (Inherited from <a href="#">NativeWindow</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetLifetimeService</a>	Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from <a href="#">MarshalByRefObject</a> .)
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">InitializeLifetimeService</a>	Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from <a href="#">MarshalByRefObject</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone(Boolean)</a>	Creates a shallow copy of the current <a href="#">MarshalByRefObject</a> (Inherited from <a href="#">MarshalByRefObject</a> .)
💡	<a href="#">OnHandleChange</a>	Specifies a notification method.

is called when the handle for window is changed.  
(Inherited from [NativeWindow](#))

💡	<a href="#">OnThreadException</a>	When overridden in a derived class, manages an unhandled thread exception. (Inherited from <a href="#">NativeWindow</a> )
💡	<a href="#">OnWM_COPYDATA</a>	Called when the window receives WM_COPYDATA message
💡	<a href="#">RaiseBytesReceived</a>	Raises the BytesReceived event
💡	<a href="#">RaiseError</a>	Raises the Error event
💡	<a href="#">RefreshTargets</a>	Refreshes the list of target windows.
💡	<a href="#">ReleaseHandle</a>	Releases the handle associated with this window. (Inherited from <a href="#">NativeWindow</a> )
💡	<a href="#">Send(Byte)</a>	Sends a message via WM_COPYDATA
💡 <b>S</b>	<a href="#">Send(IntPtr, Byte, IntPtr)</a>	Sends a byte array from the local window to windows matching filter predicate
💡 <b>S</b>	<a href="#">Send(IntPtr, Byte, IntPtr, ActionWin32Exception)</a>	Sends a byte array from the local window to windows matching filter predicate
💡 <b>S</b>	<a href="#">Send(IntPtr, Byte, PredicateString, ActionWin32Exception)</a>	Sends a byte array from the local window to windows matching filter predicate
💡	<a href="#">ToString</a>	Returns a string that represents

current object.  
(Inherited from [Object](#).)



### [WndProc](#)

The window procedure  
(Overrides  
[NativeWindow.WndProc](#)(Mes

[Top](#)

## ► Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ◀ See Also

[Reference](#)

[CopyData Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataEnable Method

Enables a window to receive WM\_COPYDATA messages

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static Win32Exception Enable(  
    IntPtr hWnd  
)
```

### Parameters

*hWnd*

Type: [SystemIntPtr](#)

The handle of the window to enable

### Return Value

Type: [Win32Exception](#)

Null upon success, otherwise a Win32Exception containing exception information

## ► See Also

Reference

[CopyData Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataOnWM\_COPYDATA Method

Called when the window receives a WM\_COPYDATA message

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
protected virtual void OnWM_COPYDATA(  
    COPYDATASTRUCT cds  
)
```

## Parameters

*cds*

Type: [W.InterProcessCOPYDATASTRUCT](#)

The COPYDATASTRUCT associated with the WM\_COPYDATA windows message

## ► See Also

[Reference](#)

[CopyData Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataRaiseBytesReceived Method

Raises the BytesReceived event

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
protected void RaiseBytesReceived(  
    byte[] bytes  
)
```

[Copy](#)

## Parameters

*bytes*

Type: [SystemByte](#)

The bytes received

## ► See Also

Reference

[CopyData Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataRaiseError Method

Raises the Error event

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
protected void RaiseError(  
    Win32Exception e  
)
```

### Parameters

e

Type: [System.ComponentModel.Win32Exception](#)  
The exception

## ► See Also

[Reference](#)

[CopyData Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataRefreshTargets Method

Refreshes the list of target windows

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public void RefreshTargets()
```

## ► See Also

[Reference](#)

[CopyData Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataSend Method

## ▪ Overload List

	Name	Description
≡	<a href="#">Send(Byte)</a>	Sends a message via WM_COPYDATA
≡ S	<a href="#">Send(IntPtr, Byte, IntPtr)</a>	Sends a byte array from the source window to windows matching the filter predicate
≡ S	<a href="#">Send(IntPtr, Byte, IntPtr, ActionWin32Exception)</a>	Sends a byte array from the source window to windows matching the filter predicate
≡ S	<a href="#">Send(IntPtr, Byte, PredicateString, ActionWin32Exception)</a>	Sends a byte array from the source window to windows matching the filter predicate

[Top](#)

## ▪ See Also

Reference

[CopyData Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataSend Method (Byte)

Sends a message via WM\_COPYDATA

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public void Send(  
    byte[] message  
)
```

### Parameters

*message*

Type: [SystemByte](#)

The message to send

## ► See Also

[Reference](#)

[CopyData Class](#)

[Send Overload](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataSend Method (IntPtr, Byte, IntPtr)

Sends a byte array from the source window to windows matching the filter predicate

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public static Win32Exception Send(  
    IntPtr hSourceWnd,  
    byte[] message,  
    IntPtr hTargetWnd  
)
```

## Parameters

*hSourceWnd*

Type: [SystemIntPtr](#)

The window which is sending the message

*message*

Type: [SystemByte](#)

The array of bytes to send

*hTargetWnd*

Type: [SystemIntPtr](#)

The receiving window

## Return Value

Type: [Win32Exception](#)

An exception if one occurs while sending the message

## See Also

Reference

[CopyData Class](#)

[Send Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataSend Method (IntPtr, Byte, IntPtr, Action<Win32Exception>)

Sends a byte array from the source window to windows matching the filter predicate

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public static void Send(
    IntPtr hSourceWnd,
    byte[] message,
    IntPtr hTargetWnd,
    Action<Win32Exception> onError = null
)
```

## Parameters

*hSourceWnd*

Type: [System.IntPtr](#)

The window which is sending the message

*message*

Type: [System.Byte](#)

The array of bytes to send

*hTargetWnd*

Type: [System.IntPtr](#)

The receiving window

*onError* (**Optional**)

Type: [SystemActionWin32Exception](#)

Called if an error occurs

## ▲ See Also

[Reference](#)

[CopyData Class](#)

[Send Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataSend Method (IntPtr, Byte, PredicateString, Action<Win32Exception>)

Sends a byte array from the source window to windows matching the filter predicate

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public static void Send(
    IntPtr hSourceWnd,
    byte[] message,
    Predicate<string> filter,
    Action<Win32Exception> onError = null
)
```

## Parameters

*hSourceWnd*

Type: [System.IntPtr](#)

The window which is sending the message

*message*

Type: [System.Byte](#)

The array of bytes to send

*filter*

Type: [System.PredicateString](#)

Used to target one or more windows based on Window Text

***onError (Optional)***

Type: [SystemActionWin32Exception](#)

Called if an error occurs

## ▲ See Also

Reference

[CopyData Class](#)

[Send Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataWndProc Method

The window procedure

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
protected override void WndProc(  
    ref Message m  
)
```

### Parameters

*m*

Type: [System.Windows.FormsMessage](#)

The message received

## ► See Also

[Reference](#)

[CopyData Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyData Events

The [CopyData](#) type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">BytesReceived</a>	Raised when a message has been received
 <a href="#">Error</a>	Raised when an error occurs while sending or receiving messages

[Top](#)

## ▪ See Also

[Reference](#)

[CopyData Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataBytesReceived Event

Raised when a message has been received

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public event Action<byte[]> BytesReceived
```

Value

Type: [SystemActionByte](#)

## ► See Also

[Reference](#)

[CopyData Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataError Event

Raised when an error occurs while sending or receiving messages

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public event Action<Win32Exception> Error
```

Value

Type: [SystemActionWin32Exception](#)

## ► See Also

[Reference](#)

[CopyData Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataCopyDataStruct Class

Helper class which converts byte arrays and COPYDATASTRUCTs

## ► Inheritance Hierarchy

[SystemObject](#) [W.InterProcessCopyDataCopyDataStruct](#)

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
protected class CopyDataStruct : IDisposable
```

The [CopyDataCopyDataStruct](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">CopyDataCopyDataStruct</a>	Constructs a new CopyDataStruct from the specified byte array

[Top](#)

## ► Methods

	Name	Description
≡ S		

	<a href="#">AsBytes(Message)</a>	Converts the contents of the COPYDATASTRUCT in the IParam of the specified message m to a byte array
≡  S	<a href="#">AsBytes(COPYDATASTRUCT)</a>	Converts the contents of the COPYDATASTRUCT to a byte array
≡ 	<a href="#">Dispose</a>	Disposes the CopyDataStruct and releases resource
≡ 	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡ 	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡ 	<a href="#">GetCopyDataStruct</a>	Get the COPYDATASTRUCT representing the byte array

 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the

constructor of any class  
which has  
`IOwnedProperty`  
members  
(Defined by  
[PropertyHostExtensions](#).)

---

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the <code>IsDirty</code> value of each field and property of type <code>IProperty</code> (Defined by <a href="#">PropertyHostExtensions</a> .)



## Lock

Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)



## MarkAsClean

Scans each field and  
property of type IProperty  
and sets it's IsDirty flag to  
false  
(Defined by  
[PropertyHostExtensions](#).)



## Unlock

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

Reference

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataCopyDataStruct Constructor

Constructs a new CopyDataStruct from the specified byte array

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public CopyDataStruct(  
    byte[] itemBytes  
)
```

[Copy](#)

### Parameters

*itemBytes*

Type: [SystemByte](#)

The byte array used to create the COPYDATASTRUCT

## ► See Also

Reference

[CopyDataCopyDataStruct Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataStruct Methods

The [CopyDataCopyDataStruct](#) type exposes the following members.

## ▪ Methods

Name	Description
   <a href="#">AsBytes(Message)</a>	Converts the contents of the COPYDATASTRUCT in the IParam of the specified message m to a byte array
   <a href="#">AsBytes(COPYDATASTRUCT)</a>	Converts the contents of the COPYDATASTRUCT to a byte array
  <a href="#">Dispose</a>	Disposes the CopyDataStruct and releases resource
  <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations

before it is reclaimed by garbage collection.  
(Inherited from [Object](#).)

≡	<a href="#">GetCopyDataStruct</a>	Get the COPYDATASTRUCT representing the byte array
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ► Extension Methods

	Name	Description
 	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded.

Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)



### [IsDirty](#)

Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)



### [Lock](#)

Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and property of type IProperty and sets it's IsDirty flag to false  
(Defined by [PropertyHostExtensions](#).)



### [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

[Top](#)

## See Also

### Reference

[CopyData](#)[CopyDataStruct Class](#)  
[W.Interop Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataStructAsBytes Method

## ▪ Overload List

Name	Description
  <a href="#">AsBytes(Message)</a>	Converts the contents of the COPYDATASTRUCT in the IParam of the specified message m to a byte array
  <a href="#">AsBytes(COPYDATASTRUCT)</a>	Converts the contents of the COPYDATASTRUCT to a byte array

[Top](#)

## ▪ See Also

### Reference

[CopyDataCopyDataStruct Class](#)  
[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataCopyDataStructAsBytes Method (Message)

Converts the contents of the COPYDATASTRUCT in the IParam of the specified message m to a byte array

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static byte[] AsBytes(  
    Message m  
)
```

### Parameters

*m*

Type: [System.Windows.FormsMessage](#)  
The windows message

### Return Value

Type: [Byte](#)

A contents of the COPYDATASTRUCT

## ► See Also

### Reference

[CopyDataCopyDataStruct Class](#)

[AsBytes Overload](#)

[W.InterProcess Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataCopyDataStructAsBytes Method (COPYDATASTRUCT)

Converts the contents of the COPYDATASTRUCT to a byte array

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public static byte[] AsBytes(  
    COPYDATASTRUCT cds  
)
```

[Copy](#)

### Parameters

*cds*

Type: [W.InterProcessCOPYDATASTRUCT](#)

The COPYDATASTRUCT containing a byte array

### Return Value

Type: [Byte](#)

A contents of the COPYDATASTRUCT

## ► See Also

[Reference](#)

[CopyDataCopyDataStruct Class](#)

[AsBytes Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataCopyDataStructDispose Method

Disposes the CopyDataStruct and releases resource

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

```
public void Dispose()
```

[Copy](#)

Implements  
[IDisposable.Dispose](#)

## ► See Also

Reference

[CopyDataCopyDataStruct Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataCopyDataStructGetCopyD Method

Get the COPYDATASTRUCT representing the byte array

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public COPYDATASTRUCT GetCopyDataStruct()
```

Return Value

Type: [COPYDATASTRUCT](#)

The COPYDATASTRUCT representing the byte array

## ► See Also

Reference

[CopyDataCopyDataStruct Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTMessage Class

Sends and receives Generics via WM\_COPYDATA

## ► Inheritance Hierarchy

```
SystemObject SystemMarshalByRefObject
  System.Windows.FormsNativeWindow
    W.InterProcessCopyData
      W.InterProcessCopyDataTMessage
```

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public sealed class CopyData<TMessage> : CopyData
```

### Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for  
"T:W.InterProcess.CopyData`1"]

The [CopyDataTMessage](#) type exposes the following members.

## ► Constructors

Name	Description
<a href="#">CopyDataTMessage(String)</a>	Constructs a new

CopyData instance which can only listen for messages

	<a href="#">CopyDataTMessage(PredicateString, Boolean)</a>	Constructs a new CopyData instance
	<a href="#">CopyDataTMessage(IntPtr, PredicateString, Boolean)</a>	Constructs a new CopyData instance
	<a href="#">CopyDataTMessage(Form, PredicateString, Boolean)</a>	Constructs a new CopyData instance

[Top](#)

## Properties

	Name	Description
	<a href="#">Handle</a>	Gets the handle for this window. (Inherited from <a href="#">NativeWindow</a> .)

[Top](#)

## Methods

	Name	Description
	<a href="#">AssignHandle</a>	Assigns a handle to this

		window. (Inherited from <a href="#">NativeWindow</a> .)
≡	<a href="#">CreateHandle</a>	Creates a window and its handle with the specified creation parameters. (Inherited from <a href="#">NativeWindow</a> .)
≡	<a href="#">CreateObjRef</a>	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <a href="#">MarshalByRefObject</a> .)
≡	<a href="#">DefWndProc</a>	Invokes the default window procedure associated with this window. (Inherited from <a href="#">NativeWindow</a> .)
≡	<a href="#">DestroyHandle</a>	Destroys the window and its handle. (Inherited from <a href="#">NativeWindow</a> .)
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default

		hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetLifetimeService</a>	Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from <a href="#">MarshalByRefObject</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">InitializeLifetimeService</a>	Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from <a href="#">MarshalByRefObject</a> .)
≡	<a href="#">RefreshTargets</a>	Refreshes the list of target windows (Inherited from <a href="#">CopyData</a> .)
≡	<a href="#">ReleaseHandle</a>	Releases the handle associated with this window. (Inherited from <a href="#">NativeWindow</a> .)
≡	<a href="#">Send(Byte)</a>	Sends a message via WM_COPYDATA (Inherited from <a href="#">CopyData</a> .)

	<a href="#">Send(TMessage)</a>	Sends a message via WM_COPYDATA
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Events

	Name	Description
⚡	<a href="#">BytesReceived</a>	Raised when a message has been received (Inherited from <a href="#">CopyData</a> .)
⚡	<a href="#">Error</a>	Raised when an error occurs while sending or receiving messages (Inherited from <a href="#">CopyData</a> .)
⚡	<a href="#">MessageReceived</a>	Raised when a message has been received

[Top](#)

## Extension Methods

	Name	Description
💡	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡	<a href="#">InitializeProperties</a>	Scans the fields and

properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members  
(Defined by [PropertyHostExtensions](#).)

---

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of

type IProperty  
(Defined by  
[PropertyHostExtensions](#).)

---

 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets it's IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## ► See Also

Reference

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTMessage Constructor

## ▪ Overload List

Name	Description
<code>CopyDataTMessage(String)</code>	Constructs a new CopyData instance which can only listen for messages
<code>CopyDataTMessage(PredicateString, Boolean)</code>	Constructs a new CopyData instance
<code>CopyDataTMessage(IntPtr, PredicateString, Boolean)</code>	Constructs a new CopyData instance
<code>CopyDataTMessage(Form, PredicateString, Boolean)</code>	Constructs a new CopyData instance

[Top](#)

## ▪ See Also

[Reference](#)

## CopyDataTMessage Class W.InterProcess Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTMessage Constructor (String)

Constructs a new CopyData instance which can only listen for messages

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public CopyData(  
    string windowText  
)
```

[Copy](#)

### Parameters

*windowText*

Type: [System.String](#)

The Window Text for the underlying NativeWindow

## ► See Also

[Reference](#)

[CopyDataTMessage Class](#)

[CopyDataTMessage Overload](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataTMessage Constructor (PredicateString, Boolean)

Constructs a new CopyData instance

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public CopyData(  
    Predicate<string> filter,  
    bool findAll  
)
```

[Copy](#)

## Parameters

*filter*

Type: [System.PredicateString](#)

The predicate used to filter target windows by Window Text

*findAll*

Type: [System.Boolean](#)

If True, multiple windows can be targeted, otherwise only the first window found will be targeted

## ► See Also

[Reference](#)

[CopyDataTMessage Class](#)

[CopyDataTMessage Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTMessage Constructor (IntPtr, PredicateString, Boolean)

Constructs a new CopyData instance

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public CopyData(  
    IntPtr hSourceWnd,  
    Predicate<string> filter,  
    bool findAll  
)
```

## Parameters

*hSourceWnd*

Type: [System.IntPtr](#)

The window handle which will send and receive messages

*filter*

Type: [System.PredicateString](#)

The predicate used to filter target windows by Window Text

*findAll*

Type: [System.Boolean](#)

If True, multiple windows can be targeted, otherwise only the first window found will be targeted

## ► See Also

## Reference

[CopyDataTMessage Class](#)

[CopyDataTMessage Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTMessage Constructor (Form, PredicateString, Boolean)

Constructs a new CopyData instance

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C# Copy
public CopyData(
    Form parent,
    Predicate<string> filter,
    bool findAll
)
```

## Parameters

*parent*

Type: [System.Windows.FormsForm](#)

The form which will send and receive messages

*filter*

Type: [System.PredicateString](#)

The predicate used to filter target windows by Window Text

*findAll*

Type: [SystemBoolean](#)

If True, multiple windows can be targeted, otherwise only the first window found will be targeted

## ► See Also

## Reference

[CopyDataTMessage Class](#)

[CopyDataTMessage Overload](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTMessage Properties

The [CopyDataTMessage](#) generic type exposes the following members.

## Properties

Name	Description
 <a href="#">Handle</a>	Gets the handle for this window. (Inherited from <a href="#">NativeWindow</a> .)

[Top](#)

## See Also

Reference

[CopyDataTMessage Class](#)  
[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTMessage Methods

The [CopyDataTMessage](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">AssignHandle</a>	Assigns a handle to this window. (Inherited from <a href="#">NativeWindow</a> .)
<a href="#">CreateHandle</a>	Creates a window and its handle with the specified creation parameters. (Inherited from <a href="#">NativeWindow</a> .)
<a href="#">CreateObjRef</a>	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from <a href="#">MarshalByRefObject</a> .)
<a href="#">DefWndProc</a>	Invokes the default window procedure associated with this window. (Inherited from <a href="#">NativeWindow</a> .)

---

≡	<a href="#">DestroyHandle</a>	Destroys the window and its handle. (Inherited from <a href="#">NativeWindow</a> .)
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetLifetimeService</a>	Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from <a href="#">MarshalByRefObject</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">InitializeLifetimeService</a>	Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from <a href="#">MarshalByRefObject</a> .)
≡	<a href="#">RefreshTargets</a>	Refreshes the list of target windows (Inherited from <a href="#">CopyData</a> .)

---

 <a href="#">ReleaseHandle</a>	Releases the handle associated with this window. (Inherited from <a href="#">NativeWindow</a> .)
 <a href="#">Send(Byte)</a>	Sends a message via WM_COPYDATA (Inherited from <a href="#">CopyData</a> .)
 <a href="#">Send(TMessage)</a>	Sends a message via WM_COPYDATA
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members

(Defined by  
[PropertyHostExtensions](#).)

---

✿ [InLock\(Action\)](#) Overloaded.  
Performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

✿ [InLockTType\(FuncTType\)](#) Overloaded.  
Performs the function in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

✿ [InLockAsync\(Action\)](#) Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

✿ [InLockAsyncTType\(FuncTType\)](#) Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

✿ [IsDirty](#) Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---

✿ [Lock](#) Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)



## [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets it's `IsDirty` flag to false  
(Defined by [PropertyHostExtensions.](#))



## [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions.](#))

---

[Top](#)

## ▲ See Also

### Reference

[CopyDataTMessage Class](#)  
[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTMessageSend Method

## ▪ Overload List

Name	Description
 <a href="#">Send(Byte)</a>	Sends a message via WM_COPYDATA (Inherited from <a href="#">CopyData</a> .)
 <a href="#">Send(TMessage)</a>	Sends a message via WM_COPYDATA

[Top](#)

## ▪ See Also

[Reference](#)

[CopyDataTMessage Class](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataTMessageSend Method (*TMessage*)

Sends a message via WM\_COPYDATA

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public void Send(  
    TMessage message  
)
```

[Copy](#)

## Parameters

*message*

Type: [TMessage](#)

The message to send

## ► See Also

Reference

[CopyDataTMessage Class](#)

[Send Overload](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataTMessage Events

The [CopyDataTMessage](#) generic type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">BytesReceived</a>	Raised when a message has been received (Inherited from <a href="#">CopyData</a> .)
 <a href="#">Error</a>	Raised when an error occurs while sending or receiving messages (Inherited from <a href="#">CopyData</a> .)
 <a href="#">MessageReceived</a>	Raised when a message has been received

[Top](#)

## ▪ See Also

[Reference](#)

[CopyDataTMessage Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataTMessageMessageReceived Event

Raised when a message has been received

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

```
C#  
public event Action<TMessage> MessageReceived
```

[Copy](#)

Value

Type: [SystemAction<TMessage>](#)

## ► See Also

Reference

[CopyDataTMessage Class](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# CopyDataLogger Class

Logs messages to the specified window via WM\_COPYDATA messages

## ► Inheritance Hierarchy

[SystemObject](#) [W.InterProcessCopyDataLogger](#)

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

Copy

```
public static class CopyDataLogger
```

The [CopyDataLogger](#) type exposes the following members.

## ► Methods

	Name	Description
 	<a href="#">LogTheMessage</a>	Log a message to the specified window via WM_COPYDATA messaging

[Top](#)

## ► Examples

```
Log.LogTheMessage += (category, message) =>  
    W.InterProcess.CopyDataLogger.LogTheMessage("ConsoleLogger",
```

true, category, message);

## ► See Also

    Reference

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataLogger Methods

The [CopyDataLogger](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">LogTheMessage</a>	Log a message to the specified window via WM_COPYDATA messaging

[Top](#)

## ▪ See Also

Reference

[CopyDataLogger Class](#)  
[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# CopyDataLoggerLogTheMessage Method

Log a message to the specified window via WM\_COPYDATA messaging

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public static void LogTheMessage(  
    string windowCaption,  
    string message  
)
```

## Parameters

*windowCaption*

Type: [SystemString](#)

[Missing <param name="windowCaption"/> documentation for  
"M:W.InterProcess.CopyDataLogger.LogTheMessage(System.String,System.String)"]

*message*

Type: [SystemString](#)

[Missing <param name="message"/> documentation for  
"M:W.InterProcess.CopyDataLogger.LogTheMessage(System.String,System.String)"]

## ► See Also

[Reference](#)

[CopyDataLogger Class](#)

## [W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# COPYDATASTRUCT Structure

[Missing <summary> documentation for "T:W.InterProcess.COPYDATASTRUCT"]

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

```
public struct COPYDATASTRUCT
```

[Copy](#)

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

## ► Methods

	Name	Description
≡	<a href="#">Equals</a>	Indicates whether this instance and a specified object are equal. (Inherited from <a href="#">ValueType</a> .)
≡	<a href="#">GetHashCode</a>	Returns the hash code for this instance. (Inherited from <a href="#">ValueType</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns the fully qualified type name of this instance. (Inherited from <a href="#">ValueType</a> .)

[Top](#)

## Fields

Name	Description
cbData	
dwData	
lpData	

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by

[MonitorExtensions.](#))

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )

## [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# COPYDATASTRUCT Methods

The [COPYDATASTRUCT](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Equals</a>	Indicates whether this instance and a specified object are equal. (Inherited from <a href="#">ValueType</a> .)
≡	<a href="#">GetHashCode</a>	Returns the hash code for this instance. (Inherited from <a href="#">ValueType</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns the fully qualified type name of this instance. (Inherited from <a href="#">ValueType</a> .)

[Top](#)

## ▪ Extension Methods

	Name	Description
≡ ≡	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)



## InitializeProperties

Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members  
(Defined by [PropertyHostExtensions](#).)



## InLock(Action)

Overloaded.  
Performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)



## InLockTType(FuncTType)

Overloaded.  
Performs the function in a Monitor lock  
(Defined by [MonitorExtensions](#).)



## InLockAsync(Action)

Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)



## InLockAsyncTType(FuncTType)

Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)



## IsDirty

Scans the IsDirty value of

each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)



### [Lock](#)

Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and property of type IProperty and sets its IsDirty flag to false  
(Defined by [PropertyHostExtensions](#).)



### [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

[Top](#)

## ▲ See Also

[Reference](#)

[COPYDATASTRUCT Structure](#)

[W.InterProcess Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# COPYDATASTRUCT Fields

The [COPYDATASTRUCT](#) type exposes the following members.

## ▪ Fields

Name	Description
◆ cbData	
◆ dwData	
◆ lpData	

[Top](#)

## ▪ See Also

Reference

[COPYDATASTRUCT Structure](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# COPYDATASTRUCTcbData

## Field

[Missing <summary> documentation for  
"F:W.InterProcess.COPYDATASTRUCT.cbData"]

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

### ► Syntax

C#

[Copy](#)

```
public int cbData
```

Field Value

Type: [Int32](#)

### ► See Also

[Reference](#)

[COPYDATASTRUCT Structure](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# COPYDATASTRUCTdwData Field

[Missing <summary> documentation for "F:W.InterProcess.COPYDATASTRUCT.dwData"]

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public IntPtr dwData
```

Field Value

Type: [IntPtr](#)

## ► See Also

[Reference](#)

[COPYDATASTRUCT Structure](#)

[W.InterProcess Namespace](#)

## Tungsten

W<sup>74</sup>

# COPYDATASTRUCTIpData

## Field

[Missing <summary> documentation for "F:W.InterProcess.COPYDATASTRUCT.IpData"]

**Namespace:** [W.InterProcess](#)

**Assembly:** Tungsten.InterProcess (in Tungsten.InterProcess.dll)

Version: 2.0.0

### ► Syntax

C#

Copy

```
public IntPtr lpData
```

Field Value

Type: [IntPtr](#)

### ► See Also

[Reference](#)

[COPYDATASTRUCT Structure](#)

[W.InterProcess Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# W.IO.Pipes Namespace

[Missing <summary> documentation for "N:W.IO.Pipes"]

## ► Classes

	Class	Description
	<a href="#">Pipe</a>	The untyped base Pipe class
	<a href="#">PipeTMessage</a>	The base generic Pipe class
	<a href="#">PipeClient</a>	A pipe client. This class sends and receives byte arrays.
	<a href="#">PipeClientTMessage</a>	The generic version of PipeClient. This class expects all messages to be of the specified type.
	<a href="#">PipeHost</a>	Hosts a number of PipeServers. This class sends and receives byte arrays.
	<a href="#">PipeHostTMessage</a>	The generic version of PipeHost. This class expects all messages to be of the specified type.



## PipeLogger

Sends log messages to a remote server via a named pipe



## PipeReadWriteExtensions

Read/Write functionality for Pipe



## PipeServer

A Pipe server. This class sends and receives only byte arrays.



## PipeServerTMessage

The generic version of PipeServer. This class expects all messages to be of the specified type.

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Pipe Class

The untyped base Pipe class

## ► Inheritance Hierarchy

SystemObject W.IO.PipesPipe  
W.IO.PipesPipeTMessage

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public abstract class Pipe : IDisposable
```

The [Pipe](#) type exposes the following members.

## ► Constructors

	Name	Description
	<a href="#">Pipe</a>	Initializes a new instance of the <a href="#">Pipe</a> class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe



[InBufferSize](#)



[OutBufferSize](#)



[Stream](#)

The PipeStream on which to send and receive data

[Top](#)

## Methods

	Name	Description
≡▼	<a href="#">Dispose</a>	Disconnects and disposes the pipe
≡▼	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡▼	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡▼	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡▼	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡▼	<a href="#">Listen</a>	Continuously waits for data from the pipe. The

		BytesReceived event is raised when data arrives, then immediately waits for more data.
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disconnects and disposes the pipe
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data.
💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired.
💡	<a href="#">StopListening</a>	After the next bytes received, stops waiting for data
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Events

	Name	Description
	<a href="#">Disconnected</a>	Raised when the pipe has disconnected

[Top](#)

## Fields

	Name	Description
	<a href="#">_shouldListen</a>	

[Top](#)

## Extension Methods

	Name	Description
 	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)

⬇️	InLock(Action)	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	InLockTType(FuncTType)	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	InLockAsync(Action)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	InLockAsyncTType(FuncTType)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	IsDirty	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
⬇️	Lock	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	MarkAsClean	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Pipe Constructor

Initializes a new instance of the [Pipe](#) class

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
protected Pipe()
```

## ► See Also

Reference

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Pipe Properties

The [Pipe](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe
	<a href="#">InBufferSize</a>	
	<a href="#">OutBufferSize</a>	
	<a href="#">Stream</a>	The PipeStream on which to send and receive data

[Top](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeId Property

A unique id for this Pipe

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public string Id { get; }
```

Property Value

Type: [String](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeInBufferSize Property

[Missing <summary> documentation for "P:W.IO.Pipes.Pipe.InBufferSize"]

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

[Copy](#)

```
public static Nullable<int> InBufferSize { get; }
```

Property Value

Type: [NullableInt32](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeOutBufferSize Property

[Missing <summary> documentation for "P:W.IO.Pipes.Pipe.OutBufferSize"]

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public static Nullable<int> OutBufferSize { get;
```

Property Value

Type: [NullableInt32](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeStream Property

The PipeStream on which to send and receive data

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public PipeStream Stream { get; protected set; }
```

Property Value

Type: [PipeStream](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Pipe Methods

The [Pipe](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Dispose</a>	Disconnects and disposes the pipe
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Listen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives,

		then immediately waits for more data.
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disconnects and disposes the pipe
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data.
💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired.
💡	<a href="#">StopListening</a>	After the next bytes received, stops waiting for data
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ► Extension Methods

	Name	Description
 	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded.

Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---



### [IsDirty](#)

Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---



### [Lock](#)

Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)

---



### [MarkAsClean](#)

Scans each field and property of type IProperty and sets its IsDirty flag to false  
(Defined by [PropertyHostExtensions](#).)

---



### [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ► See Also

**Reference**

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeDispose Method

Disconnects and disposes the pipe

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements

[IDisposableDispose](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeListen Method

Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data.

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

```
public void Listen()
```

[Copy](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeOnDispose Method

Disconnects and disposes the pipe

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
protected virtual void OnDispose()
```

## ► See Also

Reference

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeOnListen Method

Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data.

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
protected virtual void OnListen()
```

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeOnStopListening Method

After the next bytes received, stops waiting for data

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
protected virtual void OnStopListening()
```

## ► See Also

Reference

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeRaiseDisconnection Method

Raises the Disconnected event. Pass in an exception if desired.

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
protected void RaiseDisconnection(  
    Pipe sender,  
    Exception e  
)
```

### Parameters

*sender*

Type: [W.IO.PipesPipe](#)

A reference to the caller

*e*

Type: [SystemException](#)

An exception if one was captured

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeStopListening Method

After the next bytes received, stops waiting for data

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public void StopListening()
```

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Pipe Events

The [Pipe](#) type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">Disconnected</a>	Raised when the pipe has disconnected

[Top](#)

## ▪ See Also

Reference

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeDisconnected Event

Raised when the pipe has disconnected

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public event Action<Pipe, Exception> Disconnected
```

Value

Type: [SystemActionPipe, Exception](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Pipe Fields

The [Pipe](#) type exposes the following members.

## ► Fields

Name	Description
 <a href="#">_shouldListen</a>	

[Top](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Pipe\_shouldListen Field

[Missing <summary> documentation for "F:W.IO.Pipes.Pipe.\_shouldListen"]

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

```
protected bool _shouldListen
```

[Copy](#)

Field Value

Type: [Boolean](#)

## ► See Also

[Reference](#)

[Pipe Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeTMessage Class

The base generic Pipe class

## ► Inheritance Hierarchy

```
SystemObject W.IO.PipesPipe
  W.IO.PipesPipeTMessage
    W.IO.PipesPipeClientTMessage
    W.IO.PipesPipeServerTMessage
```

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public abstract class Pipe<TMessage> : Pipe
```

### Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for  
"T:W.IO.Pipes.Pipe`1"]

The [PipeTMessage](#) type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">PipeTMessage</a>	Initializes a new instance of the <a href="#">PipeTMessage</a> class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function.

		(Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Listen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Overrides <a href="#">PipeOnListen</a> .)
💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)

💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseMessageReceived</a>	Called by the PipeExtensions extension methods to raise the MessageReceived event
🌐	<a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
🌐	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Events

	Name	Description
⚡	<a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)
⚡	<a href="#">MessageReceived</a>	Raised when a message has been received

[Top](#)

## Fields

	Name	Description
💡	<a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

[Top](#)

## Extension Methods

	Name	Description
💡	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
💡	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
💡	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">ReadTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">ReadAsyncTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)

[Unlock](#) Performs a Monitor unlock.  
(Defined by [MonitorExtensions](#).)

---

 [WriteTMessage](#) Write a message to the pipe  
(Defined by [PipeReadWriteExtensions](#))

---

 [WriteAsyncTMessage](#) Asynchronously write a message to the pipe  
(Defined by [PipeReadWriteExtensions](#))

---

[Top](#)

## ▲ See Also

[Reference](#)  
[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeTMessage Constructor

Initializes a new instance of the [PipeTMessage](#) class

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public Pipe()
```

## ► See Also

Reference

[PipeTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeTMessage Properties

The [PipeTMessage](#) generic type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PipeTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeTMessage Methods

The [PipeTMessage](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
<a href="#">Listen</a>	Continuously waits for data from the pipe. The

		BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Overrides <a href="#">PipeOnListen</a> .)
💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseMessageReceived</a>	Called by the

		PipeExtensions extension methods to raise the MessageReceived event
≡▼	<a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
≡▼	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Extension Methods

Name	Description
≡▼ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
≡▼ <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
≡▼ <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a

		Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)	
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor loc (Defined by <a href="#">MonitorExtensions</a> .)	
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor loc (Defined by <a href="#">MonitorExtensions</a> .)	
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)	
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)	
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets it's IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)	

	<a href="#">ReadTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
⬇	<a href="#">ReadAsyncTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
⬇	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
⬇	<a href="#">WriteTMessage</a>	Write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
⬇	<a href="#">WriteAsyncTMessage</a>	Asynchronously write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )

[Top](#)

## ▲ See Also

### Reference

[PipeTMessage Class](#)

[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeTMessageOnListen Method

Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data.

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
protected override void OnListen()
```

## ► See Also

Reference

[PipeTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeTMessageRaiseMessageReceived Method

Called by the PipeExtensions extension methods to raise the MessageReceived event

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
protected void RaiseMessageReceived(  
    Pipe<TMessage> pipe,  
    TMessage message  
)
```

## Parameters

*pipe*

Type: [W.IO.PipesPipeTMessage](#)

[Missing <param name="pipe"/> documentation for  
"M:W.IO.Pipes.Pipe`1.RaiseMessageReceived(W.IO.Pipes.Pipe{`0},`0)"]

*message*

Type: [TMessage](#)

[Missing <param name="message"/> documentation for  
"M:W.IO.Pipes.Pipe`1.RaiseMessageReceived(W.IO.Pipes.Pipe{`0},`0)"]

## ► See Also

[Reference](#)

[PipeTMessage Class](#)

## [W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeTMessage Events

The [PipeTMessage](#) generic type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)
	<a href="#">MessageReceived</a>	Raised when a message has been received

[Top](#)

## ▪ See Also

[Reference](#)

[PipeTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeTMessageMessageReceived Event

Raised when a message has been received

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public event Action<Pipe<TMessage>, TMessage> MessageReceived;
```

Value

Type: [SystemActionPipe<TMessage, TMessage>](#)

## ► Remarks

This event will only be raised if TMessage != byte[]

## ► See Also

Reference

[PipeTMessage Class](#)

[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeTMessage Fields

The [PipeTMessage](#) generic type exposes the following members.

## ► Fields

Name	Description
 <a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PipeTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClient Class

A pipe client. This class sends and receives byte arrays.

## ► Inheritance Hierarchy

```
SystemObject W.IO.PipesPipe
  W.IO.PipesPipeByte
    W.IO.PipesPipeClientByte
      W.IO.PipesPipeClient
```

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public class PipeClient : PipeClient<byte[]>
```

The [PipeClient](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PipeClient</a>	Initializes a new instance of the <a href="#">PipeClient</a> class

[Top](#)

## ► Properties

	Name	Description
--	------	-------------

	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## Methods

	Name	Description
	<a href="#">Connect</a>	Attempts to connect the pipe to a pipe server (Inherited from <a href="#">PipeClientTMessage</a> .)
	<a href="#">ConnectAsync</a>	Attempts to asynchronously connect the pipe to a pipe server (Inherited from <a href="#">PipeClientTMessage</a> .)
	<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is

		reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Listen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">PipeTMessage</a> .)

---

💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseMessageReceived</a>	Called by the PipeExtensions extension methods to raise the MessageReceived event (Inherited from <a href="#">PipeTMessage</a> .)
💡	<a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## Events

	Name	Description
⚡	<a href="#">Connected</a>	Raised when a connection

---

		attempt succeeds (Inherited from <a href="#">PipeClientTMessage</a> .)
	<a href="#">ConnectionFailed</a>	Raised when a connection attempt fails (Inherited from <a href="#">PipeClientTMessage</a> .)
	<a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)
	<a href="#">MessageReceived</a>	Raised when a message has been received (Inherited from <a href="#">PipeTMessage</a> .)

---

[Top](#)

## Fields

	Name	Description
	<a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

[Top](#)

## Extension Methods

	Name	Description
 	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> )
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and

sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members  
(Defined by [PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)

	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
✖	<a href="#">MarkAsClean</a>	Scans each field and property of type <code>IProperty</code> and sets its <code>IsDirty</code> flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
✖	<a href="#">ReadByte</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
✖	<a href="#">ReadAsyncByte</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
✖	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
✖	<a href="#">WriteByte</a>	Write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
✖	<a href="#">WriteAsyncByte</a>	Asynchronously write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )

[Top](#)

## See Also

## Reference

### [W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClient Constructor

Initializes a new instance of the [PipeClient](#) class

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public PipeClient()
```

## ► See Also

Reference

[PipeClient Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClient Properties

The [PipeClient](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PipeClient Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClient Methods

The [PipeClient](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Connect</a>	Attempts to connect the pipe to a pipe server (Inherited from <a href="#">PipeClientTMessage</a> .)
<a href="#">ConnectAsync</a>	Attempts to asynchronously connect the pipe to a pipe server (Inherited from <a href="#">PipeClientTMessage</a> .)
<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.

		(Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Listen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">PipeTMessage</a> .)

	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseMessageReceived</a>	Called by the PipeExtensions extension methods to raise the MessageReceived event (Inherited from <a href="#">PipeTMessage</a> .)
💡	<a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ► Extension Methods

Name	Description
💡 	<a href="#">AsTType</a> Use Generic syntax for the as operator.

(Defined by [AsExtensions](#))

 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type <a href="#">IProperty</a>

(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type <code>IProperty</code> and sets its <code>IsDirty</code> flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">ReadByte</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
 <a href="#">ReadAsyncByte</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">WriteByte</a>	Write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
 <a href="#">WriteAsyncByte</a>	Asynchronously write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )

[Top](#)

## ▲ See Also

Reference

[PipeClient Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClient Events

The [PipeClient](#) type exposes the following members.

## Events

	Name	Description
⚡	<a href="#">Connected</a>	Raised when a connection attempt succeeds (Inherited from <a href="#">PipeClientTMessage</a> .)
⚡	<a href="#">ConnectionFailed</a>	Raised when a connection attempt fails (Inherited from <a href="#">PipeClientTMessage</a> .)
⚡	<a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)
⚡	<a href="#">MessageReceived</a>	Raised when a message has been received (Inherited from <a href="#">PipeTMessage</a> .)

[Top](#)

## See Also

[Reference](#)

[PipeClient Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClient Fields

The [PipeClient](#) type exposes the following members.

## ► Fields

	Name	Description
💡	<a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PipeClient Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessage Class

The generic version of PipeClient. This class expects all messages to be of the specified type.

## ► Inheritance Hierarchy

```
SystemObject W.IO.PipesPipe
W.IO.PipesPipeTMessage
W.IO.PipesPipeClientTMessage
W.IO.PipesPipeClient
```

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public class PipeClient<TMessage> : Pipe<TMessage>
```

Type Parameters

*TMessage*

The message type to send and receive

The [PipeClientTMessage](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PipeClientTMessage</a>	Initializes a new instance of the <a href="#">PipeClientTMessage</a>

class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Connect</a>	Attempts to connect the pipe to a pipe server
	<a href="#">ConnectAsync</a>	Attempts to asynchronously connect the pipe to a pipe server
 	<a href="#">Create</a>	Creates a new PipeClient and attempts to connect the pipe to a pipe server
 	<a href="#">CreateAsync</a>	Creates a new PipeClient and attempts to asynchronously connect the pipe to a

		pipe server
	<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">Listen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">Pipe</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy

		of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">PipeTMessage</a> .)
💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseMessageReceived</a>	Called by the PipeExtensions extension methods to raise the MessageReceived event (Inherited from <a href="#">PipeTMessage</a> .)

≡	<a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Events

	Name	Description
⚡	<a href="#">Connected</a>	Raised when a connection attempt succeeds
⚡	<a href="#">ConnectionFailed</a>	Raised when a connection attempt fails
⚡	<a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)
⚡	<a href="#">MessageReceived</a>	Raised when a message has been received (Inherited from <a href="#">PipeTMessage</a> .)

[Top](#)

## Fields

	Name	Description
💡	<a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

---

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">ReadTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
 <a href="#">ReadAsyncTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">WriteTMessage</a>	Write a message to the pipe

(Defined by  
[PipeReadWriteExtensions](#))

---



## [WriteAsyncTMessage](#)

Asynchronously write a message to the pipe  
(Defined by  
[PipeReadWriteExtensions](#))

---

[Top](#)

## ► See Also

[Reference](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessage Constructor

Initializes a new instance of the [PipeClientTMessage](#) class

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public PipeClient()
```

## ► See Also

Reference

[PipeClientTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessage Properties

The [PipeClientTMessage](#) generic type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

### Reference

[PipeClientTMessage Class](#)  
[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessage Methods

The [PipeClientTMessage](#) generic type exposes the following members.

## ▲ Methods

	Name	Description
≡ ♀	<a href="#">Connect</a>	Attempts to connect the pipe to a pipe server
≡ ♀	<a href="#">ConnectAsync</a>	Attempts to asynchronously connect the pipe to a pipe server
≡ ♀ S	<a href="#">Create</a>	Creates a new PipeClient and attempts to connect the pipe to a pipe server
≡ ♀ S	<a href="#">CreateAsync</a>	Creates a new PipeClient and attempts to asynchronously connect the pipe to a pipe server
≡ ♀	<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
≡ ♀	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object.

		(Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Listen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is

		raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">PipeTMessage</a> .)
💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseMessageReceived</a>	Called by the PipeExtensions extension methods to raise the MessageReceived event (Inherited from <a href="#">PipeTMessage</a> .)
💡	<a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the operator. (Defined by <a href="#">AsExtensions</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs

the action in a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

 [IsDirty](#) Scans the IsDirty value of each field and property of type IProperty  
(Defined by  
[PropertyHostExtensions](#).)

---

 [Lock](#) Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

 [MarkAsClean](#) Scans each field and property of type IProperty and sets its IsDirty flag to false  
(Defined by  
[PropertyHostExtensions](#).)

---

 [ReadTMessage](#) Waits for a message to be read from the pipe  
(Defined by  
[PipeReadWriteExtensions](#))

---

 [ReadAsyncTMessage](#) Waits for a message to be read from the pipe  
(Defined by  
[PipeReadWriteExtensions](#))

---

 [Unlock](#) Performs a Monitor unlock  
(Defined by  
[MonitorExtensions](#).)

---

 [WriteTMessage](#) Write a message to the pipe  
(Defined by  
[PipeReadWriteExtensions](#))



## [WriteAsyncTMessage](#)

Asynchronously write a message to the pipe  
(Defined by [PipeReadWriteExtensions](#))

---

[Top](#)

## ▲ See Also

[Reference](#)

[PipeClientTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessageConnect Method

Attempts to connect the pipe to a pipe server

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public bool Connect(
    string serverName,
    string pipeName,
    TokenImpersonationLevel tokenImpersonationLevel,
    int msTimeout
)
```

### Parameters

*serverName*

Type: [System.String](#)

The name or ip of the machine hosting the server pipe

*pipeName*

Type: [System.String](#)

The name of the pipe

*tokenImpersonationLevel*

Type: [System.Security.PrincipalTokenImpersonationLevel](#)

The impersonation type for the pipe to use

*msTimeout*

Type: [System.Int32](#)

The maximum amount of time, in milliseconds, to wait for the

server to connect

#### Return Value

Type: [Boolean](#)

True if a connection was established, otherwise False

## See Also

### Reference

[PipeClientTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessageConnectAsync Method

Attempts to asynchronously connect the pipe to a pipe server

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public Task<bool> ConnectAsync(
    string serverName,
    string pipeName,
    TokenImpersonationLevel tokenImpersonationLevel,
    int msTimeout
)
```

## Parameters

*serverName*

Type: [System.String](#)

The name or ip of the machine hosting the server pipe

*pipeName*

Type: [System.String](#)

The name of the pipe

*tokenImpersonationLevel*

Type: [System.Security.PrincipalTokenImpersonationLevel](#)

The impersonation type for the pipe to use

*msTimeout*

Type: [System.Int32](#)

The maximum amount of time, in milliseconds, to wait for the

server to connect

#### Return Value

Type: [TaskBoolean](#)

True if a connection was established, otherwise False

## See Also

### Reference

[PipeClientTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessageCreate Method

Creates a new PipeClient and attempts to connect the pipe to a pipe server

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public static PipeClient Create(
    string serverName,
    string pipeName,
    TokenImpersonationLevel tokenImpersonationLevel,
    int msTimeout
)
```

## Parameters

*serverName*

Type: [SystemString](#)

The name or ip of the machine hosting the server pipe

*pipeName*

Type: [SystemString](#)

The name of the pipe

*tokenImpersonationLevel*

Type: [System.Security.PrincipalTokenImpersonationLevel](#)

The impersonation type for the pipe to use

*msTimeout*

Type: [SystemInt32](#)

The maximum amount of time, in milliseconds, to wait for the server to connect

## Return Value

Type: [PipeClient](#)

True if a connection was established, otherwise False

## ▲ See Also

Reference

[PipeClientTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessageCreateAsync Method

Creates a new PipeClient and attempts to asynchronously connect the pipe to a pipe server

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public static Task<Pipe> CreateAsync(
    string serverName,
    string pipeName,
    TokenImpersonationLevel tokenImpersonationLevel,
    int msTimeout
)
```

## Parameters

*serverName*

Type: [System.String](#)

The name or ip of the machine hosting the server pipe

*pipeName*

Type: [System.String](#)

The name of the pipe

*tokenImpersonationLevel*

Type: [System.Security.PrincipalTokenImpersonationLevel](#)

The impersonation type for the pipe to use

*msTimeout*

Type: [System.Int32](#)

The maximum amount of time, in milliseconds, to wait for the server to connect

## Return Value

Type: [TaskPipe](#)

True if a connection was established, otherwise False

## ◀ See Also

Reference

[PipeClientTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessage Events

The [PipeClientTMessage](#) generic type exposes the following members.

## Events

	Name	Description
	<a href="#">Connected</a>	Raised when a connection attempt succeeds
	<a href="#">ConnectionFailed</a>	Raised when a connection attempt fails
	<a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)
	<a href="#">MessageReceived</a>	Raised when a message has been received (Inherited from <a href="#">PipeTMessage</a> .)

[Top](#)

## See Also

### Reference

[PipeClientTMessage Class](#)  
[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeClientTMessageConnected Event

Raised when a connection attempt succeeds

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public event Action<Pipe> Connected
```

Value

Type: [SystemActionPipe](#)

## ► See Also

Reference

[PipeClientTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessageConnectionFailed Event

Raised when a connection attempt fails

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public event Action<Pipe, Exception> ConnectionFailed;
```

Value

Type: [SystemActionPipe, Exception](#)

## ► See Also

Reference

[PipeClientTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeClientTMessage Fields

The [PipeClientTMessage](#) generic type exposes the following members.

## ► Fields

	Name	Description
	<a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

Reference

[PipeClientTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHost Class

Hosts a number of PipeServers. This class sends and receives byte arrays.

## ► Inheritance Hierarchy

[SystemObject](#) [W.IO.PipesPipeHostByte](#)  
[W.IO.PipesPipeHost](#)

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public class PipeHost : PipeHost<byte[]>
```

The [PipeHost](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PipeHost</a>	Initializes a new instance of the <a href="#">PipeHost</a> class

[Top](#)

## ► Methods

	Name	Description
≡		

	<a href="#">Dispose</a>	Stops the host and releases resources (Inherited from <a href="#">PipeHostTMessage</a> .)
≡▼	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡▼	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡▼	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡▼	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡▼	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡▼	<a href="#">Start</a>	Creates the specified number of pipe servers and starts listening for clients (Inherited from <a href="#">PipeHostTMessage</a> .)
≡▼	<a href="#">Stop</a>	Disconnects and disposes all of the pipe servers (Inherited from

[PipeHostTMessage.\)](#)

---

 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
--	--

---

[Top](#)

## Events

Name	Description
 <a href="#">MessageReceived</a>	Raised when a pipe server has received data from a client (Inherited from <a href="#">PipeHostTMessage</a> .)

---

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by

[PropertyHostExtensions.\)](#)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	Scans each field and

## [MarkAsClean](#)

property of type [IProperty](#)  
and sets it's `IsDirty` flag to  
`false`

(Defined by  
[PropertyHostExtensions](#).)



## [Unlock](#)

Performs a Monitor  
unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHost Constructor

Initializes a new instance of the [PipeHost](#) class

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public PipeHost()
```

## ► See Also

Reference

[PipeHost Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHost Methods

The [PipeHost](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Dispose</a>	Stops the host and releases resources (Inherited from <a href="#">PipeHostTMessage</a> .)
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> .

(Inherited from [Object](#).)

 <a href="#">Start</a>	Creates the specified number of pipe servers and starts listening for clients (Inherited from <a href="#">PipeHostTMessage</a> .)
 <a href="#">Stop</a>	Disconnects and disposes all of the pipe servers (Inherited from <a href="#">PipeHostTMessage</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ► Extension Methods

Name	Description
 <a href="#">AsTTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)

•	InLock(Action)	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	InLockTType(FuncTType)	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	InLockAsync(Action)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	InLockAsyncTType(FuncTType)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	IsDirty	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
•	Lock	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	MarkAsClean	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[PipeHost Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHost Events

The [PipeHost](#) type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">MessageReceived</a>	Raised when a pipe server has received data from a client (Inherited from <a href="#">PipeHostTMessage</a> .)

[Top](#)

## ▪ See Also

[Reference](#)

[PipeHost Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHostTMessage Class

The generic version of PipeHost. This class expects all messages to be of the specified type.

## ► Inheritance Hierarchy

[SystemObject](#) [W.IO.PipesPipeHostTMessage](#)  
[W.IO.PipesPipeHost](#)

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public class PipeHost<TMessage> : IDisposable
```

### Type Parameters

*TMessage*

The message type to send and receive

The [PipeHostTMessage](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PipeHostTMessage</a>	Initializes a new instance of the <a href="#">PipeHostTMessage</a> class

[Top](#)

## Methods

Name	Description
<a href="#">Dispose</a>	Stops the host and releases resources
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
<a href="#">Start</a>	Creates the specified number of pipe servers and starts listening for clients
<a href="#">Stop</a>	Disconnects and disposes all

of the pipe servers

---

 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
--	--

---

[Top](#)

## Events

Name	Description
 <a href="#">MessageReceived</a>	Raised when a pipe server has received data from a client

---

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)

---

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHostTMessage Constructor

Initializes a new instance of the [PipeHostTMessage](#) class

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public PipeHost()
```

## ► See Also

Reference

[PipeHostTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHostTMessage Methods

The [PipeHostTMessage](#) generic type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Dispose</a>	Stops the host and releases resources
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

	<a href="#">Start</a>	Creates the specified number of pipe servers and starts listening for clients
	<a href="#">Stop</a>	Disconnects and disposes all of the pipe servers
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ◀ See Also

[Reference](#)

[PipeHostTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHostTMessageDispose Method

Stops the host and releases resources

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements  
[IDisposable.Dispose](#)

## ► See Also

Reference

[PipeHostTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHostTMessageStart Method

Creates the specified number of pipe servers and starts listening for clients

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

[Copy](#)

```
public uint Start(  
    string pipeName,  
    int maxConnections  
)
```

## Parameters

*pipeName*

Type: [SystemString](#)

The name of the pipe

*maxConnections*

Type: [SystemInt32](#)

The maximum number of pipe servers to create.

## Return Value

Type: [UInt32](#)

[Missing <returns> documentation for  
"M:W.IO.Pipes.PipeHost`1.Start(System.String,System.Int32)"]

## ► Remarks

Because PipeHost creates a PipeServer for each possible

connection, this value cannot be negative. This breaks the standard paradigm for pipes. If this does not work for you, use PipeServer instead.

## ▲ See Also

Reference

[PipeHostTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHostTMessageStop Method

Disconnects and disposes all of the pipe servers

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public void Stop()
```

## ► See Also

Reference

[PipeHostTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHostTMessage Events

The [PipeHostTMessage](#) generic type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">MessageReceived</a>	Raised when a pipe server has received data from a client

[Top](#)

## ▪ See Also

[Reference](#)

[PipeHostTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeHostTMessageMessageReceivedEvent

Raised when a pipe server has received data from a client

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public event Action<PipeHost<TMessage>, Pipe<TMessage>,
    TMessage> MessageReceived;
```

Value

Type: [SystemActionPipeHostTMessage, PipeTMessage, TMessage](#)

## ► See Also

Reference

[PipeHostTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeLogger Class

Sends log messages to a remote server via a named pipe

## ► Inheritance Hierarchy

[SystemObject](#) [W.IO.PipesPipeLogger](#)

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

[Copy](#)

```
public static class PipeLogger
```

The PipeLogger type exposes the following members.

## ► Properties

	Name	Description
 	<a href="#">PipeClient</a>	The named pipe client used to send log messages

[Top](#)

## ► Methods

	Name	Description
 	<a href="#">LogTheMessage</a>	Log a message to the remote machine. Note that this

message is slower due to  
parsing the remoteIP with each  
call

---

[Top](#)

## ▲ Examples

```
Log.LogTheMessage += (category, message) =>
    W.IO.Pipes.PipeLogger.LogTheMessage(".", "PipeLogger", true,
        category, message);
```

## ▲ See Also

Reference

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeLogger Properties

The [PipeLogger](#) type exposes the following members.

## Properties

	Name	Description
 	<a href="#">PipeClient</a>	The named pipe client used to send log messages

[Top](#)

## See Also

Reference

[PipeLogger Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeLoggerPipeClient Property

The named pipe client used to send log messages

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public static PipeClient PipeClient { get; }
```

Property Value

Type: [PipeClient](#)

## ► See Also

[Reference](#)

[PipeLogger Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeLogger Methods

The [PipeLogger](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">LogTheMessage</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

[Top](#)

## ▪ See Also

[Reference](#)

[PipeLogger Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeLoggerLogTheMessage Method

Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

[Copy](#)

```
public static void LogTheMessage(  
    string server,  
    string pipeName,  
    string message  
)
```

## Parameters

*server*

Type: [SystemString](#)

The name of the server hosting the named pipe

*pipeName*

Type: [SystemString](#)

The name of the named pipe

*message*

Type: [SystemString](#)

The log message

## ► See Also

Reference

[PipeLogger Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeReadWriteExtensions Class

Read/Write functionality for Pipe

## ► Inheritance Hierarchy

[SystemObject](#) [W.IO.PipesPipeReadWriteExtensions](#)

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

[Copy](#)

```
public static class PipeReadWriteExtensions
```

The PipeReadWriteExtensions type exposes the following members.

## ► Methods

	Name	Description
≡  	<a href="#">ReadTMessage</a>	Waits for a message to be read from the pipe
≡  	<a href="#">ReadAsyncTMessage</a>	Waits for a message to be read from the pipe
≡  	<a href="#">WriteTMessage</a>	Write a message to the pipe
≡  	<a href="#">WriteAsyncTMessage</a>	Asynchronously write a message to the pipe

[Top](#)

## ◀ See Also

Reference

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeReadWriteExtensions Methods

The [PipeReadWriteExtensions](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">ReadTMessage</a>	Waits for a message to be read from the pipe
 	<a href="#">ReadAsyncTMessage</a>	Waits for a message to be read from the pipe
 	<a href="#">WriteTMessage</a>	Write a message to the pipe
 	<a href="#">WriteAsyncTMessage</a>	Asynchronously write a message to the pipe

[Top](#)

## ▪ See Also

Reference

[PipeReadWriteExtensions Class](#)

[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeReadWriteExtensionsReadTMe Method

Waits for a message to be read from the pipe

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public static TMMessage Read<TMMessage>(
    this Pipe<TMMessage> pipe
)
```

### Parameters

*pipe*

Type: [W.IO.PipesPipe](#) *TMessage*

The pipe from which to read data

### Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for  
"M:W.IO.Pipes.PipeReadWriteExtensions.Read`1(W.IO.Pipes.Pipe{`0})"]

### Return Value

Type: [TMessage](#)

The message received or null if the read failed (the pipe was closed)

### Usage Note

In Visual Basic and C#, you can call this method as an instance

method on any object of type [PipeTMessage](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

[Reference](#)

[PipeReadWriteExtensions Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeReadWriteExtensionsReadAsynchronous Method

Waits for a message to be read from the pipe

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public static Task<TMessage> ReadAsync<TMessage>(
    this Pipe<TMessage> pipe
)
```

### Parameters

*pipe*

Type: [W.IO.PipesPipe](#) *TMessage*  
The pipe from which to read data

### Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for  
"M:W.IO.Pipes.PipeReadWriteExtensions.ReadAsync`1(W.IO.Pipes.Pipe{`0})"]

### Return Value

Type: [Task](#) *TMessage*

The message received or null if the read failed (the pipe was closed)

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [PipeTMessage](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[PipeReadWriteExtensions Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeReadWriteExtensionsWriteTMessage Method

Write a message to the pipe

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C#  
public static bool Write<TMessage>(  
    this Pipe<TMessage> pipe,  
    TMessage message  
)
```

[Copy](#)

### Parameters

*pipe*

Type: [W.IO.PipesPipe](#) *TMessage*

The pipe on which to write data

*message*

Type: *TMessage*

The message to write

### Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for  
"M:W.IO.Pipes.PipeReadWriteExtensions.Write``1(W.IO.Pipes.Pipe{``0},``0)"]

### Return Value

Type: [Boolean](#)

True if the message was sent successfully, otherwise false

#### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [PipeTMessage](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[PipeReadWriteExtensions Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeReadWriteExtensions.WriteAsy<sup>r</sup> Method

Asynchronously write a message to the pipe

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public static Task<bool> WriteAsync<TMessage>(
    this Pipe<TMessage> pipe,
    TMessage message
)
```

### Parameters

*pipe*

Type: [W.IO.PipesPipe`1](#)  
*TMessage*

The pipe on which to write data

*message*

Type: [TMessage](#)

The message to write

### Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for  
"M:W.IO.Pipes.PipeReadWriteExtensions.WriteAsync`1(W.IO.Pipes.Pipe{`0},`0)"

### Return Value

Type: [TaskBoolean](#)

True if the message was sent successfully, otherwise false

#### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [PipeTMessage](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[PipeReadWriteExtensions Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServer Class

A Pipe server. This class sends and receives only byte arrays.

## ► Inheritance Hierarchy

```
SystemObject W.IO.PipesPipe
  W.IO.PipesPipeByte
    W.IO.PipesPipeServerByte
      W.IO.PipesPipeServer
```

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public class PipeServer : PipeServer<byte[]>
```

The [PipeServer](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PipeServer</a>	Initializes a new instance of the PipeServer class

[Top](#)

## ► Properties

	Name	Description
--	------	-------------

	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ▲ Methods

	Name	Description
	<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)

≡	<a href="#">Listen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disposes the PipeServer and release resources (Inherited from <a href="#">PipeServerTMessage</a> .)
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">PipeTMessage</a> .)
💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseMessageReceived</a>	Called by the PipeExtensions

extension methods to raise the MessageReceived event  
(Inherited from [PipeTMessage](#).)

---

 <a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">WaitForConnection</a>	Creates a new NamedPipeServerStream and Waits for a client to connect (Inherited from <a href="#">PipeServerTMessage</a> .)

---

[Top](#)

## Events

Name	Description
 <a href="#">Connected</a>	Raised when a client has connected to the server (Inherited from <a href="#">PipeServerTMessage</a> .)
 <a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)



[MessageReceived](#) Raised when a message has been received  
(Inherited from [PipeTMessage](#).)



[StartException](#) Raised if an exception occurs while creating the NamedPipeServerStream  
(Inherited from [PipeServerTMessage](#).)

[Top](#)

## Fields

Name	Description
<a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

[Top](#)

## Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> )
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by

[PropertyHostExtensions.\)](#)

 <a href="#">ReadByte</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">ReadAsyncByte</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">WriteByte</a>	Write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">WriteAsyncByte</a>	Asynchronously write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)

[Top](#)

## ▲ See Also

[Reference](#)

[W.IO.Pipes Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServer Constructor

Initializes a new instance of the [PipeServer](#) class

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public PipeServer()
```

## ► See Also

Reference

[PipeServer Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServer Properties

The [PipeServer](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PipeServer Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServer Methods

The [PipeServer](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
<a href="#">Listen</a>	Continuously waits for data from the pipe. The

		BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disposes the PipeServer and release resources (Inherited from <a href="#">PipeServerTMessage</a> .)
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">PipeTMessage</a> .)
💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseMessageReceived</a>	Called by the PipeExtensions extension methods to raise the

MessageReceived event  
(Inherited from  
[PipeTMessage](#).)

 <a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">WaitForConnection</a>	Creates a new NamedPipeServerStream and Waits for a client to connect (Inherited from <a href="#">PipeServerTMessage</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members

(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor loc (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor loc (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty

and sets it's IsDirty flag to false  
(Defined by [PropertyHostExtensions](#).)

---

 <a href="#">ReadByte</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">ReadAsyncByte</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">WriteByte</a>	Write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">WriteAsyncByte</a>	Asynchronously write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)

---

[Top](#)

## ▲ See Also

[Reference](#)

[PipeServer Class](#)

[W.IO.Pipes Namespace](#)



## Tungsten

W<sup>74</sup>

# PipeServer Events

The [PipeServer](#) type exposes the following members.

## Events

Name	Description
 <a href="#">Connected</a>	Raised when a client has connected to the server (Inherited from <a href="#">PipeServerTMessage</a> .)
 <a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)
 <a href="#">MessageReceived</a>	Raised when a message has been received (Inherited from <a href="#">PipeTMessage</a> .)
 <a href="#">StartException</a>	Raised if an exception occurs while creating the NamedPipeServerStream (Inherited from <a href="#">PipeServerTMessage</a> .)

[Top](#)

## See Also

[Reference](#)

[PipeServer Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServer Fields

The [PipeServer](#) type exposes the following members.

## ► Fields

Name	Description
 <a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PipeServer Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServerTMessage Class

The generic version of PipeServer. This class expects all messages to be of the specified type.

## ► Inheritance Hierarchy

```
SystemObject W.IO.PipesPipe
W.IO.PipesPipeTMessage
W.IO.PipesPipeServerTMessage
W.IO.PipesPipeServer
```

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

```
C# Copy
public class PipeServer<TMessage> : Pipe<TMessage>
```

Type Parameters

*TMessage*

The message type to send and receive

The [PipeServerTMessage](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">PipeServerTMessage</a>	Initializes a new instance of the

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)

---

≡♪	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡♪	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡♪	<a href="#">Listen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">Pipe</a> .)
💡♪	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡♪	<a href="#">OnDispose</a>	Disposes the PipeServer and release resources (Overrides <a href="#">PipeOnDispose</a> .)
💡♪	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">PipeTMessage</a> .)
💡♪	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data

		(Inherited from <a href="#">Pipe</a> .)
⚡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
⚡	<a href="#">RaiseMessageReceived</a>	Called by the PipeExtensions extension methods to raise the MessageReceived event (Inherited from <a href="#">PipeTMessage</a> .)
≡⚡	<a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
≡⚡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
≡⚡	<a href="#">WaitForConnection</a>	Creates a new NamedPipeServerStream and Waits for a client to connect

[Top](#)

## Events

	Name	Description
⚡	<a href="#">Connected</a>	Raised when a client has connected to the server

---

	<a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)
	<a href="#">MessageReceived</a>	Raised when a message has been received (Inherited from <a href="#">PipeTMessage</a> .)
	<a href="#">StartException</a>	Raised if an exception occurs while creating the NamedPipeServerStream

---

[Top](#)

## Fields

Name	Description
 <a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

---

[Top](#)

## Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members

(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty

and sets it's IsDirty flag to false  
(Defined by [PropertyHostExtensions](#).)

 <a href="#">ReadTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">ReadAsyncTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">WriteTMessage</a>	Write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">WriteAsyncTMessage</a>	Asynchronously write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)

[Top](#)

## ▲ See Also

[Reference](#)

[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeServerTMessage Constructor

Initializes a new instance of the [PipeServerTMessage](#) class

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
public PipeServer()
```

## ► See Also

[Reference](#)

[PipeServerTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServerTMessage Properties

The [PipeServerTMessage](#) generic type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Id</a>	A unique id for this Pipe (Inherited from <a href="#">Pipe</a> .)
	<a href="#">Stream</a>	The PipeStream on which to send and receive data (Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

### Reference

[PipeServerTMessage Class](#)  
[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServerTMessage Methods

The [PipeServerTMessage](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disconnects and disposes the pipe (Inherited from <a href="#">Pipe</a> .)
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
<a href="#">Listen</a>	Continuously waits for

data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data.  
(Inherited from [Pipe](#).)

💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnDispose</a>	Disposes the PipeServer and release resources (Overrides <a href="#">PipeOnDispose</a> .)
💡	<a href="#">OnListen</a>	Continuously waits for data from the pipe. The BytesReceived event is raised when data arrives, then immediately waits for more data. (Inherited from <a href="#">PipeTMessage</a> .)
💡	<a href="#">OnStopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseDisconnection</a>	Raises the Disconnected event. Pass in an exception if desired. (Inherited from <a href="#">Pipe</a> .)
💡	<a href="#">RaiseMessageReceived</a>	Called by the PipeExtensions extension methods to

raise the  
MessageReceived event  
(Inherited from  
[PipeTMessage](#).)

---

 <a href="#">StopListening</a>	After the next bytes received, stops waiting for data (Inherited from <a href="#">Pipe</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">WaitForConnection</a>	Creates a new NamedPipeServerStream and Waits for a client to connect

---

[Top](#)

## ◀ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">OwnedProperty</a> )

[PropertyHostExtensions.\)](#)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor loc (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor loc (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets it's IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---

 <a href="#">ReadTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">ReadAsyncTMessage</a>	Waits for a message to be read from the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">WriteTMessage</a>	Write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)
 <a href="#">WriteAsyncTMessage</a>	Asynchronously write a message to the pipe (Defined by <a href="#">PipeReadWriteExtensions</a> .)

---

[Top](#)

## ► See Also

[Reference](#)

[PipeServerTMessage Class](#)

[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeServerTMessageOnDispose Method

Disposes the PipeServer and release resources

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

Copy

```
protected override void OnDispose()
```

## ► See Also

Reference

[PipeServerTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServerTMessageWaitForConne Method

Creates a new NamedPipeServerStream and Waits for a client to connect

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

[Copy](#)

```
public bool WaitForConnection(  
    string pipeName,  
    int maxConnections = -1  
)
```

### Parameters

*pipeName*

Type: [SystemString](#)

The name of the pipe

*maxConnections (Optional)*

Type: [SystemInt32](#)

The maximum number of pipes with this name

### Return Value

Type: [Boolean](#)

True if the server was created and is waiting, otherwise False

## ► See Also

## Reference

[PipeServerTMessage Class](#)  
[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServerTMessage Events

The [PipeServerTMessage](#) generic type exposes the following members.

## Events

	Name	Description
	<a href="#">Connected</a>	Raised when a client has connected to the server
	<a href="#">Disconnected</a>	Raised when the pipe has disconnected (Inherited from <a href="#">Pipe</a> .)
	<a href="#">MessageReceived</a>	Raised when a message has been received (Inherited from <a href="#">PipeTMessage</a> .)
	<a href="#">StartException</a>	Raised if an exception occurs while creating the NamedPipeServerStream

[Top](#)

## See Also

Reference

[PipeServerTMessage Class](#)  
[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeServerTMessageConnected Event

Raised when a client has connected to the server

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

[Copy](#)

```
public event Action<Pipe> Connected
```

Value

Type: [SystemActionPipe](#)

## ► See Also

Reference

[PipeServerTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PipeServerTMessageStartException Event

Raised if an exception occurs while creating the NamedPipeServerStream

**Namespace:** [W.IO.Pipes](#)

**Assembly:** Tungsten.IO.Pipes (in Tungsten.IO.Pipes.dll) Version: 2.0.3

## ► Syntax

C#

[Copy](#)

```
public event Action<Pipe, Exception> StartException
```

Value

Type: [SystemActionPipe, Exception](#)

## ► See Also

Reference

[PipeServerTMessage Class](#)

[W.IO.Pipes Namespace](#)

## Tungsten

W<sup>74</sup>

# PipeServerTMessage Fields

The [PipeServerTMessage](#) generic type exposes the following members.

## ► Fields

	Name	Description
	<a href="#">_shouldListen</a>	(Inherited from <a href="#">Pipe</a> .)

[Top](#)

## ► See Also

[Reference](#)

[PipeServerTMessage Class](#)

[W.IO.Pipes Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# W.Logging Namespace

[Missing <summary> documentation for "N:W.Logging"]

## ► Classes

Class	Description
 <a href="#">Log</a>	Exposes static methods for logging. LogTheMessage can be assigned a new value for customized logging.
 <a href="#">LogMessageHistory</a>	Maintains a history of Log information
 <a href="#">LogMessageHistoryLogMessage</a>	An individual log message
 <a href="#">LogPropertyHost</a>	Provides

## ► Enumerations

Enumeration	Description
 <a href="#">LogLogMessageCategory</a>	The log message type

## Tungsten

W<sup>74</sup>

# Log Class

Exposes static methods for logging. LogTheMessage can be assigned a new value for customized logging.

## ► Inheritance Hierarchy

[SystemObject](#) [W.LoggingLog](#)

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public static class Log
```

The [Log](#) type exposes the following members.

## ► Properties

	Name	Description
 	<a href="#">AddTimestamp</a>	If True, log messages will be prefixed with a timestamp

[Top](#)

## ► Methods

	Name	Description
 	<a href="#">e(Exception)</a>	Log an Exception
 		

	<code>e(String, Object)</code>	Log a formatted exception message. This method uses string.Format to format the message.
 	<code>i(String, Object)</code>	Log a formatted informational message. This method uses string.Format to format the message.
 	<code>i(String, String, Int32)</code>	Log a formatted informational message. This method uses string.Format to format the message.
 	<code>v(String, Object)</code>	Log a formatted verbose message. This method uses string.Format to format the message.
 	<code>v(String, String, Int32)</code>	Log a formatted verbose message. This method uses string.Format to format the message.
 	<code>w(String, Object)</code>	Log a formatted warning message. This method uses string.Format to format the message.
 	<code>w(String, String, Int32)</code>	Log a formatted warning message. This method uses string.Format to format the message.

[Top](#)

## Events

	Name	Description
 	<a href="#">LogTheMessage</a>	Configure this Action to log

messages the way you like to

---

[Top](#)

## ► See Also

Reference

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Log Properties

The [Log](#) type exposes the following members.

## Properties

Name	Description
 <b>S</b> <a href="#">AddTimestamp</a>	If True, log messages will be prefixed with a timestamp

[Top](#)

## See Also

[Reference](#)

[Log Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogAddTimestamp Property

If True, log messages will be prefixed with a timestamp

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public static bool AddTimestamp { get; set; }
```

Property Value

Type: [Boolean](#)

## ► See Also

Reference

[Log Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Log Methods

## ↳ Methods

	Name	Description
≡  	<a href="#">e(Exception)</a>	Log an Exception
≡  	<a href="#">e(String, Object)</a>	Log a formatted exception message. This method uses string.Format to format the message.
≡  	<a href="#">i(String, Object)</a>	Log a formatted informational message. This method uses string.Format to format the message.
≡  	<a href="#">i(String, String, Int32)</a>	Log a formatted informational message. This method uses string.Format to format the message.
≡  	<a href="#">v(String, Object)</a>	Log a formatted verbose message. This method uses string.Format to format the message.
≡  	<a href="#">v(String, String, Int32)</a>	Log a formatted verbose message. This method uses string.Format to format the message.
≡  	<a href="#">w(String, Object)</a>	Log a formatted warning message. This method uses string.Format to format the message.



w(String,  
String,  
Int32)

Log a formatted warning message.  
This method uses string.Format to  
format the message.

---

[Top](#)

## ▲ See Also

[Reference](#)

[Log Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Log Method

## ▪ Overload List

	Name	Description
 	<a href="#">e(Exception)</a>	Log an Exception
 	<a href="#">e(String, Object)</a>	Log a formatted exception message. This method uses string.Format to format the message.

[Top](#)

## ▪ See Also

[Reference](#)

[Log Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Log Method (Exception)

Log an Exception

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C#  
public static void e(  
    Exception e  
)
```

[Copy](#)

### Parameters

e

Type: [SystemException](#)

The exception to log. This will be boxed with `ToString()`.

## ► See Also

Reference

[Log Class](#)

[e Overload](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Loge Method (String, Object)

Log a formatted exception message. This method uses string.Format to format the message.

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C#  
public static void e(  
    string format,  
    params Object[] args  
)
```

[Copy](#)

### Parameters

*format*

Type: [System.String](#)

Format of the message

*args*

Type: [System.Object](#)

Parameters to be passed during message formatting

## ► See Also

Reference

[Log Class](#)

[e Overload](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# Logi Method

## ▪ Overload List

	Name	Description
 	<code>i(String, Object)</code>	Log a formatted informational message. This method uses string.Format to format the message.
 	<code>i(String, String, Int32)</code>	Log a formatted informational message. This method uses string.Format to format the message.

[Top](#)

## ▪ See Also

Reference

[Log Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Logi Method (String, Object)

Log a formatted informational message. This method uses string.Format to format the message.

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C#  
public static void i(  
    string format,  
    params Object[] args  
)
```

[Copy](#)

### Parameters

*format*

Type: [System.String](#)

Format of the message

*args*

Type: [System.Object](#)

Parameters to be passed during message formatting

## ► See Also

Reference

[Log Class](#)

[i Overload](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# Logi Method (String, String, Int32)

Log a formatted informational message. This method uses string.Format to format the message.

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public static void i(
    string message,
    string callerName = "",
    int callerLineNumber = 0
)
```

### Parameters

*message*

Type: [SystemString](#)

Format of the message

*callerName (Optional)*

Type: [SystemString](#)

The name of the caller

*callerLineNumber (Optional)*

Type: [SystemInt32](#)

The line number of the caller

## ► See Also

Reference

Log Class  
i Overload  
W.Logging Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Logv Method

## ▪ Overload List

Name	Description
  <b>v(String, Object)</b>	Log a formatted verbose message. This method uses string.Format to format the message.
  <b>v(String, String, Int32)</b>	Log a formatted verbose message. This method uses string.Format to format the message.

[Top](#)

## ▪ See Also

[Reference](#)

[Log Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Logv Method (String, Object)

Log a formatted verbose message. This method uses `string.Format` to format the message.

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public static void v(  
    string format,  
    params Object[] args  
)
```

### Parameters

*format*

Type: [System.String](#)

Format of the message

*args*

Type: [System.Object](#)

Parameters to be passed during message formatting

## ► See Also

Reference

[Log Class](#)

[v Overload](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# Logv Method (String, String, Int32)

Log a formatted verbose message. This method uses string.Format to format the message.

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public static void v(
    string message,
    string callerName = "",
    int callerLineNumber = 0
)
```

## Parameters

*message*

Type: [System.String](#)

Format of the message

*callerName (Optional)*

Type: [System.String](#)

The name of the caller

*callerLineNumber (Optional)*

Type: [System.Int32](#)

The line number of the caller

## ► See Also

Reference

Log Class  
v Overload  
W.Logging Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Logw Method

## ▪ Overload List

	Name	Description
≡  	w(String, Object)	Log a formatted warning message. This method uses string.Format to format the message.
≡  	w(String, String, Int32)	Log a formatted warning message. This method uses string.Format to format the message.

[Top](#)

## ▪ See Also

[Reference](#)

[Log Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Logw Method (String, Object)

Log a formatted warning message. This method uses string.Format to format the message.

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C#  
public static void w(  
    string format,  
    params Object[] args  
)
```

[Copy](#)

### Parameters

*format*

Type: [System.String](#)

Format of the message

*args*

Type: [System.Object](#)

Parameters to be passed during message formatting

## ► See Also

Reference

[Log Class](#)

[w Overload](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# Logw Method (String, String, Int32)

Log a formatted warning message. This method uses string.Format to format the message.

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public static void w(
    string message,
    string callerName = "",
    int callerLineNumber = 0
)
```

### Parameters

*message*

Type: [System.String](#)

Format of the message

*callerName (Optional)*

Type: [System.String](#)

The name of the caller

*callerLineNumber (Optional)*

Type: [System.Int32](#)

The line number of the caller

## ► See Also

Reference

Log Class  
w Overload  
W.Logging Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Log Events

The [Log](#) type exposes the following members.

## Events

	Name	Description
 	<a href="#">LogTheMessage</a>	Configure this Action to log messages the way you like to

[Top](#)

## See Also

[Reference](#)

[Log Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogLogTheMessage Event

Configure this Action to log messages the way you like to

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public static event Action<string> LogTheMessage
```

Value

Type: [SystemActionString](#)

## ► Remarks

This method needs to be replaced with something useful to you.

## ► See Also

Reference

[Log Class](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# LogLogMessageCategory Enumeration

The log message type

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public enum LogMessageCategory
```

## ► Members

Member name	Value	Description
Verbose	0	Denotes verbose message
Information	1	Denotes a informational message
Warning	2	Denotes a warning message
Error	3	Denotes an error message

## ► See Also

Reference

## [W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogMessageHistory Class

Maintains a history of Log information

## ► Inheritance Hierarchy

SystemObject W.LoggingLogPropertyHost  
W.LoggingLogMessageHistory

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public class MessageHistory : LogPropertyHost
```

The [LogMessageHistory](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">LogMessageHistory</a>	Constructs a new LogMessageHistory

[Top](#)

## ► Properties

	Name	Description
📋	<a href="#">Enabled</a>	If True, log messages will be

added to the history. If False, no history is maintained.



#### [MaximumNumberOfMessages](#)

The maximum number of historical messages to maintain. When the maximum is reached, the oldest messages are removed as needed.



#### [Messages](#)

The history of log messages

[Top](#)

## Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.

		(Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">LogPropertyHost</a> .)
	<a href="#">SetValue</a>	(Inherited from <a href="#">LogPropertyHost</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Events

Name	Description
 <a href="#">PropertyChanged</a>	Raised when the value of a property has changed (Inherited from <a href="#">LogPropertyHost</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ◀ See Also

Reference

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogMessageHistory Constructor

Constructs a new LogMessageHistory

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public MessageHistory(  
    bool newestFirst = true  
)
```

### Parameters

*newestFirst* **(Optional)**

Type: [SystemBoolean](#)

If True, log messages are inserted at the start of the collection rather than appended to the end

## ► See Also

[Reference](#)

[LogMessageHistory Class](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# MessageHistory Properties

The [LogMessageHistory](#) type exposes the following members.

## Properties

Name	Description
 <a href="#">Enabled</a>	If True, log messages will be added to the history. If False, no history is maintained.
 <a href="#">MaximumNumberOfMessages</a>	The maximum number of historical messages to maintain. When the maximum is reached, the oldest messages are removed as needed.
 <a href="#">Messages</a>	The history of log messages

[Top](#)

## See Also

[Reference](#)

## LogMessageHistory Class W.Logging Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogMessageHistoryEnabled Property

If True, log messages will be added to the history. If False, no history is maintained.

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

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

Property Value

Type: [Boolean](#)

## ► See Also

Reference

[LogMessageHistory Class](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# LogMessageHistoryMaximumNumberOfMessages Property

The maximum number of historical messages to maintain. When the maximum is reached, the oldest messages are removed as needed.

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C#  
public int MaximumNumberOfMessages { get; }
```

Property Value

Type: [Int32](#)

## ► See Also

Reference

[LogMessageHistory Class](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# LogMessageHistoryMessages Property

The history of log messages

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public ObservableCollection<LogMessageHistoryLog>
```

Property Value

Type: [ObservableCollectionLogMessageHistoryLogMessage](#)

## ► See Also

Reference

[LogMessageHistory Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MessageHistory Methods

The [LogMessageHistory](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow

---

		copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">LogPropertyHost</a> .)
💡	<a href="#">SetValue</a>	(Inherited from <a href="#">LogPropertyHost</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## ► Extension Methods

Name	Description
💡 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members

(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)



## [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets it's `IsDirty` flag to `false`

(Defined by [PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor unlock

(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ► See Also

### Reference

[LogMessageHistory Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MessageHistory Events

The [LogMessageHistory](#) type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">PropertyChanged</a>	Raised when the value of a property has changed (Inherited from <a href="#">LogPropertyHost</a> .)

[Top](#)

## ▪ See Also

[Reference](#)

[LogMessageHistory Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogMessageHistoryLogMessage Class

An individual log message

## ▪ Inheritance Hierarchy

```
SystemObject W.LoggingLogPropertyHost  
W.LoggingLogMessageHistoryLogMessage
```

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ▪ Syntax

C#

[Copy](#)

```
public class LogMessage : LogPropertyHost
```

The [LogMessageHistoryLogMessage](#) type exposes the following members.

## ▪ Constructors

	Name	Description
≡	<a href="#">LogMessageHistoryLogMessage</a>	Constructs a new LogMessage

[Top](#)

## ▪ Properties

	Name	Description
	<a href="#">Message</a>	The logged message

[Top](#)

## Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current

		Object. (Inherited from Object.)
💡	RaiseOnPropertyChanged	Raises the PropertyChanged event (Inherited from LogPropertyHost.)
💡	SetValue	(Inherited from LogPropertyHost.)
💡	ToString	Returns a string that represents the current object. (Inherited from Object.)

[Top](#)

## Events

	Name	Description
⚡	PropertyChanged	Raised when the value of a property has changed (Inherited from LogPropertyHost.)

[Top](#)

## Extension Methods

	Name	Description
💡	AsTType	Use Generic syntax for the as operator.

(Defined by  
[AsExtensions](#).)

---

InitializeProperties

Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has [IOwnedProperty](#) members

(Defined by  
[PropertyHostExtensions](#).)

---

InLock(Action)

Overloaded.  
Performs the action in a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

InLockTType(FuncTType)

Overloaded.  
Performs the function in a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

InLockAsync(Action)

Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

InLockAsyncTType(FuncTType)

Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by

[MonitorExtensions.\)](#)

 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

[Top](#)

## ▲ See Also

[Reference](#)

[W.Logging Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogMessageHistoryLogMessage Constructor

Constructs a new LogMessage

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

```
public LogMessage()
```

[Copy](#)

## ► See Also

Reference

[LogMessageHistoryLogMessage Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogMessage Properties

The [LogMessageHistoryLogMessage](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">Message</a>	The logged message

[Top](#)

## See Also

### Reference

[LogMessageHistoryLogMessage Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogMessageHistoryLogMessageMe Property

The logged message

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public string Message { get; }
```

Property Value

Type: [String](#)

## ► See Also

Reference

[LogMessageHistoryLogMessage Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogMessage Methods

The [LogMessageHistory](#)[LogMessage](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)

💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event (Inherited from <a href="#">LogPropertyHost</a> .)
💡	<a href="#">SetValue</a>	(Inherited from <a href="#">LogPropertyHost</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ► Extension Methods

Name	Description
💡 ⚡ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has

IOwnedProperty  
members  
(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IP <i>roperty</i> (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock

(Defined by  
[MonitorExtensions](#).)

---



### [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets it's `IsDirty` flag to false

(Defined by  
[PropertyHostExtensions](#).)

---



### [Unlock](#)

Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

### Reference

[LogMessageHistory](#)[LogMessage Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogMessage Events

The [LogMessageHistoryLogMessage](#) type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">PropertyChanged</a>	Raised when the value of a property has changed (Inherited from <a href="#">LogPropertyHost</a> .)

[Top](#)

## ▪ See Also

Reference

[LogMessageHistoryLogMessage Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogPropertyHost Class

Provides

## ► Inheritance Hierarchy

```
SystemObject W.LoggingLogPropertyHost
  W.LoggingLogMessageHistory
    W.LoggingLogMessageHistoryLogMessage
```

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
public class PropertyHost : INotifyPropertyChanged
```

The [LogPropertyHost](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">LogPropertyHost</a>	Initializes a new instance of the LogPropertyHost class

[Top](#)

## ► Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether

		the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">RaiseOnPropertyChanged</a>	Raises the <a href="#">PropertyChanged</a> event
💡	<a href="#">SetValue</a>	
💡	<a href="#">ToString</a>	Returns a string that

represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## Events

Name	Description
 <a href="#">PropertyChanged</a>	Raised when the value of a property has changed

[Top](#)

## Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a

Monitor lock  
(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by

[PropertyHostExtensions.\)](#)

---



## [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions.](#))

---

[Top](#)

## ► See Also

Reference

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogPropertyHost Constructor

Initializes a new instance of the [LogPropertyHost](#) class

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

Copy

```
public PropertyHost()
```

## ► See Also

[Reference](#)

[LogPropertyHost Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHost Methods

The [LogPropertyHost](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow

copy of the current [Object](#).  
(Inherited from [Object](#).)



### RaiseOnPropertyChanged

Raises the [PropertyChanged](#) event



### SetValue



### ToString

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## Unlock

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

Reference

[LogPropertyHost Class](#)  
[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogPropertyHostRaiseOnPropertyChanged Method

Raises the PropertyChanged event

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
protected void RaiseOnPropertyChanged(
    Object sender,
    string propertyName
)
```

### Parameters

*sender*

Type: [SystemObject](#)

The sender is the owner of the property

*propertyName*

Type: [SystemString](#)

The name of the caller (the property which changed)

## ► See Also

[Reference](#)

[LogPropertyHost Class](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# LogPropertyHostSetValue Method

[Missing <summary> documentation for  
"M:W.Logging.Log.PropertyHost.SetValue(System.Object,System.Action,System.String)"]

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

```
C# Copy
protected virtual void SetValue(
    Object owner,
    Action assignValue,
    string callerMemberName = "")
```

## Parameters

*owner*

Type: [SystemObject](#)

[Missing <param name="owner"/> documentation for  
"M:W.Logging.Log.PropertyHost.SetValue(System.Object,System.Action,System.String)"]

*assignValue*

Type: [SystemAction](#)

[Missing <param name="assignValue"/> documentation for  
"M:W.Logging.Log.PropertyHost.SetValue(System.Object,System.Action,System.String)"]

*callerMemberName (Optional)*

Type: [SystemString](#)

[Missing <param name="callerMemberName"/> documentation for  
"M:W.Logging.Log.PropertyHost.SetValue(System.Object,System.Action,System.String)"]

## ▲ See Also

Reference

[LogPropertyHost Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# PropertyHost Events

The [LogPropertyHost](#) type exposes the following members.

## ▪ Events

Name	Description
 <a href="#">PropertyChanged</a>	Raised when the value of a property has changed

[Top](#)

## ▪ See Also

Reference

[LogPropertyHost Class](#)

[W.Logging Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LogPropertyHostPropertyChanged Event

Raised when the value of a property has changed

**Namespace:** [W.Logging](#)

**Assembly:** Tungsten.Logging (in Tungsten.Logging.dll) Version: 2.0.0

## ► Syntax

C#

[Copy](#)

```
public event PropertyChangedEventHandler Property
```

Value

Type: [System.ComponentModel.PropertyChangedEventHandler](#)

Implements

[INotifyPropertyChanged.PropertyChanged](#)

## ► See Also

Reference

[LogPropertyHost Class](#)

[W.Logging Namespace](#)

## Tungsten

W<sup>74</sup>

# W.Net Namespace

[Missing <summary> documentation for "N:W.Net"]

## ► Classes

Class	Description
 <a href="#">ExtensionMethods</a>	Extension methods for W.Net
 <a href="#">Tcp</a>	
 <a href="#">TcpGeneric</a>	
 <a href="#">TcpGenericSecureTcpClientTMessage</a>	
 <a href="#">TcpGenericSecureTcpHostTMessage</a>	
 <a href="#">TcpGenericTcpClientTMessage</a>	
 <a href="#">TcpGenericTcpHostTMessage</a>	
 <a href="#">TcpSecureTcpClient</a>	
 <a href="#">TcpSecureTcpHost</a>	
 <a href="#">TcpSecureTcpLogger</a>	Sends log messages to a remote server via secure Tcp
 <a href="#">TcpTcpClient</a>	
 <a href="#">TcpTcpHost</a>	

---

	<a href="#">TcpTcpLogger</a>	Sends log messages to a remote server via Tcp
	<a href="#">Udp</a>	Provides simple UDP reading and writing
	<a href="#">UdpGeneric</a>	Contains the generic implementation of W.Net.UdpPeer
	<a href="#">UdpGenericUdpPeerTType</a>	A generic Udp peer
	<a href="#">UdpUdpLogger</a>	Sends log messages to a remote server via Udp
	<a href="#">UdpUdpPeer</a>	A Udp peer
	<a href="#">WakeOnLAN</a>	Methods to broadcast a magic packet to wake up a machine with the given MAC address

---

## ► Interfaces

Interface	Description
-----------	-------------

---

~o [TcpIClient](#)

---

~o [TcpIInitialize](#)

---

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ExtensionMethods Class

Extension methods for W.Net

## ► Inheritance Hierarchy

[SystemObject](#) [W.NetExtensionMethods](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static class ExtensionMethods
```

The [ExtensionMethods](#) type exposes the following members.

## ► Methods

### Name



[GetBroadcastAddress](#)



[RequestAsync\(TcpSecureTcpClient, Byte, Int32\)](#)

- 
- ≡   RequestAsync(TcpTcpClient, Byte, Int32)
  
  - ≡   RequestAsync(UdpUdpPeer, Byte, IPEndPoint, Int32)

---

  RequestAsyncTMessage(TcpGenericSecureTcpClientTMessage, TMessage, Int32)

---

  RequestAsyncTMessage(TcpGenericTcpClientTMessage, TMessage, Int32)

---

  RequestAsyncTMessage(UdpGenericUdpPeerTMessage, TMessage, IPEndPoint, Int32)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ExtensionMethods Methods

The [ExtensionMethods](#) type exposes the following members.

## ▪ Methods

Name
≡vip S GetBroadcastAddress
≡vip S RequestAsync(TcpSecureTcpClient, Byte, Int32)
≡vip S RequestAsync(TcpTcpClient, Byte, Int32)

---

≡  RequestAsync(UdpUdpPeer, Byte, IPPEndPoint, Int32)

---

≡  RequestAsyncTMessage(TcpGenericSecureTcpClientTMessage, Int32)



---

`RequestAsyncTMessage(TcpGenericTcpClientTMessage,  
TMessage, Int32)`



---

`RequestAsyncTMessage(UdpGenericUdpPeerTMessage,  
TMessage, IPEndPoint, Int32)`

---

[Top](#)

◀ **See Also**

Reference

[ExtensionMethods Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ExtensionMethodsGetBroadcastAddress Method

Determines the broadcast address from an ip address and subnet mask

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static IPAddress GetBroadcastAddress(
    this IPAddress address,
    IPAddress subnetMask
)
```

### Parameters

*address*

Type: [System.NetIPAddress](#)

The IP address

*subnetMask*

Type: [System.NetIPAddress](#)

The subnet mask

### Return Value

Type: [IPAddress](#)

The broadcast IP address associated with the given IP address and subnet mask

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IPAddress](#). When you use instance

method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► Remarks

Taken from: <https://blogs.msdn.microsoft.com/knom/2008/12/31/ip-address-calculations-with-c-subnetmasks-networks/>

## ► See Also

Reference

[ExtensionMethods Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ExtensionMethodsRequestAsync Method

## ▪ Overload List

Name
≡  RequestAsyncTMessage(TcpGenericSecureTcpClientTMessage, TMessage, Int32)
≡  RequestAsyncTMessage(TcpGenericTcpClientTMessage, TMessage, Int32)

- 
- ≡   RequestAsync(TcpSecureTcpClient, Byte, Int32)
  - ≡   RequestAsync(TcpTcpClient, Byte, Int32)
  - ≡   RequestAsyncTMessage(UdpGenericUdpPeerTMessage, TMessage, IPPEndPoint, Int32)



---

[RequestAsync\(UdpUdpPeer, Byte, IPEndPoint, Int32\)](#)

---

[Top](#)

## ▲ See Also

[Reference](#)

[ExtensionMethods Class](#)

[W.Net Namespace](#)

---



## Tungsten

W<sup>74</sup>

# ExtensionMethodsRequestAsync<TMessage>(TcpGenericSecureTcpClient<TMessage>, TMessage, Int32)

Sends a request to the server and waits for a response. Can be used along with, or in lieu of, the regular method of calling Write and handling the BytesReceived event.

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static Task<TMessage> RequestAsync<TMessage>
    (this TcpGenericSecureTcpClient<TMessage>
        TMessage request,
        int msTimeout
    )
```

## Parameters

*client*

Type: [W.NetTcpGenericSecureTcpClient<TMessage>](#)

The W.Net.Tcp.Generic.SecureTcpClient<TMessage> instance

*request*

Type: [TMessage](#)

The request to be made to the server

*msTimeout*

Type: [SystemInt32](#)

The time to wait for a response

## Type Parameters

### *TMessage*

The message Type

## Return Value

Type: [Task<TMessage>](#)

default(*TMessage*) if a timeout occurs, otherwise the server's response

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [TcpGenericSecureTcpClient<TMessage>](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[ExtensionMethods Class](#)

[RequestAsync Overload](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# ExtensionMethodsRequestAsync $TMessage$ Method (TcpGenericTcpClient $TMessage$ , Int32)

Sends a request to the server and waits for a response. Can be used along with, or in lieu of, the regular method of calling Write and handling the BytesReceived event.

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static Task<TMessage> RequestAsync<TMessage>
    this TcpGenericTcpClient<TMessage> client,
    TMessage request,
    int msTimeout
)
```

## Parameters

*client*

Type: [W.NetTcpGenericTcpClient \$TMessage\$](#)

The W.Net.Tcp.Generic.TcpClient<TMessage> instance  
*request*

Type: [TMessage](#)

The request to be made to the server

*msTimeout*

Type: [SystemInt32](#)

The time to wait for a response

## Type Parameters

### *TMessage*

The message Type

## Return Value

Type: [Task<TMessage>](#)

default(*TMessage*) if a timeout occurs, otherwise the server's response

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [TcpGenericTcpClient<TMessage>](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[ExtensionMethods Class](#)

[RequestAsync Overload](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# ExtensionMethodsRequestAsync Method (TcpSecureTcpClient, Byte, Int32)

Sends a request to the server and waits for a response. Can be used along with, or in lieu of, the regular method of calling Write and handling the BytesReceived event.

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static Task<byte[]> RequestAsync(  
    this TcpSecureTcpClient client,  
    byte[] request,  
    int msTimeout  
)
```

## Parameters

*client*

Type: [W.NetTcpSecureTcpClient](#)

The W.Net.Tcp.SecureTcpClient instance

*request*

Type: [SystemByte](#)

The request to be made to the server

*msTimeout*

Type: [SystemInt32](#)

The time to wait for a response

## Return Value

Type: [TaskByte](#)

Null if a timeout occurs, otherwise the server's response

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [TcpSecureTcpClient](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[ExtensionMethods Class](#)

[RequestAsync Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ExtensionMethodsRequestAsync Method (TcpTcpClient, Byte, Int32)

Sends a request to the server and waits for a response. Can be used along with, or in lieu of, the regular method of calling Write and handling the BytesReceived event.

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static Task<byte[]> RequestAsync(  
    this TcpTcpClient client,  
    byte[] request,  
    int msTimeout  
)
```

## Parameters

*client*

Type: [W.NetTcpTcpClient](#)

The W.Net.Tcp.TcpClient instance

*request*

Type: [SystemByte](#)

The request to be made to the server

*msTimeout*

Type: [SystemInt32](#)

The time to wait for a response

## Return Value

Type: [TaskByte](#)

Null if a timeout occurs, otherwise the server's response

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [TcpTcpClient](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[ExtensionMethods Class](#)

[RequestAsync Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ExtensionMethodsRequestAsync<sub>TMessage</sub> Method (UdpGenericUdpPeer<TMessage>, IPEndPoint, Int32)

Sends a request to a generic Udp peer and waits for a response. Can be used along with, or in lieu of, the regular method of calling Write and handling the BytesReceived event.

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static Task<TMessage> RequestAsync<TMessage>(
    this UdpGenericUdpPeer<TMessage> peer,
    TMessage request,
    IPEndPoint remoteEndPoint,
    int msTimeout
)
```

## Parameters

*peer*

Type: [W.NetUdpGenericUdpPeer<TMessage>](#)

The W.Net.Udp.Generic.UdpPeer<TMessage> instance

*request*

Type: [TMessage](#)

The request to be made to the server

*remoteEndPoint*

Type: [System.NetIPEndPoint](#)

The IPEndPoint of the remote Udp peer

*msTimeout*

Type: [SystemInt32](#)

The time to wait for a response

## Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for  
"M:W.Net.ExtensionMethods.RequestAsync`1(W.Net.Udp.Generic.UdpPeer{`0},`

## Return Value

Type: [Task<TMessage>](#)

default(*TMessage*) if a timeout occurs, otherwise the server's response

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [UdpGenericUdpPeer<TMessage>](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[ExtensionMethods Class](#)

[RequestAsync Overload](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# ExtensionMethodsRequestAsync Method (UdpUdpPeer, Byte, IPEndPoint, Int32)

Sends a request to a Udp peer and waits for a response. Can be used along with, or in lieu of, the regular method of calling Write and handling the BytesReceived event.

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static Task<byte[]> RequestAsync(  
    this UdpUdpPeer peer,  
    byte[] request,  
    IPEndPoint remoteEndPoint,  
    int msTimeout  
)
```

## Parameters

*peer*

Type: [W.NetUdpUdpPeer](#)

The W.Net.Udp.UdpPeer instance

*request*

Type: [SystemByte](#)

The request to be made to the server

*remoteEndPoint*

Type: [System.NetIPEndPoint](#)

The IPEndPoint of the remote Udp peer

*msTimeout*

Type: [SystemInt32](#)

The time to wait for a response

Return Value

Type: [TaskByte](#)

Null if a timeout occurs, otherwise the server's response

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [UdpUdpPeer](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ▲ See Also

Reference

[ExtensionMethods Class](#)

[RequestAsync Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Tcp Class

[Missing <summary> documentation for "T:W.Net.Tcp"]

## ► Inheritance Hierarchy

SystemObject W.NetTcp

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static class Tcp
```

## ► See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGeneric Class

[Missing <summary> documentation for "T:W.Net.Tcp.Generic"]

## ► Inheritance Hierarchy

SystemObject W.NetTcpGeneric

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public static class Generic
```

## ► See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericSecureTcpClientTMessage Class

[Missing <summary> documentation for "T:W.Net.Tcp.Generic.SecureTcpClient`1"]

## ► Inheritance Hierarchy

```
SystemObject W.NetTcpTcpClient
  W.NetTcpSecureTcpClient
    W.NetTcpGenericSecureTcpClientTMessage
```

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#
public class SecureTcpClient<TMessage> : TcpSecure
  ↑
  ↓
```

### Type Parameters

#### *TMessage*

[Missing <typeparam name="TMessage"/> documentation for "T:W.Net.Tcp.Generic.SecureTcpClient`1"]

The [TcpGenericSecureTcpClientTMessage](#) type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">Constructors</a>	

<a href="#">TcpGenericSecureTcpClientTMessage</a>	Initializes a new instance of the <a href="#">TcpGenericSecureTcpClient</a> class.
---	--

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Encryption</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
	<a href="#">IsSecure</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
	<a href="#">IsServerSide</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Socket</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Connect</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
	<a href="#">Disconnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Disconnect(Boolean, Boolean)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)

💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnConnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
💡	<a href="#">OnInitialize</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
💡	<a href="#">OnReceived</a>	(Overrides <a href="#">TcpSecureTcpClientOnReceived</a> (Byt
💡	<a href="#">OnSend</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Write(TMessage)</a>	
💡	<a href="#">Write(Byte)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## Events

Name	Description
------	-------------

	<a href="#">BytesReceived</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Connected</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Disconnected</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">MessageReceived</a>	
	<a href="#">SecureFailed</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)

[Top](#)

## Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" sets the member's Owner property to "owner" This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

•	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property type IProperty (Defined by <a href="#">PropertyHostExtensions</a> )
•	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> )
•	<a href="#">RequestAsync(Byte, Int32)</a>	Overloaded. Sends a request to the host

server and waits for response. Can be used along with, or in lieu of the regular method `Write` for calling `Write` and handling the `BytesReceived` event.  
(Defined by [ExtensionMethods](#).)



### [RequestAsync\(Byte, Int32\)](#)

Overloaded.  
Sends a request to the server and waits for response. Can be used along with, or in lieu of the regular method `Write` for calling `Write` and handling the `BytesReceived` event.  
(Defined by [ExtensionMethods](#).)



### [RequestAsync\(TMessage, Int32\)](#)

Overloaded.  
Sends a request to the server and waits for response. Can be used along with, or in lieu of the regular method `Write` for calling `Write` and handling the `BytesReceived` event.  
(Defined by [ExtensionMethods](#).)



### [Unlock](#)

Performs a Monitor unlock.  
(Defined by [Object](#))

[MonitorExtensions.\)](#)

---

[Top](#)

## ► See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericSecureTcpClientTMessage Constructor

Initializes a new instance of the [TcpGenericSecureTcpClientTMessage](#) class

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public SecureTcpClient(  
    int keySize  
)
```

### Parameters

*keySize*

Type: [System.Int32](#)

[Missing <param name="keySize"/> documentation for  
"M:W.Net.Tcp.Generic.SecureTcpClient`1.#ctor(System.Int32)"]

## ► See Also

[Reference](#)

[TcpGenericSecureTcpClientTMessage Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# SecureTcpClientTMessage Properties

The [TcpGenericSecureTcpClientTMessage](#) generic type exposes the following members.

## Properties

	Name	Description
	<a href="#">Encryption</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
	<a href="#">IsSecure</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
	<a href="#">IsServerSide</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Socket</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## See Also

### Reference

[TcpGenericSecureTcpClientTMessage Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# SecureTcpClientTMessage Methods

The [TcpGenericSecureTcpClientTMessage](#) generic type exposes the following members.

## Methods

Name	Description
<a href="#">Connect</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
<a href="#">Disconnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
<a href="#">Disconnect(Boolean, Boolean)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance (Inherited from <a href="#">Object</a> .)

💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnConnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
💡	<a href="#">OnInitialize</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
💡	<a href="#">OnReceived</a>	(Overrides <a href="#">TcpSecureTcpClient.OnReceived(Byte)</a> )
💡	<a href="#">OnSend</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Write(TMessage)</a>	
≡	<a href="#">Write(Byte)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## ◀ Extension Methods

Name	Description
💡 ⚡ <a href="#">AsTType</a>	Use Generic syntax the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" sets the member's Owner property to "owner". This method should be called in the constructor of any class which has

IOwnedProperty  
members  
(Defined by  
[PropertyHostExtens](#))

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously per- the action in a Monit lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously per- the action in a Monit lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty val- each field and prope- type IProperty (Defined by <a href="#">PropertyHostExtens</a> )
 <a href="#">Lock</a>	Performs a Monitor l

(Defined by  
[MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and property of type [IPro](#) and sets its [IsDirty](#) flag to false

(Defined by  
[PropertyHostExtensi](#)



### [RequestAsync\(Byte, Int32\)](#)

Overloaded.  
Sends a request to the server and waits for response. Can be used along with, or in lieu of the regular method [Write](#) for calling [Write](#) and handling the [BytesReceived](#) event.

(Defined by  
[ExtensionMethods](#).)



### [RequestAsync\(Byte, Int32\)](#)

Overloaded.  
Sends a request to the server and waits for response. Can be used along with, or in lieu of the regular method [Write](#) for calling [Write](#) and handling the [BytesReceived](#) event.

(Defined by  
[ExtensionMethods](#).)



### [RequestAsyncTMessage\(TMessage, Int32\)](#)

Overloaded.  
Sends a request to the server and waits for response. Can be used

along with, or in lieu  
the regular method c  
calling Write and  
handling the  
BytesReceived even  
(Defined by  
[ExtensionMethods.](#))



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions.](#))

---

[Top](#)

## ► See Also

Reference

[TcpGenericSecureTcpClientTMessage Class](#)  
[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericSecureTcpClientTMessage Method

[Missing <summary> documentation for "M:W.Net.Tcp.Generic.SecureTcpClient`1.OnReceived(System.Byte[]@)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
protected override void OnReceived(
    ref byte[] bytes
)
```

### Parameters

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for "M:W.Net.Tcp.Generic.SecureTcpClient`1.OnReceived(System.Byte[]@)"]

## ► See Also

[Reference](#)

[TcpGenericSecureTcpClientTMessage Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# SecureTcpClientTMessageWrite Method

## ► Overload List

Name	Description
 <a href="#">Write(TMessage)</a>	
 <a href="#">Write(Byte)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## ► See Also

Reference

[TcpGenericSecureTcpClientTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericSecureTcpClientTMessage Method (TMessage)

[Missing <summary> documentation for "M:W.Net.Tcp.Generic.SecureTcpClient`1.Write(`0)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public void Write(  
    TMessage message  
)
```

[Copy](#)

### Parameters

*message*

Type: [TMessage](#)

[Missing <param name="message"/> documentation for "M:W.Net.Tcp.Generic.SecureTcpClient`1.Write(`0)"]

## ► See Also

[Reference](#)

[TcpGenericSecureTcpClientTMessage Class](#)

[Write Overload](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# SecureTcpClientTMessage Events

The [TcpGenericSecureTcpClientTMessage](#) generic type exposes the following members.

## Events

	Name	Description
⚡	<a href="#">BytesReceived</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
⚡	<a href="#">Connected</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
⚡	<a href="#">Disconnected</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
⚡	<a href="#">MessageReceived</a>	
⚡	<a href="#">SecureFailed</a>	(Inherited from <a href="#">TcpSecureTcpClient</a> .)

[Top](#)

## See Also

### Reference

[TcpGenericSecureTcpClientTMessage Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpGenericSecureTcpClientTMessageEvent

[Missing <summary> documentation for "E:W.Net.Tcp.Generic.SecureTcpClient`1.MessageReceived"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public event Action<TcpGenericSecureTcpClient<TMessage>, TMessage> MessageReceived;
```

### Value

Type: [SystemAction<TcpGenericSecureTcpClient<TMessage>, TMessage>](#)

## ► See Also

### Reference

[TcpGenericSecureTcpClientTMessage Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpGenericSecureTcpHostTMessage Class

[Missing <summary> documentation for "T:W.Net.Tcp.Generic.SecureTcpHost`1"]

## ► Inheritance Hierarchy

SystemObject W.NetTcpTcpHost  
W.NetTcpGenericSecureTcpHostTMessage

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public class SecureTcpHost<TMessage> : TcpTcpHost
```

### Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for  
"T:W.Net.Tcp.Generic.SecureTcpHost`1"]

The [TcpGenericSecureTcpHostTMessage](#) type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">TcpGenericSecureTcpHostTMessage</a>	Initializes a new inst

[Top](#)

## Properties

	Name	Description
	<a href="#">IsListening</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">OnCreateServer</a>	(Inherited from <a href="#">TcpTcpHost</a> .)

[Top](#)

## Methods

	Name	Description
	<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">Listen(IPEndPoint)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)

---

	<a href="#">Listen(IPEndPoint, Int32)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">OnBytesReceived</a>	(Overrides <a href="#">TcpTcpHost.OnBytesReceived(TcpIClientByte)</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## Events

Name	Description
	<a href="#">BytesReceived</a> (Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">MessageReceived</a>

---

[Top](#)

## Extension Methods

Name	Description
 	<a href="#">AsTType</a> Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a> Scans the fields and properties of "owner" and

sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members  
(Defined by [PropertyHostExtensions](#).)

---

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty

(Defined by  
[PropertyHostExtensions](#).)



### [Lock](#)

Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and  
property of type `IProperty`  
and sets it's `IsDirty` flag to  
false  
(Defined by  
[PropertyHostExtensions](#).)



### [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

[Top](#)

## ► See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericSecureTcpHostTMessage Constructor

Initializes a new instance of the [TcpGenericSecureTcpHostTMessage](#) class

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public SecureTcpHost(  
    int keySize  
)
```

[Copy](#)

### Parameters

*keySize*

Type: [System.Int32](#)

[Missing <param name="keySize"/> documentation for  
"M:W.Net.Tcp.Generic.SecureTcpHost`1.#ctor(System.Int32)"]

## ► See Also

[Reference](#)

[TcpGenericSecureTcpHostTMessage Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# SecureTcpHostTMessage Properties

The [TcpGenericSecureTcpHostTMessage](#) generic type exposes the following members.

## Properties

	Name	Description
	<a href="#">IsListening</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">OnCreateServer</a>	(Inherited from <a href="#">TcpTcpHost</a> .)

[Top](#)

## See Also

### Reference

[TcpGenericSecureTcpHostTMessage Class](#)  
[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SecureTcpHostTMessage Methods

The [TcpGenericSecureTcpHostTMessage](#) generic type exposes the following members.

## Methods

Name	Description
<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
<a href="#">Listen(IPEndPoint)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
<a href="#">Listen(IPEndPoint, Int32)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current

[Object](#).

(Inherited from [Object](#).)

---

 [OnBytesReceived](#) (Overrides [TcpTCPHost.OnBytesReceived\(TcpIClie](#)  
[Byte\)](#).)

---

 [ToString](#) Returns a string that represents the  
current object.  
(Inherited from [Object](#).)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[TcpGenericSecureTcpHostTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericSecureTcpHostTMessage

## Method

[Missing <summary> documentation for  
"M:W.Net.Tcp.Generic.SecureTcpHost`1.OnBytesReceived(W.Net.Tcp.IClient,System.Byte[

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

### ► Syntax

```
C# Copy
protected override void OnBytesReceived(
    TcpIClient client,
    byte[] bytes
)
```

### Parameters

*client*

Type: [W.NetTcpIClient](#)

[Missing <param name="client"/> documentation for  
"M:W.Net.Tcp.Generic.SecureTcpHost`1.OnBytesReceived(W.Net.Tcp.IClient,System.Byte[

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for  
"M:W.Net.Tcp.Generic.SecureTcpHost`1.OnBytesReceived(W.Net.Tcp.IClient,System.Byte[

### ► See Also

[Reference](#)

[TcpGenericSecureTcpHostTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SecureTcpHostTMessage Events

The [TcpGenericSecureTcpHostTMessage](#) generic type exposes the following members.

## Events

	Name	Description
	<a href="#">BytesReceived</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">MessageReceived</a>	

[Top](#)

## See Also

### Reference

[TcpGenericSecureTcpHostTMessage Class](#)  
[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericSecureTcpHost $TMessage$ Event

[Missing <summary> documentation for "E:W.Net.Tcp.Generic.SecureTcpHost`1.MessageReceived"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public event Action<TcpGenericSecureTcpHost<TMessage>, TMessage> MessageReceived;
```

### Value

Type: [SystemAction<TcpGenericSecureTcpHost<TMessage>, TMessage>](#),  
[TcpGenericSecureTcpClient<TMessage>](#), [TMessage](#)

## ► See Also

### Reference

[TcpGenericSecureTcpHost<TMessage> Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpGenericTcpClientTMessage Class

[Missing <summary> documentation for "T:W.Net.Tcp.Generic.TcpClient`1"]

## ► Inheritance Hierarchy

SystemObject W.NetTcpTcpClient  
W.NetTcpGenericTcpClientTMessage

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public class TcpClient<TMessage> : TcpTcpClient
```

### Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for  
"T:W.Net.Tcp.Generic.TcpClient`1"]

The [TcpGenericTcpClientTMessage](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">TcpGenericTcpClientTMessage</a>	Initializes a new instance of the <a href="#">TcpGenericTcpClientTMessage</a> class.

class

---

[Top](#)

## ► Properties

	Name	Description
	<a href="#">IsServerSide</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Socket</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Connect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Disconnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Disconnect(Boolean, Boolean)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
		

	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
🔑	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnConnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
💡	<a href="#">OnInitialize</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
💡	<a href="#">OnReceived</a>	(Overrides <a href="#">TcpTcpClientOnReceived(Byte)</a> .)
💡	<a href="#">OnSend</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Write(TMessage)</a>	
≡	<a href="#">Write(Byte)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## Events

	Name	Description
⚡	<a href="#">BytesReceived</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
⚡	<a href="#">Connected</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

---

 [Disconnected](#) (Inherited from [TcpClient](#).)

---

 [MessageReceived](#)

---

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" sets the member's Owner property to "owner" This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

⬇️	InLockAsync(Action)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	InLockAsyncTType(Func<T>)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	IsDirty	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> )
⬇️	Lock	Performs a Monitor Lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	MarkAsClean	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> )
⬇️	RequestAsync(Byte, Int32)	Overloaded. Sends a request to the server and waits for response. Can be used along with, or in lieu of the regular method of calling Write and handling the response.

BytesReceived ever  
(Defined by  
[ExtensionMethods](#).)

---



[RequestAsyncTMessage\(TMessage, Int32\)](#)

Overloaded.  
Sends a request to the server and waits for response. Can be used along with, or in lieu of the regular method calling Write and handling the BytesReceived event (Defined by [ExtensionMethods](#).)

---



[Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericTcpClientTMessage Constructor

Initializes a new instance of the [TcpGenericTcpClientTMessage](#) class

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

```
public TcpClient()
```

[Copy](#)

## ► See Also

[Reference](#)

[TcpGenericTcpClientTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpClientTMessage Properties

The [TcpGenericTcpClientTMessage](#) generic type exposes the following members.

## Properties

	Name	Description
	<a href="#">IsServerSide</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Socket</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## See Also

[Reference](#)

[TcpGenericTcpClientTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpClientTMessage Methods

The [TcpGenericTcpClientTMessage](#) generic type exposes the following members.

## ▪ Methods

	Name	Description
≡◆	<a href="#">Connect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
≡◆	<a href="#">Disconnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
💡	<a href="#">Disconnect(Boolean, Boolean)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
≡◆	<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
≡◆	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡◆	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡◆	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance.

(Inherited from [Object](#).)

💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnConnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
💡	<a href="#">OnInitialize</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
💡	<a href="#">OnReceived</a>	(Overrides <a href="#">TcpTcpClientOnReceived(Byte)</a> .)
💡	<a href="#">OnSend</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
≡💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
≡💡	<a href="#">Write(TMessage)</a>	
≡💡	<a href="#">Write(Byte)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## ◀ Extension Methods

Name	Description
💡 ↴ ⏚ <a href="#">AsTType</a>	Use Generic syntax the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" sets the member's Owner property to "owner". This method should be called in t

constructor of any class which has `IOwnedProperty` members  
(Defined by [PropertyHostExtensions](#))

---

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the <code>IsDirty</code> value of each field and property <code>IProperty</code> (Defined by <a href="#">PropertyHostExtensions</a> )



## Lock

Performs a Monitor I  
(Defined by  
[MonitorExtensions](#).)



## MarkAsClean

Scans each field and  
property of type IPro  
and sets its IsDirty f  
false  
(Defined by  
[PropertyHostExtensi](#)



## RequestAsync(Byte, Int32)

Overloaded.  
Sends a request to t  
server and waits for  
response. Can be us  
along with, or in lieu  
the regular method c  
calling Write and  
handling the  
BytesReceived even  
(Defined by  
[ExtensionMethods](#).)



## RequestAsyncTMessage(TMessage, Int32)

Overloaded.  
Sends a request to t  
server and waits for  
response. Can be us  
along with, or in lieu  
the regular method c  
calling Write and  
handling the  
BytesReceived even  
(Defined by  
[ExtensionMethods](#).)



## Unlock

Performs a Monitor  
unlock  
(Defined by

[MonitorExtensions.\)](#)

---

[Top](#)

## ▲ See Also

### Reference

[TcpGenericTcpClientTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericTcpClientTMessageOnReceived Method

[Missing <summary> documentation for "M:W.Net.Tcp.Generic.TcpClient`1.OnReceived(System.Byte[]@)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
protected override void OnReceived(  
    ref byte[] bytes  
)
```

[Copy](#)

## Parameters

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for "M:W.Net.Tcp.Generic.TcpClient`1.OnReceived(System.Byte[]@)"]

## ► See Also

[Reference](#)

[TcpGenericTcpClientTMessage Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpClientTMessageWrite Method

## ► Overload List

	Name	Description
≡	<a href="#">Write(TMessage)</a>	
≡	<a href="#">Write(Byte)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## ► See Also

Reference

[TcpGenericTcpClientTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericTcpClientTMessageWrite Method (*TMessage*)

[Missing <summary> documentation for "M:W.Net.Tcp.Generic.TcpClient`1.Write(`0)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public void Write(  
    TMessage message  
)
```

[Copy](#)

## Parameters

*message*

Type: [TMessage](#)

[Missing <param name="message"/> documentation for  
"M:W.Net.Tcp.Generic.TcpClient`1.Write(`0)"]

## ► See Also

### Reference

[TcpGenericTcpClientTMessage Class](#)

[Write Overload](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpClientTMessage Events

The [TcpGenericTcpClientTMessage](#) generic type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">BytesReceived</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Connected</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Disconnected</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">MessageReceived</a>	

[Top](#)

## ▪ See Also

### Reference

[TcpGenericTcpClientTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericTcpClient $TMessage$ MessageReceived Event

[Missing <summary> documentation for "E:W.Net.Tcp.Generic.TcpClient`1.MessageReceived"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public event Action<TcpGenericTcpClient<TMessage>
```

**Value**

Type: [SystemAction](#) $TcpGenericTcpClient$  $TMessage$ ,  $TMessage$

## ► See Also

Reference

[TcpGenericTcpClient](#) $TMessage$  Class

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpGenericTcpHostTMessage Class

[Missing <summary> documentation for "T:W.Net.Tcp.Generic.TcpHost`1"]

## ► Inheritance Hierarchy

SystemObject W.NetTcpTcpHost  
W.NetTcpGenericTcpHostTMessage

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public class TcpHost<TMessage> : TcpTcpHost
```

## Type Parameters

*TMessage*

[Missing <typeparam name="TMessage"/> documentation for "T:W.Net.Tcp.Generic.TcpHost`1"]

The [TcpGenericTcpHostTMessage](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">TcpGenericTcpHostTMessage</a>	Initializes a new instance of the

[Top](#)

## ► Properties

	Name	Description
	<a href="#">IsListening</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">OnCreateServer</a>	(Inherited from <a href="#">TcpTcpHost</a> .)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">Listen(IPEndPoint)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)

---

	<a href="#">Listen(IPEndPoint, Int32)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">OnBytesReceived</a>	(Overrides <a href="#">TcpTcpHost.OnBytesReceived(TcpIClientByte)</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## Events

Name	Description
	<a href="#">BytesReceived</a> (Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">MessageReceived</a>

---

[Top](#)

## Extension Methods

Name	Description
 	<a href="#">AsTType</a> Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a> Scans the fields and properties of "owner" and

sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members  
(Defined by [PropertyHostExtensions](#).)

---

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty

(Defined by  
[PropertyHostExtensions](#).)



### [Lock](#)

Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and  
property of type `IProperty`  
and sets it's `IsDirty` flag to  
false  
(Defined by  
[PropertyHostExtensions](#).)



### [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

[Top](#)

## ► See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericTcpHostTMessage Constructor

Initializes a new instance of the [TcpGenericTcpHostTMessage](#) class

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

```
public TcpHost()
```

[Copy](#)

## ► See Also

[Reference](#)

[TcpGenericTcpHostTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpHostTMessage Properties

The [TcpGenericTcpHostTMessage](#) generic type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">IsListening</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">OnCreateServer</a>	(Inherited from <a href="#">TcpTcpHost</a> .)

[Top](#)

## ► See Also

Reference

[TcpGenericTcpHostTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpHostTMessage Methods

The [TcpGenericTcpHostTMessage](#) generic type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Listen(IPEndPoint)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
≡	<a href="#">Listen(IPEndPoint, Int32)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)



[OnBytesReceived](#) (Overrides [TcpTcpHostOnBytesReceived\(TcpIClie  
Byte\).](#))



[ToString](#) Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## ◀ Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in

		a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ◀ See Also

[Reference](#)

[TcpGenericTcpHostTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericTcpHostTMessageOnBytesReceived Method

[Missing <summary> documentation for "M:W.Net.Tcp.Generic.TcpHost`1.OnBytesReceived(W.Net.Tcp.IClient,System.Byte[])"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
protected override void OnBytesReceived(
    TcpIClient client,
    byte[] bytes
)
```

## Parameters

*client*

Type: [W.NetTcpIClient](#)

[Missing <param name="client"/> documentation for "M:W.Net.Tcp.Generic.TcpHost`1.OnBytesReceived(W.Net.Tcp.IClient,System.Byte[])"]

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for "M:W.Net.Tcp.Generic.TcpHost`1.OnBytesReceived(W.Net.Tcp.IClient,System.Byte[])"]

## ► See Also

[Reference](#)

[TcpGenericTcpHostTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpHostTMessage Events

The [TcpGenericTcpHostTMessage](#) generic type exposes the following members.

## ▪ Events

	Name	Description
	<a href="#">BytesReceived</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">MessageReceived</a>	

[Top](#)

## ▪ See Also

[Reference](#)

[TcpGenericTcpHostTMessage Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpGenericTcpHost $<TMessage>$ .MessageReceived Event

[Missing <summary> documentation for "E:W.Net.Tcp.Generic.TcpHost`1.MessageReceived"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public event Action<TcpGenericTcpHost<TMessage>,
```

### Value

Type: [SystemAction<TcpGenericTcpHost<TMessage>, TcpGenericTcpClient<TMessage>, TMessage>](#)

## ► See Also

### Reference

[TcpGenericTcpHost<TMessage> Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpIClient Interface

[Missing <summary> documentation for "T:W.Net.Tcp.IClient"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public interface IClient : IDisposable
```

The [TcpIClient](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Socket</a>	

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Connect</a>	
	<a href="#">Dispose</a>	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from <a href="#">IDisposable</a> .)
	<a href="#">Write</a>	

[Top](#)

## Events

Name	Description
 BytesReceived	
 Connected	
 Disconnected	

[Top](#)

## See Also

Reference

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IClient Properties

The [TcpIClient](#) type exposes the following members.

## ► Properties

Name	Description
 <a href="#">Socket</a>	

[Top](#)

## ► See Also

[Reference](#)

[TcpIClient Interface](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpIClientSocket Property

[Missing <summary> documentation for "P:W.Net.Tcp.IClient.Socket"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
Socket Socket { get; }
```

Property Value

Type: [Socket](#)

## ► See Also

[Reference](#)

[TcpIClient Interface](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IClient Methods

The [TcpIClient](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">Connect</a>	
 <a href="#">Dispose</a>	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. (Inherited from <a href="#">IDisposable</a> .)
 <a href="#">Write</a>	

[Top](#)

## ▪ See Also

[Reference](#)

[TcpIClient Interface](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpIClientConnect Method

[Missing <summary> documentation for  
"M:W.Net.Tcp.IClient.Connect(System.Net.IPEndPoint)"]

**Namespace:** W.Net

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

```
void Connect(  
    IPEndPoint ep  
)
```

Copy

## Parameters

ep

Type: System.NetIPEndPoint

[Missing <param name="ep"/> documentation for  
"M:W.Net.Tcp.IClient.Connect(System.Net.IPEndPoint)"]

## ► See Also

Reference

[TcpIClient Interface](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpIClientWrite Method

[Missing <summary> documentation for "M:W.Net.Tcp.IClient.Write(System.Byte[])"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
void Write(  
    byte[] bytes  
)
```

## Parameters

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for  
"M:W.Net.Tcp.IClient.Write(System.Byte[])"]

## ► See Also

[Reference](#)

[TcpIClient Interface](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# IClient Events

The [TcpIClient](#) type exposes the following members.

## Events

Name	Description
 BytesReceived	
 Connected	
 Disconnected	

[Top](#)

## See Also

Reference

[TcpIClient Interface](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpIClientBytesReceived Event

[Missing <summary> documentation for "E:W.Net.Tcp.IClient.BytesReceived"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
event Action<TcpIClient, byte[]> BytesReceived
```

**Value**

Type: [SystemActionTcpIClient, Byte](#)

## ► See Also

**Reference**

[TcpIClient Interface](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpIClientConnected Event

[Missing <summary> documentation for "E:W.Net.Tcp.IClient.Connected"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
event Action<TcpIClient> Connected
```

Value

Type: [SystemActionTcpIClient](#)

## ► See Also

Reference

[TcpIClient Interface](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpIClientDisconnected Event

[Missing <summary> documentation for "E:W.Net.Tcp.IClient.Disconnected"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
event Action<TcpIClient> Disconnected
```

Value

Type: [SystemActionTcpIClient](#)

## ► See Also

Reference

[TcpIClient Interface](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpIInitialize Interface

[Missing <summary> documentation for "T:W.Net.Tcp.IInitialize"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public interface IInitialize
```

The [TcpIInitialize](#) type exposes the following members.

## ► Methods

	Name	Description
	<a href="#">Initialize</a>	

[Top](#)

## ► See Also

[Reference](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# IInitialize Methods

The [TcpIInitialize](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">Initialize</a>	

[Top](#)

## ▪ See Also

[Reference](#)

[TcpIInitialize Interface](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpIInitialize.Initialize Method

[Missing <summary> documentation for "M:W.Net.Tcp.IInitialize.Initialize(System.Object[])"]

**Namespace:** W.Net

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
_____  
bool Initialize(  
    params Object[] args  
)
```

Copy

### Parameters

*args*

Type: SystemObject

[Missing <param name="args"/> documentation for "M:W.Net.Tcp.IInitialize.Initialize(System.Object[])"]

### Return Value

Type: Boolean

[Missing <returns> documentation for "M:W.Net.Tcp.IInitialize.Initialize(System.Object[])"]

## ► See Also

Reference

[TcpIInitialize Interface](#)

[W.Net Namespace](#)



## Tungsten

W<sup>74</sup>

# TcpSecureTcpClient Class

[Missing <summary> documentation for "T:W.Net.Tcp.SecureTcpClient"]

## ► Inheritance Hierarchy

```
SystemObject W.NetTcpTcpClient
  W.NetTcpSecureTcpClient
    W.NetTcpGenericSecureTcpClientTMessage
```

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class SecureTcpClient : TcpTcpClient
```

The [TcpSecureTcpClient](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">TcpSecureTcpClient</a>	Initializes a new instance of the <a href="#">TcpSecureTcpClient</a> class

[Top](#)

## ► Properties

	Name	Description
--	------	-------------



## Encryption



### IsSecure



### IsServerSide

(Inherited from [TcpTcpClient](#).)



### Socket

(Inherited from [TcpTcpClient](#).)

[Top](#)

## Methods

Name	Description
<a href="#">Connect</a>	
<a href="#">Disconnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
<a href="#">Disconnect(Boolean, Boolean)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)

	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnConnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
💡	<a href="#">OnInitialize</a>	(Overrides <a href="#">TcpTcpClientOnInitialize(Object)</a> .)
💡	<a href="#">OnReceived</a>	(Overrides <a href="#">TcpTcpClientOnReceived(Byte)</a> .)
💡	<a href="#">OnSend</a>	(Overrides <a href="#">TcpTcpClientOnSend(Byte)</a> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Write</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## Events

	Name	Description
⚡	<a href="#">BytesReceived</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
⚡	<a href="#">Connected</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
⚡	<a href="#">Disconnected</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
⚡	<a href="#">SecureFailed</a>	

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock

(Defined by  
[MonitorExtensions](#).)



### [InLockAsyncTType\(FuncTType\)](#)

Overloaded.  
Asynchronously performs  
the action in a Monitor  
lock  
(Defined by  
[MonitorExtensions](#).)



### [IsDirty](#)

Scans the IsDirty value of  
each field and property of  
type IProperty  
(Defined by  
[PropertyHostExtensions](#).)



### [Lock](#)

Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and  
property of type IProperty  
and sets its IsDirty flag to  
false  
(Defined by  
[PropertyHostExtensions](#).)



### [RequestAsync\(Byte, Int32\)](#)

Overloaded.  
Sends a request to the  
server and waits for a  
response. Can be used  
along with, or in lieu of,  
the regular method of  
calling Write and  
handling the  
BytesReceived event.  
(Defined by  
[ExtensionMethods](#).)



## RequestAsync(Byte, Int32)

Overloaded.

Sends a request to the server and waits for a response. Can be used along with, or in lieu of, the regular method of calling Write and handling the BytesReceived event.  
(Defined by [ExtensionMethods](#).)



## Unlock

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpClient Constructor

Initializes a new instance of the [TcpSecureTcpClient](#) class

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public SecureTcpClient(  
    int keySize  
)
```

### Parameters

*keySize*

Type: [System.Int32](#)

[Missing <param name="keySize"/> documentation for  
"M:W.Net.Tcp.SecureTcpClient.#ctor(System.Int32)"]

## ► See Also

Reference

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# SecureTcpClient Properties

The [TcpSecureTcpClient](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">Encryption</a>	
	<a href="#">IsSecure</a>	
	<a href="#">IsServerSide</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
	<a href="#">Socket</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## See Also

### Reference

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpClientEncryption Property

[Missing <summary> documentation for "P:W.Net.Tcp.SecureTcpClient.Encryption"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected AssymmetricEncryption Encryption { get;
```

Property Value

Type: [AssymmetricEncryption](#)

## ► See Also

Reference

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpClientIsSecure Property

[Missing <summary> documentation for "P:W.Net.Tcp.SecureTcpClient.IsSecure"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected bool IsSecure { get; }
```

Property Value

Type: [Boolean](#)

## ► See Also

Reference

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SecureTcpClient Methods

The [TcpSecureTcpClient](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">Connect</a>	
 <a href="#">Disconnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
 <a href="#">Disconnect(Boolean, Boolean)</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
 <a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)

💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnConnect</a>	(Inherited from <a href="#">TcpTcpClient</a> .)
💡	<a href="#">OnInitialize</a>	(Overrides <a href="#">TcpTcpClientOnInitialize(Object)</a> .)
💡	<a href="#">OnReceived</a>	(Overrides <a href="#">TcpTcpClientOnReceived(Byte)</a> .)
💡	<a href="#">OnSend</a>	(Overrides <a href="#">TcpTcpClientOnSend(Byte)</a> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Write</a>	(Inherited from <a href="#">TcpTcpClient</a> .)

[Top](#)

## ◀ Extension Methods

Name	Description
💡  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡  <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has

[IOwnedProperty](#)  
members  
(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type <a href="#">IProperty</a> (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock

(Defined by  
[MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets its `IsDirty` flag to `false`

(Defined by  
[PropertyHostExtensions](#).)



### [RequestAsync\(Byte, Int32\)](#)

Overloaded.  
Sends a request to the server and waits for a response. Can be used along with, or in lieu of, the regular method of calling `Write` and handling the `BytesReceived` event.

(Defined by  
[ExtensionMethods](#).)



### [RequestAsync\(Byte, Int32\)](#)

Overloaded.  
Sends a request to the server and waits for a response. Can be used along with, or in lieu of, the regular method of calling `Write` and handling the `BytesReceived` event.

(Defined by  
[ExtensionMethods](#).)



### [Unlock](#)

Performs a Monitor unlock  
(Defined by  
[MonitorExtensions](#).)

[Top](#)

## ◀ See Also

[Reference](#)

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpClientConnect Method

[Missing <summary> documentation for "M:W.Net.Tcp.SecureTcpClient.Connect(System.Net.IPEndPoint)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public void Connect(  
    IPEndPoint ep  
)
```

[Copy](#)

## Parameters

*ep*

Type: [System.NetIPEndPoint](#)

[Missing <param name="ep"/> documentation for "M:W.Net.Tcp.SecureTcpClient.Connect(System.Net.IPEndPoint)"]

## ► See Also

[Reference](#)

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpSecureTcpClientOnInitialize Method

[Missing <summary> documentation for "M:W.Net.Tcp.SecureTcpClient.OnInitialize(System.Object[])"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
protected override bool OnInitialize(  
    params Object[] args  
)
```

[Copy](#)

## Parameters

*args*

Type: [SystemObject](#)

[Missing <param name="args"/> documentation for "M:W.Net.Tcp.SecureTcpClient.OnInitialize(System.Object[])"]

## Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:W.Net.Tcp.SecureTcpClient.OnInitialize(System.Object[])"]

## ► See Also

[Reference](#)

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpClientOnReceived Method

[Missing <summary> documentation for  
"M:W.Net.Tcp.SecureTcpClient.OnReceived(System.Byte[]@)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
protected override void OnReceived(
    ref byte[] bytes
)
```

### Parameters

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for  
"M:W.Net.Tcp.SecureTcpClient.OnReceived(System.Byte[]@)"]

## ► See Also

[Reference](#)

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpSecureTcpClientOnSend Method

[Missing <summary> documentation for "M:W.Net.Tcp.SecureTcpClient.OnSend(System.Byte[]@)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
protected override void OnSend(
    ref byte[] bytes
)
```

### Parameters

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for "M:W.Net.Tcp.SecureTcpClient.OnSend(System.Byte[]@)"]

## ► See Also

[Reference](#)

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# SecureTcpClient Events

The [TcpSecureTcpClient](#) type exposes the following members.

## Events

Name	Description
 BytesReceived	(Inherited from <a href="#">TcpTcpClient</a> .)
 Connected	(Inherited from <a href="#">TcpTcpClient</a> .)
 Disconnected	(Inherited from <a href="#">TcpTcpClient</a> .)
 SecureFailed	

[Top](#)

## See Also

### Reference

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpClientSecureFailed Event

[Missing <summary> documentation for "E:W.Net.Tcp.SecureTcpClient.SecureFailed"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public event Action<TcpSecureTcpClient> SecureFai
```

Value

Type: [SystemActionTcpSecureTcpClient](#)

## ► See Also

Reference

[TcpSecureTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpSecureTcpHost Class

[Missing <summary> documentation for "T:W.Net.Tcp.SecureTcpHost"]

## ► Inheritance Hierarchy

SystemObject W.NetTcpTcpHost  
W.NetTcpSecureTcpHost

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class SecureTcpHost : TcpTcpHost
```

The [TcpSecureTcpHost](#) type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">TcpSecureTcpHost</a>	Initializes a new instance of the <a href="#">TcpSecureTcpHost</a> class

[Top](#)

## ► Properties

Name	Description
 <a href="#">IsListening</a>	(Inherited from <a href="#">TcpTcpHost</a> .)



[OnCreateServer](#) (Inherited from [TcpTcpHost](#).)

[Top](#)

## Methods

	Name	Description
≡	<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Listen(IPEndPoint)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
≡	<a href="#">Listen(IPEndPoint, Int32)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

---

	<a href="#">OnBytesReceived</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## Events

Name	Description
	<a href="#">BytesReceived</a> (Inherited from <a href="#">TcpTcpHost</a> .)

---

[Top](#)

## Extension Methods

Name	Description
	<a href="#">AsTType</a> Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a> Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a> Overloaded.

		Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
⬇️	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false

(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpHost Constructor

Initializes a new instance of the [TcpSecureTcpHost](#) class

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public SecureTcpHost(  
    int keySize  
)
```

[Copy](#)

### Parameters

*keySize*

Type: [System.Int32](#)

[Missing <param name="keySize"/> documentation for  
"M:W.Net.Tcp.SecureTcpHost.#ctor(System.Int32)"]

## ► See Also

[Reference](#)

[TcpSecureTcpHost Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# SecureTcpHost Properties

The [TcpSecureTcpHost](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">IsListening</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">OnCreateServer</a>	(Inherited from <a href="#">TcpTcpHost</a> .)

[Top](#)

## ► See Also

Reference

[TcpSecureTcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SecureTcpHost Methods

The [TcpSecureTcpHost](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Dispose</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Listen(IPEndPoint)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
≡	<a href="#">Listen(IPEndPoint, Int32)</a>	(Inherited from <a href="#">TcpTcpHost</a> .)

	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">OnBytesReceived</a>	(Inherited from <a href="#">TcpTcpHost</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ◀ Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[TcpSecureTcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SecureTcpHost Events

The [TcpSecureTcpHost](#) type exposes the following members.

## Events

Name	Description
 BytesReceived	(Inherited from <a href="#">TcpTcpHost</a> .)

[Top](#)

## See Also

[Reference](#)

[TcpSecureTcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpLogger Class

Sends log messages to a remote server via secure Tcp

## ► Inheritance Hierarchy

[SystemObject](#) [W.NetTcpSecureTcpLogger](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static class SecureTcpLogger
```

## ► Methods

	Name	Description
 	<a href="#">LogTheMessage(IPEndPoint, Int32, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call
 	<a href="#">LogTheMessage(String, Int32, Int32, String)</a>	Log a message to the remote machine. Note that this message is slower due to

parsing the  
remoteIP with  
each call

---

[Top](#)

## ▲ Examples

```
Log.LogTheMessage += (category, message) =>  
    W.Net.Tcp.SecureTcpLogger.LogTheMessage("127.0.0.1", 5555,  
    true, category, message);
```

## ▲ See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SecureTcpLogger Methods

## ▲ Methods

	Name	Description
 	<a href="#">LogTheMessage(IPEndPoint, Int32, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call
 	<a href="#">LogTheMessage(String, Int32, Int32, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

[Top](#)

## ▲ See Also

Reference

[TcpSecureTcpLogger Class](#)  
[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# SecureTcpLoggerLogTheMessage Method

## Overload List

Name	Description
  <a href="#">LogTheMessage(IPEndPoint, Int32, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call
  <a href="#">LogTheMessage(String, Int32, Int32, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

[Top](#)

## See Also

### Reference

[TcpSecureTcpLogger Class](#)  
[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpLoggerLogTheMessage Method (IPEndPoint, Int32, String)

Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static void LogTheMessage(
    IPPEndPoint remoteIPEndPoint,
    int keySize,
    string message
)
```

### Parameters

*remoteIPEndPoint*

Type: [System.NetIPPEndPoint](#)

The IPEndPoint of the remote server

*keySize*

Type: [SystemInt32](#)

The encryption key size

*message*

Type: [SystemString](#)

The log message

## ► See Also

Reference

TcpSecureTcpLogger Class  
LogTheMessage Overload  
W.Net Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpSecureTcpLoggerLogTheMessage Method (String, Int32, Int32, String)

Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static void LogTheMessage(
    string remoteIP,
    int remotePort,
    int keySize,
    string message
)
```

## Parameters

*remoteIP*

Type: [SystemString](#)

The IP address or name of the remote server

*remotePort*

Type: [SystemInt32](#)

The port of the remote Udp server

*keySize*

Type: [SystemInt32](#)

The encryption key size

*message*

Type: [SystemString](#)

The log message

## ▲ See Also

### Reference

[TcpSecureTcpLogger Class](#)

[LogTheMessage Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpClient Class

[Missing <summary> documentation for "T:W.Net.Tcp.TcpClient"]

## ► Inheritance Hierarchy

```
SystemObject W.NetTcpTcpClient
  W.NetTcpGenericTcpClientTMessage
    W.NetTcpSecureTcpClient
```

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public class TcpClient : TcpIInitialize, TcpIClient
    IDisposable
```

The [TcpTcpClient](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">TcpTcpClient</a>	Initializes a new instance of the <a href="#">TcpTcpClient</a> class

[Top](#)

## ► Properties

	Name	Description
--	------	-------------



[IsServerSide](#)



[Socket](#)

[Top](#)

## ▲ Methods

	Name	Description
≡	<a href="#">Connect</a>	
≡	<a href="#">Disconnect</a>	
💡	<a href="#">Disconnect(Boolean, Boolean)</a>	
≡	<a href="#">Dispose</a>	Releases all resources used by the <a href="#">TcpClient</a>
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance.

(Inherited from [Object](#).)

💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">OnConnect</a>	
💡	<a href="#">OnInitialize</a>	
💡	<a href="#">OnReceived</a>	
💡	<a href="#">OnSend</a>	
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Write</a>	

[Top](#)

## Events

Name	Description
⚡	<a href="#">BytesReceived</a>
⚡	<a href="#">Connected</a>
⚡	<a href="#">Disconnected</a>

[Top](#)

## Extension Methods

Name	Description

 	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor

`lock`  
(Defined by  
[MonitorExtensions](#).)

---

 [IsDirty](#) Scans the `IsDirty` value of each field and property of type `IProperty`  
(Defined by  
[PropertyHostExtensions](#).)

---

 [Lock](#) Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

---

 [MarkAsClean](#) Scans each field and property of type `IProperty` and sets its `IsDirty` flag to false  
(Defined by  
[PropertyHostExtensions](#).)

---

 [RequestAsync](#) Sends a request to the server and waits for a response. Can be used along with, or in lieu of, the regular method of calling `Write` and handling the `BytesReceived` event.  
(Defined by  
[ExtensionMethods](#).)

---

 [Unlock](#) Performs a Monitor unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

Reference

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpClient Constructor

Initializes a new instance of the [TcpTcpClient](#) class

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

```
public TcpClient()
```

[Copy](#)

## ► See Also

[Reference](#)

[TcpTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpClient Properties

The [TcpClient](#) type exposes the following members.

## Properties

Name	Description
 <a href="#">IsServerSide</a>	
 <a href="#">Socket</a>	

[Top](#)

## See Also

[Reference](#)

[TcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpClientIsServerSide Property

[Missing <summary> documentation for "P:W.Net.Tcp.TcpClient.IsServerSide"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

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

Property Value

Type: [Boolean](#)

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpClientSocket Property

[Missing <summary> documentation for "P:W.Net.Tcp.TcpClient.Socket"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public Socket Socket { get; }
```

Property Value

Type: [Socket](#)

Implements

[TcpIClientSocket](#)

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpClient Methods

The [TcpClient](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Connect</a>	
≡	<a href="#">Disconnect</a>	
💡	<a href="#">Disconnect(Boolean, Boolean)</a>	
≡	<a href="#">Dispose</a>	
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance.

(Inherited from [Object](#).)

	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">OnConnect</a>	
	<a href="#">OnInitialize</a>	
	<a href="#">OnReceived</a>	
	<a href="#">OnSend</a>	
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Write</a>	

[Top](#)

## ► Extension Methods

Name	Description
  <a href="#">AsTTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has

[IOwnedProperty](#)  
members  
(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type <a href="#">IProperty</a> (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock

(Defined by  
[MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets its `IsDirty` flag to `false`

(Defined by  
[PropertyHostExtensions](#).)



### [RequestAsync](#)

Sends a request to the server and waits for a response. Can be used along with, or in lieu of, the regular method of calling `Write` and handling the `BytesReceived` event.

(Defined by  
[ExtensionMethods](#).)



### [Unlock](#)

Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

[Top](#)

## ▲ See Also

[Reference](#)

[TcpClient Class](#)  
[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpTcpClientConnect Method

[Missing <summary> documentation for  
"M:W.Net.Tcp.TcpClient.Connect(System.Net.IPEndPoint)"]

**Namespace:** W.Net

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public void Connect(  
    IPEndPoint ep  
)
```

## Parameters

ep

Type: System.Net.IPEndPoint

[Missing <param name="ep"/> documentation for  
"M:W.Net.Tcp.TcpClient.Connect(System.Net.IPEndPoint)"]

## Implements

[TcpIClientConnect\(IPEndPoint\)](#)

## ► See Also

### Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpClientDisconnect Method

## ► Overload List

Name	Description
 <a href="#">Disconnect</a>	
 <a href="#">Disconnect(Boolean, Boolean)</a>	

[Top](#)

## ► See Also

[Reference](#)

[TcpTcpClient Class](#)  
[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpClientDisconnect Method

[Missing <summary> documentation for "M:W.Net.Tcp.TcpClient.Disconnect"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public void Disconnect()
```

## ► See Also

[Reference](#)

[TcpTcpClient Class](#)

[Disconnect Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpClientDisconnect Method (Boolean, Boolean)

[Missing <summary> documentation for "M:W.Net.Tcp.TcpClient.Disconnect(System.Boolean,System.Boolean)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
protected void Disconnect(
    bool notifyRemote,
    bool waitForThreadToExit
)
```

### Parameters

*notifyRemote*

Type: [SystemBoolean](#)

[Missing <param name="notifyRemote"/> documentation for "M:W.Net.Tcp.TcpClient.Disconnect(System.Boolean,System.Boolean)"]

*waitForThreadToExit*

Type: [SystemBoolean](#)

[Missing <param name="waitForThreadToExit"/> documentation for "M:W.Net.Tcp.TcpClient.Disconnect(System.Boolean,System.Boolean)"]

## ► See Also

[Reference](#)

[TcpTcpClient Class](#)

[Disconnect Overload](#)

## [W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpClientDispose Method

Releases all resources used by the [TcpTcpClient](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public virtual void Dispose()
```

Implements

[IDisposableDispose](#)

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpClientOnConnect Method

[Missing <summary> documentation for  
"M:W.Net.Tcp.TcpClient.OnConnect(System.Object[])"]

**Namespace:** W.Net

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected virtual void OnConnect(  
    params Object[] args  
)
```

### Parameters

args

Type: SystemObject

[Missing <param name="args"/> documentation for  
"M:W.Net.Tcp.TcpClient.OnConnect(System.Object[])"]

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpTcpClientOnInitialize Method

[Missing <summary> documentation for  
"M:W.Net.Tcp.TcpClient.OnInitialize(System.Object[])"]

**Namespace:** W.Net

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected virtual bool OnInitialize(  
    params Object[] args  
)
```

### Parameters

args

Type: SystemObject

[Missing <param name="args"/> documentation for  
"M:W.Net.Tcp.TcpClient.OnInitialize(System.Object[])"]

### Return Value

Type: Boolean

[Missing <returns> documentation for  
"M:W.Net.Tcp.TcpClient.OnInitialize(System.Object[])"]

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)



## Tungsten

W<sup>74</sup>

# TcpTcpClientOnReceived Method

[Missing <summary> documentation for "M:W.Net.Tcp.TcpClient.OnReceived(System.Byte[]@)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
protected virtual void OnReceived(
    ref byte[] bytes
)
```

### Parameters

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for "M:W.Net.Tcp.TcpClient.OnReceived(System.Byte[]@)"]

## ► See Also

[Reference](#)

[TcpTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpTcpClientOnSend Method

[Missing <summary> documentation for  
"M:W.Net.Tcp.TcpClient.OnSend(System.Byte[]@)"]

**Namespace:** W.Net

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected virtual void OnSend(  
    ref byte[] bytes  
)
```

### Parameters

*bytes*

Type: SystemByte

[Missing <param name="bytes"/> documentation for  
"M:W.Net.Tcp.TcpClient.OnSend(System.Byte[]@)"]

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpTcpClientWrite Method

[Missing <summary> documentation for "M:W.Net.Tcp.TcpClient.Write(System.Byte[])"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public void Write(  
    byte[] bytes  
)
```

[Copy](#)

### Parameters

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for  
"M:W.Net.Tcp.TcpClient.Write(System.Byte[])"]

### Implements

[TcpIClientWrite\(Byte\)](#)

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpClient Events

The [TcpClient](#) type exposes the following members.

## ▪ Events

Name	Description
 BytesReceived	
 Connected	
 Disconnected	

[Top](#)

## ▪ See Also

Reference

[TcpClient Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpClientBytesReceived Event

[Missing <summary> documentation for "E:W.Net.Tcp.TcpClient.BytesReceived"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public event Action<TcpIClient, byte[]> BytesReceived;
```

Value

Type: [SystemActionTcpIClient, Byte](#)

Implements

[TcpIClientBytesReceived](#)

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpTcpClientConnected Event

[Missing <summary> documentation for "E:W.Net.Tcp.TcpClient.Connected"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public event Action<TcpIClient> Connected
```

Value

Type: [SystemActionTcpIClient](#)

Implements

[TcpIClientConnected](#)

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpTcpClientDisconnected Event

[Missing <summary> documentation for "E:W.Net.Tcp.TcpClient.Disconnected"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public event Action<TcpIClient> Disconnected
```

Value

Type: [SystemActionTcpIClient](#)

Implements

[TcpIClientDisconnected](#)

## ► See Also

Reference

[TcpTcpClient Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpTcpHost Class

[Missing <summary> documentation for "T:W.Net.Tcp.TcpHost"]

## ► Inheritance Hierarchy

```
SystemObject W.NetTcpTcpHost
  W.NetTcpGenericSecureTcpHostTMessage
  W.NetTcpGenericTcpHostTMessage
  W.NetTcpSecureTcpHost
```

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class TcpHost : IDisposable
```

The `TcpTcpHost` type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">TcpHost</a>	Initializes a new instance of the <code>TcpHost</code> class

[Top](#)

## ► Properties

Name	Description
------	-------------



[IsListening](#)



[OnCreateServer](#)

[Top](#)

## ► Methods

	Name	Description
≡	<a href="#">Dispose</a>	Releases all resources used by the <code>TcpTcpHost</code>
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <code>Object</code> .)
≡	<a href="#">GetType</a>	Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code> .)
≡	<a href="#">Listen(IPEndPoint)</a>	
≡	<a href="#">Listen(IPEndPoint, Int32)</a>	

**MemberwiseClone** Creates a shallow copy of the current [Object](#).  
(Inherited from [Object](#).)

---

💡 [OnBytesReceived](#)

---

≡ [ToString](#) Returns a string that represents the current object.  
(Inherited from [Object](#).)

---

[Top](#)

## ► Events

Name	Description
⚡ <a href="#">BytesReceived</a>	

[Top](#)

## ► Extension Methods

Name	Description
💡 ⚡ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by

[PropertyHostExtensions.\)](#)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	Scans each field and

## [MarkAsClean](#)

property of type [IProperty](#)  
and sets it's `IsDirty` flag to  
`false`

(Defined by  
[PropertyHostExtensions](#).)



## [Unlock](#)

Performs a Monitor  
unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpHost Constructor

Initializes a new instance of the [TcpTcpHost](#) class

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public TcpHost()
```

## ► See Also

[Reference](#)

[TcpTcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpHost Properties

The [TcpHost](#) type exposes the following members.

## Properties

Name	Description
 <a href="#">IsListening</a>	
 <a href="#">OnCreateServer</a>	

[Top](#)

## See Also

[Reference](#)

[TcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpHostIsListening Property

[Missing <summary> documentation for "P:W.Net.Tcp.TcpHost.IsListening"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public bool IsListening { get; }
```

Property Value

Type: [Boolean](#)

## ► See Also

[Reference](#)

[TcpTcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpHostOnCreateServer Property

[Missing <summary> documentation for "P:W.Net.Tcp.TcpHost.OnCreateServer"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected virtual Func<Socket, TcpIClient> OnCreateServer { get; set; }
```

Property Value

Type: [Func<Socket, TcpIClient>](#)

## ► See Also

Reference

[TcpTcpHost Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpHost Methods

The [TcpHost](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Dispose</a>	
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Listen(IPEndPoint)</a>	
≡	<a href="#">Listen(IPEndPoint, Int32)</a>	

**MemberwiseClone** Creates a shallow copy of the current [Object](#).  
(Inherited from [Object](#).)

---

💡 [OnBytesReceived](#)

---

≡ [ToString](#) Returns a string that represents the current object.  
(Inherited from [Object](#).)

---

[Top](#)

## ► Extension Methods

Name	Description
💡 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
💡 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ◀ See Also

[Reference](#)

[TcpTcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpHostDispose Method

Releases all resources used by the [TcpTcpHost](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements

[IDisposableDispose](#)

## ► See Also

Reference

[TcpTcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpHostListen Method

## ► Overload List

Name	Description
 <a href="#">Listen(IPEndPoint)</a>	
 <a href="#">Listen(IPEndPoint, Int32)</a>	

[Top](#)

## ► See Also

[Reference](#)

[TcpTCPHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpHostListen Method (IPEndPoint)

[Missing <summary> documentation for "M:W.Net.Tcp.TcpHost.Listen(System.Net.IPEndPoint)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public void Listen(  
    IPEndPoint ep  
)
```

[Copy](#)

## Parameters

*ep*

Type: [System.NetIPEndPoint](#)

[Missing <param name="ep"/> documentation for "M:W.Net.Tcp.TcpHost.Listen(System.Net.IPEndPoint)"]

## ► See Also

[Reference](#)

[TcpTcpHost Class](#)

[Listen Overload](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpTcpHostListen Method (IPEndPoint, Int32)

[Missing <summary> documentation for  
"M:W.Net.Tcp.TcpHost.Listen(System.Net.IPEndPoint,System.Int32)"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public void Listen(  
    IPEndPoint ep,  
    int backlog  
)
```

[Copy](#)

## Parameters

*ep*

Type: [System.NetIPEndPoint](#)

[Missing <param name="ep"/> documentation for  
"M:W.Net.Tcp.TcpHost.Listen(System.Net.IPEndPoint,System.Int32)"]

*backlog*

Type: [SystemInt32](#)

[Missing <param name="backlog"/> documentation for  
"M:W.Net.Tcp.TcpHost.Listen(System.Net.IPEndPoint,System.Int32)"]

## ► See Also

[Reference](#)

[TcpTcpHost Class](#)

[Listen Overload](#)

## [W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpHost.OnBytesReceived Method

[Missing <summary> documentation for  
"M:W.Net.Tcp.TcpHost.OnBytesReceived(W.Net.Tcp.IClient,System.Byte[])"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
protected virtual void OnBytesReceived(
    TcpIClient client,
    byte[] bytes
)
```

## Parameters

*client*

Type: [W.NetTcpIClient](#)

[Missing <param name="client"/> documentation for  
"M:W.Net.Tcp.TcpHost.OnBytesReceived(W.Net.Tcp.IClient,System.Byte[])"]

*bytes*

Type: [SystemByte](#)

[Missing <param name="bytes"/> documentation for  
"M:W.Net.Tcp.TcpHost.OnBytesReceived(W.Net.Tcp.IClient,System.Byte[])"]

## ► See Also

[Reference](#)

[TcpTcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpHost Events

The [TcpHost](#) type exposes the following members.

## Events

Name	Description
 BytesReceived	

[Top](#)

## See Also

[Reference](#)

[TcpHost Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpHostBytesReceived Event

[Missing <summary> documentation for "E:W.Net.Tcp.TcpHost.BytesReceived"]

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public event Action<TcpTcpHost, TcpIClient, byte[]>
```

Value

Type: [SystemAction<TcpTcpHost, TcpIClient, Byte\[\]>](#)

## ► See Also

[Reference](#)

[TcpTcpHost Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpTcpLogger Class

Sends log messages to a remote server via Tcp

## ► Inheritance Hierarchy

[SystemObject](#) [W.NetTcpTcpLogger](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static class TcpLogger
```

## ► Methods

	Name	Description
 	<a href="#">LogTheMessage(IPEndPoint, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remotelIP with each call
 	<a href="#">LogTheMessage(String, Int32, String)</a>	Log a message to the custom logger. Note that this message is slower due to parsing the

remoteIP with  
each call

---

[Top](#)

## ▀ Examples

```
Log.LogTheMessage += (category, message) =>
    W.Net.Tcp.TcpLogger.LogTheMessage("127.0.0.1", 5555, true,
        category, message);
```

## ▀ See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpLogger Methods

## ► Methods

	Name	Description
 	<a href="#">LogTheMessage(IPEndPoint, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call
 	<a href="#">LogTheMessage(String, Int32, String)</a>	Log a message to the custom logger. Note that this message is slower due to parsing the remoteIP with each call

[Top](#)

## ► See Also

### Reference

[TcpTcpLogger Class](#)  
[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# TcpLoggerLogTheMessage Method

## ▪ Overload List

Name	Description
  <a href="#">LogTheMessage(IPEndPoint, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call
  <a href="#">LogTheMessage(String, Int32, String)</a>	Log a message to the custom logger. Note that this message is slower due to parsing the remoteIP with each call

[Top](#)

## ▪ See Also

Reference

[TcpTcpLogger Class](#)

[W.Net Namespace](#)



## Tungsten

W<sup>74</sup>

# TcpTcpLoggerLogTheMessage Method (IPEndPoint, String)

Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public static void LogTheMessage(  
    IPEndPoint remoteIPEndPoint,  
    string message  
)
```

[Copy](#)

### Parameters

*remoteIPEndPoint*

Type: [System.NetIPEndPoint](#)

The IPEndPoint of the remote server

*message*

Type: [SystemString](#)

The log message

## ► See Also

[Reference](#)

[TcpTcpLogger Class](#)

[LogTheMessage Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# TcpTcpLoggerLogTheMessage Method (String, Int32, String)

Log a message to the custom logger. Note that this message is slower due to parsing the remoteIP with each call

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static void LogTheMessage(
    string remoteIP,
    int remotePort,
    string message
)
```

### Parameters

*remoteIP*

Type: [SystemString](#)

The IP address or name of the remote server

*remotePort*

Type: [SystemInt32](#)

The port of the remote Udp server

*message*

Type: [SystemString](#)

The log message

## ► See Also

Reference

TcpTcpLogger Class  
LogTheMessage Overload  
W.Net Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Udp Class

Provides simple UDP reading and writing

## ► Inheritance Hierarchy

[SystemObject](#) [W.NetUdp](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static class Udp
```

## ► Methods

	Name	Description
≡ S	<a href="#">SendAsync(IPEndPoint, Byte)</a>	Sends data to a remote via UDP
≡ S	<a href="#">SendAsync(IPEndPoint, String, Object)</a>	Sends data to a remote via UDP
≡ S	<a href="#">SendAsyncTType(IPEndPoint, TType)</a>	Sends a message to a remote via UDP

[Top](#)

## ► See Also

## Reference

### [W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Udp Methods

## ↳ Methods

	Name	Description
≡  	<a href="#">SendAsync(IPEndPoint, Byte)</a>	Sends data to a remote via UDP
≡  	<a href="#">SendAsync(IPEndPoint, String, Object)</a>	Sends data to a remote via UDP
≡  	<a href="#">SendAsyncTType(IPEndPoint, TType)</a>	Sends a message to a remote via UDP

[Top](#)

## ↳ See Also

[Reference](#)

[Udp Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# UdpSendAsync Method

## ▪ Overload List

	Name	Description
≡ 	<a href="#">SendAsync(IPEndPoint, Byte)</a>	Sends data to a remote via UDP
≡ 	<a href="#">SendAsyncTType(IPEndPoint, TType)</a>	Sends a message to a remote via UDP
≡ 	<a href="#">SendAsync(IPEndPoint, String, Object)</a>	Sends data to a remote via UDP

[Top](#)

## ▪ See Also

[Reference](#)

[Udp Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpSendAsync Method (IPEndPoint, Byte)

Sends data to a remote via UDP

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static Task SendAsync(  
    IPEndPoint ipEndPoint,  
    byte[] bytes  
)
```

## Parameters

*ipEndPoint*

Type: [System.NetIPEndPoint](#)

The remote machine's endpoint

*bytes*

Type: [SystemByte](#)

The data to send

## Return Value

Type: [Task](#)

The Task associated with this action

## ► See Also

[Reference](#)

[Udp Class](#)

SendAsync Overload  
W.Net Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpSendAsync<TType> Method (IPEndPoint, TType)

Sends a message to a remote via UDP

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static Task<TType> SendAsync<TType>(
    IPEndPoint ipEndPoint,
    TType message
)
```

## Parameters

*ipEndPoint*

Type: [System.Net.IPEndPoint](#)

The remote machine's endpoint

*message*

Type: [TType](#)

The message to send

## Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for  
"M:W.Net.Udp.SendAsync`1(System.Net.IPEndPoint,`0)"]

## Return Value

Type: [Task](#)

The Task associated with this action

## See Also

[Reference](#)

[Udp Class](#)

[SendAsync Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpSendAsync Method (IPEndPoint, String, Object)

Sends data to a remote via UDP

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static Task SendAsync(  
    IPEndPoint ipEndPoint,  
    string format,  
    params Object[] args  
)
```

## Parameters

*ipEndPoint*

Type: [System.NetIPEndPoint](#)

The remote machine's endpoint

*format*

Type: [SystemString](#)

The string format

*args*

Type: [SystemObject](#)

String formatting arguments

## Return Value

Type: [Task](#)

The Task associated with this action

## ▲ See Also

Reference

[Udp Class](#)

[SendAsync Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpGeneric Class

Contains the generic implementation of W.Net.UdpPeer

## ► Inheritance Hierarchy

[SystemObject](#) [W.NetUdpGeneric](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

```
public static class Generic
```

[Copy](#)

## ► See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpGenericUdpPeerTType Class

A generic Udp peer

## ► Inheritance Hierarchy

[SystemObject](#) [W.NetUdpGenericUdpPeerTType](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public class UdpPeer<TType> : IDisposable
```

Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for  
"T:W.Net.Udp.Generic.UdpPeer`1"]

The [UdpGenericUdpPeerTType](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">UdpGenericUdpPeerTType</a>	Constructs a new UdpServer

[Top](#)

## ◀ Methods

	Name	Description
≡	<a href="#">Dispose</a>	Disposes the UdpServer and release resources
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">SendAsync</a>	Sends a message to the specified remote client
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Events

Name	Description
 <a href="#">MessageReceived</a>	Raised when a message has been received from a client

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

	InLockTType(FuncTType)	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
✿	InLockAsync(Action)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
✿	InLockAsyncTType(FuncTType)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
✿	IsDirty	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
✿	Lock	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
✿	MarkAsClean	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
✿	RequestAsyncTMessage	Sends a request to a generic Udp peer and

waits for a response. Can be used along with, or in lieu of, the regular method of calling Write and handling the BytesReceived event.  
(Defined by [ExtensionMethods](#).)



## Unlock

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpGenericUdpPeerTType Constructor

Constructs a new UdpServer

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public UdpPeer(  
    IPEndPoint localEndPoint,  
    bool useCompression  
)
```

## Parameters

*localEndPoint*

Type: [System.Net.IPEndPoint](#)

The local IPEndPoint on which to listen for data

*useCompression*

Type: [System.Boolean](#)

If True, messages will be compressed before sending and decompressed when received

## ► See Also

Reference

[UdpGenericUdpPeerTType Class](#)

[W.Net Namespace](#)



## Tungsten

W<sup>74</sup>

# UdpPeerTType Methods

The [UdpGenericUdpPeerTType](#) generic type exposes the following members.

## ▪ Methods

	Name	Description
	<a href="#">Dispose</a>	Disposes the UdpServer and release resources
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

---

	<a href="#">SendAsync</a>	Sends a message to the specified remote client
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## ◀ Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">RequestAsyncTMessage</a>	Sends a request to a generic Udp peer and waits for a response. Can be used along with, or in lieu of, the regular

method of calling Write and handling the BytesReceived event.  
(Defined by [ExtensionMethods](#).)

---



## [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

### Reference

[UdpGenericUdpPeerTType Class](#)  
[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpGenericUdpPeerTTypeDispose Method

Disposes the UdpServer and release resources

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

```
public void Dispose()
```

[Copy](#)

Implements

[IDisposable.Dispose](#)

## ► See Also

Reference

[UdpGenericUdpPeerTType Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpGenericUdpPeer`1.TTypeSendAsy Method

Sends a message to the specified remote client

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public Task SendAsync(  
    TType message,  
    IPEndPoint remoteEndPoint  
)
```

## Parameters

*message*

Type: [TType](#)

The message to send

*remoteEndPoint*

Type: [System.Net.IPEndPoint](#)

The remote client which is listening for messages

## Return Value

Type: [Task](#)

[Missing <returns> documentation for  
"M:W.Net.Udp.Generic.UdpPeer`1.SendAsync(`0,System.Net.IPEndPoint)"]

## ► See Also

### Reference

## UdpGenericUdpPeerTType Class W.Net Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpPeerTType Events

The [UdpGenericUdpPeerTType](#) generic type exposes the following members.

## Events

Name	Description
 <a href="#">MessageReceived</a>	Raised when a message has been received from a client

[Top](#)

## See Also

[Reference](#)

[UdpGenericUdpPeerTType Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpGenericUdpPeerTTypeMessageEvent

Raised when a message has been received from a client

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public event Action<IPEndPoint, TType> MessageRec
```

Value

Type: [SystemActionIPEndPoint, TType](#)

## ► See Also

Reference

[UdpGenericUdpPeerTType Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpUdpLogger Class

Sends log messages to a remote server via Udp

## ► Inheritance Hierarchy

[SystemObject](#) [W.NetUdpUdpLogger](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static class UdpLogger
```

## ► Methods

	Name	Description
 	<a href="#">LogTheMessage(IPEndPoint, String)</a>	Log a message to the remote machine
 	<a href="#">LogTheMessage(String, Int32, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

[Top](#)

## ▪ Examples

```
Log.LogTheMessage += (category, message) =>
W.Net.Udp.UdpLogger.LogTheMessage("127.0.0.1", 5555, true,
category, message);
```

## ▪ See Also

Reference

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpLogger Methods

## ► Methods

	Name	Description
 	<a href="#">LogTheMessage(IPEndPoint, String)</a>	Log a message to the remote machine
 	<a href="#">LogTheMessage(String, Int32, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

[Top](#)

## ► See Also

Reference

[UdpLogger Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# UdpLoggerLogTheMessage Method

## ▪ Overload List

	Name	Description
 	<a href="#">LogTheMessage(IPEndPoint, String)</a>	Log a message to the remote machine
 	<a href="#">LogTheMessage(String, Int32, String)</a>	Log a message to the remote machine. Note that this message is slower due to parsing the remoteIP with each call

[Top](#)

## ▪ See Also

Reference

[UdpUdpLogger Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# UdpUdpLoggerLogTheMessage Method (IPEndPoint, String)

Log a message to the remote machine

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public static void LogTheMessage(  
    IPEndPoint remoteIPEndPoint,  
    string message  
)
```

## Parameters

*remoteIPEndPoint*

Type: [System.NetIPEndPoint](#)

The IPEndPoint of the remote log server

*message*

Type: [SystemString](#)

The log message

## ► See Also

Reference

[UdpUdpLogger Class](#)

[LogTheMessage Overload](#)

[W.Net Namespace](#)



## Tungsten

W<sup>74</sup>

# UdpUdpLoggerLogTheMessage Method (String, Int32, String)

Log a message to the remote machine. Note that this message is slower due to parsing the `remoteIP` with each call

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in `Tungsten.Net.dll`) Version: 2.0.2

## ► Syntax

```
C# Copy
public static void LogTheMessage(
    string remoteIP,
    int remotePort,
    string message
)
```

### Parameters

*remoteIP*

Type: [System.String](#)

The IP address or name of the remote server

*remotePort*

Type: [System.Int32](#)

The port of the remote Udp server

*message*

Type: [System.String](#)

The log message

## ► See Also

Reference

UdpUdpLogger Class  
LogTheMessage Overload  
W.Net Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpUdpPeer Class

A Udp peer

## ► Inheritance Hierarchy

[SystemObject](#) [W.NetUdpUdpPeer](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class UdpPeer : IDisposable
```

The [UdpUdpPeer](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">UdpUdpPeer</a>	Constructs a new UdpServer

[Top](#)

## ► Methods

	Name	Description
≡	<a href="#">Dispose</a>	Disposes the UdpServer and release resources
≡	<a href="#">Equals</a>	Determines whether the

specified object is equal to the current object.  
(Inherited from [Object](#).)

💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">SendAsync</a>	Sends bytes to the specified remote client
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Events

	Name	Description
⚡	<a href="#">BytesReceived</a>	Raised when bytes have been received from a client

---

[Top](#)

## ◀ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor

`lock`  
(Defined by  
[MonitorExtensions](#).)

---

	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">RequestAsync</a>	Sends a request to a Udp peer and waits for a response. Can be used along with, or in lieu of, the regular method of calling Write and handling the BytesReceived event. (Defined by <a href="#">ExtensionMethods</a> .)



## Unlock

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpUdpPeer Constructor

Constructs a new UdpServer

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public UdpPeer(  
    IPEndPoint localEndPoint,  
    bool useCompression  
)
```

### Parameters

*localEndPoint*

Type: [System.NetIPEndPoint](#)

The local IPEndPoint on which to listen for data

*useCompression*

Type: [SystemBoolean](#)

If True, data is compressed before sending and decompressed when received

## ► See Also

### Reference

[UdpUdpPeer Class](#)

[W.Net Namespace](#)

## Tungsten

W<sup>74</sup>

# UdpPeer Methods

The [UdpUdpPeer](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Dispose</a>	Disposes the UdpServer and release resources
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

---

	<a href="#">SendAsync</a>	Sends bytes to the specified remote client
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">RequestAsync</a>	Sends a request to a Udp peer and waits for a response. Can be used along with, or in lieu of, the regular method of

calling Write and  
handling the  
BytesReceived event.  
(Defined by  
[ExtensionMethods](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[UdpUdpPeer Class](#)  
[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpUdpPeerDispose Method

Disposes the UdpServer and release resources

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements

[IDisposableDispose](#)

## ► See Also

Reference

[UdpUdpPeer Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpUdpPeerSendAsync Method

Sends bytes to the specified remote client

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public Task SendAsync(  
    byte[] bytes,  
    IPEndPoint remoteEndPoint  
)
```

## Parameters

*bytes*

Type: [SystemByte](#)

The data to send

*remoteEndPoint*

Type: [System.NetIPEndPoint](#)

The remote client which is listening for data

## Return Value

Type: [Task](#)

[Missing <returns> documentation for  
"M:W.Net.Udp.UdpPeer.SendAsync(System.Byte[],System.Net.IPEndPoint)"]

## ► See Also

[Reference](#)

[UdpUdpPeer Class](#)

[W.Net Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpPeer Events

The [UdpUdpPeer](#) type exposes the following members.

## Events

Name	Description
 <a href="#">BytesReceived</a>	Raised when bytes have been received from a client

[Top](#)

## See Also

Reference

[UdpUdpPeer Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# UdpUdpPeerBytesReceived Event

Raised when bytes have been received from a client

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public event Action<IPEndPoint, byte[]> BytesRece
```

Value

Type: [SystemActionIPEndPoint, Byte](#)

## ► See Also

Reference

[UdpUdpPeer Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# WakeOnLAN Class

Methods to broadcast a magic packet to wake up a machine with the given MAC address

## ► Inheritance Hierarchy

[SystemObject](#) [W.NetWakeOnLAN](#)

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class WakeOnLAN
```

The [WakeOnLAN](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">WakeOnLAN</a>	Initializes a new instance of the <a href="#">WakeOnLAN</a> class

[Top](#)

## ► Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to

		the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
💡 S	<a href="#">WakeUp(String, String, Int32)</a>	Broadcasts a magic packet to wake up the machine with the given MaC address
💡 S	<a href="#">WakeUp(String, String, String)</a>	Broadcasts a magic packet to wake up the machine with the given MAC address
💡 S	<a href="#">WakeUp_via_Socket</a>	Broadcasts a magic packet

to wake up the machine  
with the given MAC  
address

---

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnerProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

[Top](#)

## ▲ See Also

Reference

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# WakeOnLAN Constructor

Initializes a new instance of the [WakeOnLAN](#) class

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public WakeOnLAN()
```

## ► See Also

[Reference](#)

[WakeOnLAN Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# WakeOnLAN Methods

The [WakeOnLAN](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object.

(Inherited from [Object](#).)

 	<a href="#">WakeUp(String, String, Int32)</a>	Broadcasts a magic packet to wake up the machine with the given MaC address
 	<a href="#">WakeUp(String, String, String)</a>	Broadcasts a magic packet to wake up the machine with the given MAC address
 	<a href="#">WakeUp_via_Socket</a>	Broadcasts a magic packet to wake up the machine with the given MAC address

[Top](#)

## Extension Methods

Name	Description
 	<a href="#">AsTType</a> Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a> Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)

•	InLock(Action)	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	InLockTType(FuncTType)	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	InLockAsync(Action)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	InLockAsyncTType(FuncTType)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	IsDirty	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
•	Lock	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
•	MarkAsClean	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

Reference

[WakeOnLAN Class](#)  
[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# WakeOnLANWakeUp Method

## ► Overload List

	Name	Description
 	<a href="#">WakeUp(String, String, Int32)</a>	Broadcasts a magic packet to wake up the machine with the given MaC address
 	<a href="#">WakeUp(String, String, String)</a>	Broadcasts a magic packet to wake up the machine with the given MAC address

[Top](#)

## ► See Also

Reference

[WakeOnLAN Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# WakeOnLANWakeUp Method (String, String, Int32)

Broadcasts a magic packet to wake up the machine with the given MaC address

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static bool WakeUp(
    string macAddress,
    string ipBroadcastAddress = "192.168.1.255",
    int port = 9
)
```

## Parameters

*macAddress*

Type: [SystemString](#)

The MAC address of the machine to wake up

*ipBroadcastAddress* **(Optional)**

Type: [SystemString](#)

The broadcast address. This should be determined by the machine's IP address and the desired network mask.

*port* **(Optional)**

Type: [SystemInt32](#)

The socket port

**Return Value**

Type: [Boolean](#)

True if the magic packet was successfully sent, otherwise False

## ► Remarks

Adapted from: <https://www.codeproject.com/Articles/5315/Wake-On-Lan-sample-for-C>

## ► See Also

Reference

[WakeOnLAN Class](#)

[WakeUp Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# WakeOnLANWakeUp Method (String, String, String)

Broadcasts a magic packet to wake up the machine with the given MAC address

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static bool WakeUp(
    string macAddress,
    string ipAddress,
    string subnetMask
)
```

### Parameters

*macAddress*

Type: [System.String](#)

The MAC address of the machine to wake up

*ipAddress*

Type: [System.String](#)

The IP address of the machine to wake

*subnetMask*

Type: [System.String](#)

The subnet mask to determine the broadcast IP address

### Return Value

Type: [Boolean](#)

True if the magic packet was successfully sent, otherwise False

## ► Remarks

Taken from:

<https://blogs.msdn.microsoft.com/knom/2008/12/31/wake-on-lan-client-with-c/>

## ► See Also

Reference

[WakeOnLAN Class](#)

[WakeUp Overload](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# WakeOnLANWakeUp\_via\_Socket Method

Broadcasts a magic packet to wake up the machine with the given MAC address

**Namespace:** [W.Net](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public static bool WakeUp_via_Socket(
    string macAddress,
    string broadcastAddress = "192.168.1.255"
    int port = 9
)
```

## Parameters

*macAddress*

Type: [SystemString](#)

The MAC address of the machine to wake up

*broadcastAddress (Optional)*

Type: [SystemString](#)

The port on which to send the magic packet (typically 0, 7 or 9)

*port (Optional)*

Type: [SystemInt32](#)

[Missing <param name="port"/> documentation for

"M:W.Net.WakeOnLAN.WakeUp\_via\_Socket(System.String,System.String,System

## Return Value

Type: [Boolean](#)

True if the magic packet was successfully sent, otherwise False

## ► Remarks

Adapted from: <https://www.codeproject.com/Articles/5315/Wake-On-Lan-sample-for-C>

## ► See Also

Reference

[WakeOnLAN Class](#)

[W.Net Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# W.Net.RPC Namespace

[Missing <summary> documentation for "N:W.Net.RPC"]

## ► Classes

Class	Description
 <a href="#">Client</a>	Make calls into a Tungsten.Net.RPC.Server over Tcp
 <a href="#">MethodDictionary</a>	Used to store and call RPC methods on a Tungsten.Net.RPC Server
 <a href="#">RPCClassAttribute</a>	Add this attribute to a class if it contains static methods with the RPCMethod attribute
 <a href="#">RPCMethodAttribute</a>	Add this attribute to a static method and Tungsten.Net.RPC.Server can automatically add the method to it's dictionary of callable methods.
 <a href="#">RPCResponse</a>	Encapsulates information related to making the RPC call and the return value
 <a href="#">RPCResponseTResponseType</a>	Encapsulates information related to making the

RPC call and the return value



## Server

Allows remote instances of Tungsten.Net.RPC.Client to call local methods.

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Client Class

Make calls into a Tungsten.Net.RPC.Server over Tcp

## ► Inheritance Hierarchy

[SystemObject](#) [W.Net.RPCCClient](#)

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class Client : IDisposable
```

The [Client](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">Client(Int32)</a>	Constructs a new Client
≡	<a href="#">Client(IPEndPoint, Int32, Int32)</a>	Constructs a new Client, initialized with the specified values

[Top](#)

## ► Properties

	Name	Description
≡		

---

	<code>CallTimeout</code>	The maximum amount of time, in milliseconds, that a call should wait for a response
	<code>RemoteEndPoint</code>	The IPPEndPoint of the server (The server must be a valid instance of W.Net.RPC.Server)

---

[Top](#)

## ◀ Methods

Name	Description
 <a href="#">Call(String, Object)</a>	Calls a method on the server
 <a href="#">CallTResponseType(String, Object)</a>	Calls a method on the server
 <a href="#">CallAsync(String, Object)</a>	Calls a method on the server
 <a href="#">CallAsyncTResponseType(String, Object)</a>	Calls a method on the server
 <a href="#">Dispose</a>	Disposes the Client and release resources
 <a href="#">Equals</a>	Determines whether the specified

object is equal to the current object.  
(Inherited from [Object](#).)

💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)



## ToString

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

# Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ► Remarks

Assymetric encryption is used to discourage sniffing.

The server must be a valid instance of W.Net.RPC.Server

## ► See Also

[Reference](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Client Constructor

## ► Overload List

	Name	Description
≡	<a href="#">Client(Int32)</a>	Constructs a new Client
≡	<a href="#">Client(IPEndPoint, Int32, Int32)</a>	Constructs a new Client, initialized with the specified values

[Top](#)

## ► See Also

[Reference](#)

[Client Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Client Constructor (Int32)

Constructs a new Client

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public Client(  
    int encryptionKeySize  
)
```

### Parameters

*encryptionKeySize*

Type: [System.Int32](#)

The encryption key size (typically 2048 or 4096; 384 to 16384 in increments of 8)

## ► See Also

[Reference](#)

[Client Class](#)

[Client Overload](#)

[W.Net.RPC Namespace](#)

## Tungsten

W<sup>74</sup>

# Client Constructor (IPEndPoint, Int32, Int32)

Constructs a new Client, initialized with the specified values

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public Client(  
    IPEndPoint remoteEndPoint,  
    int encryptionKeySize,  
    int msCallTimeout = -1  
)
```

## Parameters

*remoteEndPoint*

Type: [System.Net.IPEndPoint](#)

The server's IP address and port

*encryptionKeySize*

Type: [System.Int32](#)

The encryption key size (typically 2048 or 4096; 384 to 16384 in increments of 8)

*msCallTimeout (Optional)*

Type: [System.Int32](#)

The maximum number of milliseconds to wait for a call to complete

## ► See Also

[Reference](#)

[Client Class](#)

[Client Overload](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Client Properties

The [Client](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">CallTimeout</a>	The maximum amount of time, in milliseconds, that a call should wait for a response
	<a href="#">RemoteEndPoint</a>	The IPEndPoint of the server (The server must be a valid instance of W.Net.RPC.Server)

[Top](#)

## See Also

[Reference](#)

[Client Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ClientCallTimeout Property

The maximum amount of time, in milliseconds, that a call should wait for a response

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public int CallTimeout { get; set; }
```

Property Value

Type: [Int32](#)

## ► See Also

[Reference](#)

[Client Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ClientRemoteEndPoint Property

The IPPEndPoint of the server (The server must be a valid instance of W.Net.RPC.Server)

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public IPPEndPoint RemoteEndPoint { get; set; }
```

Property Value

Type: [IPPEndPoint](#)

## ► See Also

[Reference](#)

[Client Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Client Methods

The [Client](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Call(String, Object)</a>	Calls a method on the server
<a href="#">CallTResponseType(String, Object)</a>	Calls a method on the server
<a href="#">CallAsync(String, Object)</a>	Calls a method on the server
<a href="#">CallAsyncTResponseType(String, Object)</a>	Calls a method on the server
<a href="#">Dispose</a>	Disposes the Client and release resources
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object.

(Inherited from  
[Object](#).)



### [Finalize](#)

Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.  
(Inherited from [Object](#).)



### [GetHashCode](#)

Serves as the default hash function.  
(Inherited from [Object](#).)



### [GetType](#)

Gets the [Type](#) of the current instance.  
(Inherited from [Object](#).)



### [MemberwiseClone](#)

Creates a shallow copy of the current [Object](#).  
(Inherited from [Object](#).)



### [ToString](#)

Returns a string that

represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

---

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## ▲ See Also

[Reference](#)

[Client Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ClientCall Method

## ► Overload List

	Name	Description
≡	<a href="#">Call(String, Object)</a>	Calls a method on the server
≡	<a href="#">CallITResponseType(String, Object)</a>	Calls a method on the server

[Top](#)

## ► See Also

[Reference](#)

[Client Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ClientCall Method (String, Object)

Calls a method on the server

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public RPCResponse Call(  
    string methodName,  
    params Object[] args  
)
```

## Parameters

*methodName*

Type: [SystemString](#)

The name of the method to call

*args*

Type: [SystemObject](#)

Any arguments to pass into the method

## Return Value

Type: [RPCResponse](#)

RPCResponse containing information related to the call and the return value

## ► See Also

[Reference](#)

Client Class  
Call Overload  
W.Net.RPC Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ClientCallTResponseType

## Method (String, Object)

Calls a method on the server

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

### ► Syntax

```
C# Copy
public RPCResponse<TResponseType> Call<TResponse>
    string methodName,
    params Object[] args
)
```

### Parameters

*methodName*

Type: [System.String](#)

The name of the method to call

*args*

Type: [System.Object](#)

Any arguments to pass into the method

### Type Parameters

*TResponseType*

[Missing <typeparam name="TResponseType"/> documentation for  
"M:W.Net.RPC.Client.Call`1(System.String,System.Object[])"]

### Return Value

Type: [RPCResponse<TResponseType>](#)

RPCResponse containing information related to the call and the return value

## ▲ See Also

[Reference](#)

[Client Class](#)

[Call Overload](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ClientCallAsync Method

## ► Overload List

	Name	Description
≡	<a href="#">CallAsync(String, Object)</a>	Calls a method on the server
≡	<a href="#">CallAsyncTResponseType(String, Object)</a>	Calls a method on the server

[Top](#)

## ► See Also

[Reference](#)

[Client Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ClientCallAsync Method (String, Object)

Calls a method on the server

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public Task<RPCResponse> CallAsync(  
    string methodName,  
    params Object[] args  
)
```

## Parameters

*methodName*

Type: [SystemString](#)

The name of the method to call

*args*

Type: [SystemObject](#)

Any arguments to pass into the method

## Return Value

Type: [TaskRPCResponse](#)

RPCResponse containing information related to the call and the return value

## ► See Also

[Reference](#)

Client Class  
CallAsync Overload  
W.Net.RPC Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ClientCallAsync<TResponseType> Method (String, Object)

Calls a method on the server

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public Task<RPCResponse<TResponseType>> CallAsync(
    string methodName,
    params Object[] args
)
```

## Parameters

*methodName*

Type: [System.String](#)

The name of the method to call

*args*

Type: [System.Object](#)

Any arguments to pass into the method

## Type Parameters

*TResponseType*

[Missing <typeparam name="TResponseType"/> documentation for  
"M:W.Net.RPC.Client.CallAsync`1(System.String,System.Object[])"]

## Return Value

Type: [Task<RPCResponse<TResponseType>>](#)

RPCResponse containing information related to the call and the return value

## ▲ See Also

[Reference](#)

[Client Class](#)

[CallAsync Overload](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ClientDispose Method

Disposes the Client and release resources

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements

[IDisposableDispose](#)

## ► See Also

[Reference](#)

[Client Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MethodDictionary Class

Used to store and call RPC methods on a Tungsten.Net.RPC Server

## ► Inheritance Hierarchy

[SystemObject](#) [W.Net.RPCMethodDictionary](#)

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class MethodDictionary
```

The [MethodDictionary](#) type exposes the following members.

## ► Constructors

	Name	Description
	<a href="#">MethodDictionary</a>	Initializes a new instance of the <a href="#">MethodDictionary</a> class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Methods</a>	All of the RPC methods found. The name of each method is the complete namespace.class.methodname

hierarchy (recursive classes are allowed).

[Top](#)

## ◀ Methods

	Name	Description
≡	<a href="#">Call(Object, Exception, String, Object)</a>	Call a method on the Tungsten.Net.RPC Server. This method s
≡	<a href="#">CallITResult(String, Object)</a>	Call a method on the Tungsten.Net.RPC Server.
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> .

(Inherited from [Object](#).)

 <a href="#">Refresh</a>	Scans the server process for RPC methods (static methods with the <code>RPCMethod</code> attribute in classes with the <code>RPCClass</code> attribute)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ◀ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's <code>Owner</code> property to "owner". This method should be called in the constructor of any class which has <code>IOwnedProperty</code> members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

• [InLockTType\(FuncTType\)](#)  
Overloaded.  
Performs the function in  
a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

• [InLockAsync\(Action\)](#)  
Overloaded.  
Asynchronously performs  
the action in a Monitor  
lock  
(Defined by  
[MonitorExtensions](#).)

• [InLockAsyncTType\(FuncTType\)](#)  
Overloaded.  
Asynchronously performs  
the action in a Monitor  
lock  
(Defined by  
[MonitorExtensions](#).)

• [IsDirty](#)  
Scans the IsDirty value of  
each field and property of  
type IProperty  
(Defined by  
[PropertyHostExtensions](#).)

• [Lock](#)  
Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)

• [MarkAsClean](#)  
Scans each field and  
property of type IProperty  
and sets its IsDirty flag to  
false  
(Defined by  
[PropertyHostExtensions](#).)



## Unlock

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MethodDictionary Constructor

Initializes a new instance of the [MethodDictionary](#) class

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public MethodDictionary()
```

## ► See Also

[Reference](#)

[MethodDictionary Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MethodDictionary Properties

The [MethodDictionary](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Methods</a>	All of the RPC methods found. The name of each method is the complete namespace.class.methodname hierarchy (recursive classes are allowed).

[Top](#)

## ► See Also

Reference

[MethodDictionary Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MethodDictionaryMethods Property

All of the RPC methods found. The name of each method is the complete namespace.class.methodname hierarchy (recursive classes are allowed).

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public Dictionary<string, MethodInfo> Methods { }
```

Property Value

Type: [DictionaryString, MethodInfo](#)

## ► See Also

Reference

[MethodDictionary Class](#)

[W.Net.RPC Namespace](#)

## Tungsten

W<sup>74</sup>

# MethodDictionary Methods

The [MethodDictionary](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Call(Object, Exception, String, Object)</a>	Call a method on the Tungsten.Net.RPC Server. This method s
≡	<a href="#">CallITResult(String, Object)</a>	Call a method on the Tungsten.Net.RPC Server.
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)



[MemberwiseClone](#) Creates a shallow copy of the current [Object](#).  
(Inherited from [Object](#).)



[Refresh](#) Scans the server process for RPC methods (static methods with the [RPCMethod](#) attribute in classes with the [RPCClass](#) attribute)



[ToString](#) Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## ▪ Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's <a href="#">Owner</a> property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded.

Performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockTType\(FuncTType\)](#)  
Overloaded.  
Performs the function in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockAsync\(Action\)](#)  
Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockAsyncTType\(FuncTType\)](#)  
Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [IsDirty](#)  
Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---

⬇️ [Lock](#)  
Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [MarkAsClean](#)  
Scans each field and property of type IProperty and sets its IsDirty flag to false

(Defined by  
[PropertyHostExtensions](#).)



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[MethodDictionary Class](#)  
[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MethodDictionaryCall Method

## ▪ Overload List

	Name	Description
≡	<a href="#">CallTResult(String, Object)</a>	Call a method on the Tungsten.Net.RPC Server.
≡	<a href="#">Call(Object, Exception, String, Object)</a>	Call a method on the Tungsten.Net.RPC Server. This method s

[Top](#)

## ▪ See Also

[Reference](#)

[MethodDictionary Class](#)  
[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MethodDictionaryCallTResult

## Method (String, Object)

Call a method on the Tungsten.Net.RPC Server.

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

### ► Syntax

C#

[Copy](#)

```
public TResult Call<TResult>(  
    string method,  
    params Object[] args  
)
```

### Parameters

*method*

Type: [SystemString](#)

The namespace, class name and method name of the method to call (ie: MyNamespace.MyClass.Method1)

*args*

Type: [SystemObject](#)

Arguments, if any, to be passed into the remote method

### Type Parameters

*TResult*

The expected return type of the call

### Return Value

Type: [TResult](#)

A result of type TResult

## ▪ Remarks

If TResult does not match the return type of the method on the server, a return value cannot be expected and the call may time out.

## ▪ See Also

Reference

[MethodDictionary Class](#)

[Call Overload](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MethodDictionaryCall Method (Object, Exception, String, Object)

Call a method on the Tungsten.Net.RPC Server. This method s

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public bool Call(  
    out Object result,  
    out Exception exception,  
    string method,  
    params Object[] args  
)
```

## Parameters

*result*

Type: [SystemObject](#)

The value returned from the called method

*exception*

Type: [SystemException](#)

The exception if one occurred

*method*

Type: [SystemString](#)

The namespace, class name and method name of the method to call (ie: MyNamespace.MyClass.Method1)

*args*

Type: [SystemObject](#)

Arguments, if any, to be passed into the remote method

## Return Value

Type: [Boolean](#)

A CallResult object describing the result of the call. If the remote method does not have a return value, the value of CallResult.Result will be null.

## See Also

[Reference](#)

[MethodDictionary Class](#)

[Call Overload](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MethodDictionaryRefresh Method

Scans the server process for RPC methods (static methods with the `RPCMethod` attribute in classes with the `RPCClass` attribute)

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in `Tungsten.Net.dll`) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public void Refresh(  
    Assembly rootAssembly,  
    bool recurse  
)
```

### Parameters

*rootAssembly*

Type: [System.ReflectionAssembly](#)

The Assembly to scan for RPC methods

*recurse*

Type: [SystemBoolean](#)

If True, referenced assemblies will also be scanned

## ► Remarks

Any methods previously added manually will have to be re-added

## ► See Also

[Reference](#)

MethodDictionary Class  
W.Net.RPC Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCClassAttribute Class

Add this attribute to a class if it contains static methods with the RPCMethod attribute

## ► Inheritance Hierarchy

[SystemObject](#) [SystemAttribute](#)  
[W.Net.RPCRPCClassAttribute](#)

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public class RPCClassAttribute : Attribute
```

The [RPCClassAttribute](#) type exposes the following members.

## ► Constructors

	Name	Description
	<a href="#">RPCClassAttribute</a>	Initializes a new instance of the <a href="#">RPCClassAttribute</a> class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">TypeId</a>	When implemented in a derived class,

gets a unique identifier for this [Attribute](#).  
(Inherited from [Attribute](#).)

[Top](#)

## ◀ Methods

	Name	Description
≡	<a href="#">Equals</a>	Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <a href="#">Attribute</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Returns the hash code for this instance. (Inherited from <a href="#">Attribute</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">IsDefaultAttribute</a>	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from <a href="#">Attribute</a> .)
≡	<a href="#">Match</a>	When overridden in a derived class, returns a value that

---

		indicates whether this instance equals a specified object. (Inherited from <a href="#">Attribute</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## ◀ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock

(Defined by  
[MonitorExtensions](#).)



### [InLockTType\(FuncTType\)](#)

Overloaded.  
Performs the function in  
a Monitor lock  
(Defined by  
[MonitorExtensions](#).)



### [InLockAsync\(Action\)](#)

Overloaded.  
Asynchronously performs  
the action in a Monitor  
lock  
(Defined by  
[MonitorExtensions](#).)



### [InLockAsyncTType\(FuncTType\)](#)

Overloaded.  
Asynchronously performs  
the action in a Monitor  
lock  
(Defined by  
[MonitorExtensions](#).)



### [IsDirty](#)

Scans the IsDirty value of  
each field and property of  
type IProperty  
(Defined by  
[PropertyHostExtensions](#).)



### [Lock](#)

Performs a Monitor lock  
(Defined by  
[MonitorExtensions](#).)



### [MarkAsClean](#)

Scans each field and  
property of type IProperty  
and sets its IsDirty flag to  
false  
(Defined by  
[PropertyHostExtensions](#).)



## Unlock

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCClassAttribute Constructor

Initializes a new instance of the [RPCClassAttribute](#) class

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public RPCClassAttribute()
```

## ► See Also

[Reference](#)

[RPCClassAttribute Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCClassAttribute Properties

The [RPCClassAttribute](#) type exposes the following members.

## Properties

Name	Description
 <a href="#">TypeId</a>	When implemented in a derived class, gets a unique identifier for this <a href="#">Attribute</a> . (Inherited from <a href="#">Attribute</a> .)

[Top](#)

## See Also

Reference

[RPCClassAttribute Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCClassAttribute Methods

The [RPCClassAttribute](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Equals</a>	Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <a href="#">Attribute</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Returns the hash code for this instance. (Inherited from <a href="#">Attribute</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
<a href="#">IsDefaultAttribute</a>	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from <a href="#">Attribute</a> .)

	<a href="#">Match</a>	When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from <a href="#">Attribute</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
 	<a href="#">AsTType</a> Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a> Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a> Overloaded.

Performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockTType\(FuncTType\)](#)  
Overloaded.  
Performs the function in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockAsync\(Action\)](#)  
Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockAsyncTType\(FuncTType\)](#)  
Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [IsDirty](#)  
Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---

⬇️ [Lock](#)  
Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [MarkAsClean](#)  
Scans each field and property of type IProperty and sets its IsDirty flag to false

(Defined by  
[PropertyHostExtensions](#).)



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[RPCClassAttribute Class](#)  
[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCMethodAttribute Class

Add this attribute to a static method and Tungsten.Net.RPC.Server can automatically add the method to it's dictionary of callable methods.

## ► Inheritance Hierarchy

[SystemObject](#) [SystemAttribute](#)  
[W.Net.RPCRPCMethodAttribute](#)

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public class RPCMethodAttribute : Attribute
```

The [RPCMethodAttribute](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">RPCMethodAttribute</a>	Initializes a new instance of the <a href="#">RPCMethodAttribute</a> class

[Top](#)

## ► Properties

	Name	Description
≡		

**TypeId** When implemented in a derived class, gets a unique identifier for this [Attribute](#).  
(Inherited from [Attribute](#).)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Equals</a>	Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Returns the hash code for this instance. (Inherited from <a href="#">Attribute</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">IsDefaultAttribute</a>	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Match</a>	When overridden in a derived

class, returns a value that indicates whether this instance equals a specified object.  
(Inherited from [Attribute](#).)



#### [MemberwiseClone](#)

Creates a shallow copy of the current [Object](#).  
(Inherited from [Object](#).)



#### [ToString](#)

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## Extension Methods

Name	Description
<a href="#">AsTTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a

Monitor lock  
(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by

[PropertyHostExtensions. \)](#)



## [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions.](#))

[Top](#)

## ► Remarks

Note: Due to the way Newtonsoft.Json deserializes integers, do NOT use int (Int32) in your RPC methods as parameters or return types; use longs instead.

## ► See Also

[Reference](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCMethodAttribute Constructor

Initializes a new instance of the [RPCMethodAttribute](#) class

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public RPCMethodAttribute()
```

## ► See Also

[Reference](#)

[RPCMethodAttribute Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCMethodAttribute Properties

The [RPCMethodAttribute](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">TypeId</a>	When implemented in a derived class, gets a unique identifier for this <a href="#">Attribute</a> . (Inherited from <a href="#">Attribute</a> .)

[Top](#)

## ► See Also

Reference

[RPCMethodAttribute Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCMethodAttribute Methods

The [RPCMethodAttribute](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Equals</a>	Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <a href="#">Attribute</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Returns the hash code for this instance. (Inherited from <a href="#">Attribute</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">IsDefaultAttribute</a>	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from <a href="#">Attribute</a> .)

	<a href="#">Match</a>	When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from <a href="#">Attribute</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ▪ Extension Methods

Name	Description
 	<a href="#">AsTType</a> Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a> Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a> Overloaded.

Performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockTType\(FuncTType\)](#)  
Overloaded.  
Performs the function in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockAsync\(Action\)](#)  
Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockAsyncTType\(FuncTType\)](#)  
Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [IsDirty](#)  
Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---

⬇️ [Lock](#)  
Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [MarkAsClean](#)  
Scans each field and property of type IProperty and sets its IsDirty flag to false

(Defined by  
[PropertyHostExtensions](#).)



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[RPCMethodAttribute Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponse Class

Encapsulates information related to making the RPC call and the return value

## ► Inheritance Hierarchy

[SystemObject](#) [W.Net.RPCRPCResponse](#)  
[W.Net.RPCRPCResponseTResponseType](#)

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public class RPCResponse
```

The [RPCResponse](#) type exposes the following members.

## ► Constructors

	Name	Description
	<a href="#">RPCResponse</a>	Initializes a new instance of the <a href="#">RPCResponse</a> class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Exception</a>	May contain exception information if

there was an exception making or as a result of the call

	<a href="#">Method</a>	The name of the method called
	<a href="#">Response</a>	The return value from the method
	<a href="#">Success</a>	True if the call was successful, otherwise False

[Top](#)

## ◀ Methods

Name	Description
	<a href="#">Equals</a> Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a> Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a> Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a> Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a> Creates a shallow copy of the current <a href="#">Object</a> .

(Inherited from [Object](#).)



### [ToString](#)

Useful for debugging or displaying information quickly.  
(Overrides [Object.ToString](#).)

[Top](#)

## ▪ Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

Reference

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponse Constructor

Initializes a new instance of the [RPCResponse](#) class

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public RPCResponse()
```

## ► See Also

[Reference](#)

[RPCResponse Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponse Properties

The [RPCResponse](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">Exception</a>	May contain exception information if there was an exception making or as a result of the call
	<a href="#">Method</a>	The name of the method called
	<a href="#">Response</a>	The return value from the method
	<a href="#">Success</a>	True if the call was successful, otherwise False

[Top](#)

## See Also

[Reference](#)

[RPCResponse Class](#)

[W.Net.RPC Namespace](#)

## Tungsten

W<sup>74</sup>

# RPCResponseException Property

May contain exception information if there was an exception making or as a result of the call

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public string Exception { get; set; }
```

Property Value

Type: [String](#)

## ► See Also

Reference

[RPCResponse Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponseMethod Property

The name of the method called

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public string Method { get; set; }
```

Property Value

Type: [String](#)

## ► See Also

Reference

[RPCResponse Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponseResponse Property

The return value from the method

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

```
public Object Response { get; set; }
```

[Copy](#)

Property Value

Type: [Object](#)

## ► See Also

Reference

[RPCResponse Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponseSuccess Property

True if the call was successful, otherwise False

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

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

Property Value

Type: [Boolean](#)

## ► Remarks

Note that this is different than the return value from the method, which can be of any value or type

## ► See Also

Reference

[RPCResponse Class](#)

[W.Net.RPC Namespace](#)

## Tungsten

W<sup>74</sup>

# RPCResponse Methods

The [RPCResponse](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Useful for debugging or displaying information quickly. (Overrides <a href="#">Object.ToString</a> .)

---

[Top](#)

## ◀ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor

`lock`  
(Defined by  
[MonitorExtensions](#).)

---

	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## See Also

Reference  
[RPCResponse Class](#)

## [W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponseToString Method

Useful for debugging or displaying information quickly.

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public override string ToString()
```

Return Value

Type: [String](#)

Returns a string representation of class members and their values

## ► See Also

[Reference](#)

[RPCResponse Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponse<TResponseType> Class

Encapsulates information related to making the RPC call and the return value

## ► Inheritance Hierarchy

[SystemObject](#) [W.Net.RPCRPCResponse](#)  
[W.Net.RPCRPCResponseTResponseType](#)

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C# Copy
public class RPCResponse<TResponseType> : RPCRes
```

### Type Parameters

*TResponseType*

The Type expected as a return value from the method call

The [RPCResponseTResponseType](#) type exposes the following members.

## ► Constructors

Name	Description
<a href="#">RPCResponseTResponseType</a>	Initializes a new instance of the <a href="#">RPCResponseTResponseType</a> class.

class

---

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Exception</a>	May contain exception information if there was an exception making or as a result of the call (Inherited from <a href="#">RPCResponse</a> .)
	<a href="#">Method</a>	The name of the method called (Inherited from <a href="#">RPCResponse</a> .)
	<a href="#">Response</a>	The return value from the method
	<a href="#">Success</a>	True if the call was successful, otherwise False (Inherited from <a href="#">RPCResponse</a> .)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.

(Inherited from [Object](#).)

 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
 <a href="#">ToString</a>	Useful for debugging or displaying information quickly. (Inherited from <a href="#">RPCResponse</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members

(Defined by  
[PropertyHostExtensions](#).)

---

✿ [InLock\(Action\)](#) Overloaded.  
Performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

✿ [InLockTType\(FuncTType\)](#) Overloaded.  
Performs the function in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

✿ [InLockAsync\(Action\)](#) Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

✿ [InLockAsyncTType\(FuncTType\)](#) Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

✿ [IsDirty](#) Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---

✿ [Lock](#) Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)



## [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets it's `IsDirty` flag to `false`

(Defined by [PropertyHostExtensions.](#))

---



## [Unlock](#)

Performs a Monitor unlock

(Defined by [MonitorExtensions.](#))

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponseTResponseType Constructor

Initializes a new instance of the [RPCResponseTResponseType](#) class

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

```
public RPCResponse()
```

[Copy](#)

## ► See Also

[Reference](#)

[RPCResponseTResponseType Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponseTResponseType Properties

The [RPCResponseTResponseType](#) generic type exposes the following members.

## Properties

	Name	Description
	<a href="#">Exception</a>	May contain exception information if there was an exception making or as a result of the call (Inherited from <a href="#">RPCResponse</a> .)
	<a href="#">Method</a>	The name of the method called (Inherited from <a href="#">RPCResponse</a> .)
	<a href="#">Response</a>	The return value from the method
	<a href="#">Success</a>	True if the call was successful, otherwise False (Inherited from <a href="#">RPCResponse</a> .)

[Top](#)

## See Also

Reference

[RPCResponseTResponseType Class](#)  
[W.Net.RPC Namespace](#)

## Tungsten

W<sup>74</sup>

# RPCResponseTResponseTypeResponse Property

The return value from the method

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public TResponseType Response { get; set; }
```

Property Value

Type: [TResponseType](#)

## ► See Also

Reference

[RPCResponseTResponseType Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# RPCResponseTResponseType Methods

The [RPCResponseTResponseType](#) generic type exposes the following members.

## ▲ Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

<a href="#">ToString</a>	Useful for debugging or displaying information quickly. (Inherited from <a href="#">RPCResponse</a> .)
--------------------------	---

[Top](#)

## ◀ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

---

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## ▲ See Also

Reference

[RPCResponseType Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Server Class

Allows remote instances of Tungsten.Net.RPC.Client to call local methods.

## ► Inheritance Hierarchy

[SystemObject](#) [W.Net.RPCServer](#)

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public class Server : IDisposable
```

The [Server](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">Server</a>	Initializes the Tungsten.Net.RPC.Server and loads the RPC methods

[Top](#)

## ► Properties

	Name	Description
≡	<a href="#">API</a>	Exposes the dictionary of methods. Custom, non-attributed methods may be

added to this dictionary.

[Top](#)

## Methods

	Name	Description
≡	<a href="#">Dispose</a>	Disposes the Tungsten.Net.RPC.Server and releases resources
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Calls Dispose and deconstructs the Tungsten.Net.RPC.Server (Overrides <a href="#">ObjectFinalize</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Start</a>	Starts listening for client connections on the specified network interface and port

---

	<a href="#">Stop</a>	Stops listening for client connections
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## ► Extension Methods

Name	Description
≡ <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
≡ <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
≡ <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
≡ <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ► Remarks

Note: Due to the way Newtonsoft.Json deserializes integers, do NOT use int (Int32) in your api's as parameters or return types; use longs instead.

## ► See Also

[Reference](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Server Constructor

Initializes the Tungsten.Net.RPC.Server and loads the RPC methods

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

[Copy](#)

```
public Server(  
    int encryptionKeySize  
)
```

### Parameters

*encryptionKeySize*

Type: [System.Int32](#)

The encryption key size (typically 2048 or 4096; 384 to 16384 in increments of 8)

## ► Remarks

The client must be declared with the same value.

## ► See Also

[Reference](#)

[Server Class](#)

[W.Net.RPC Namespace](#)

## Tungsten

W<sup>74</sup>

# Server Properties

The [Server](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">API</a>	Exposes the dictionary of methods. Custom, non-attributed methods may be added to this dictionary.

[Top](#)

## ► See Also

[Reference](#)

[Server Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ServerAPI Property

Exposes the dictionary of methods. Custom, non-attributed methods may be added to this dictionary.

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public MethodDictionary API { get; }
```

Property Value

Type: [MethodDictionary](#)

## ► See Also

[Reference](#)

[Server Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Server Methods

The [Server](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Dispose</a>	Disposes the Tungsten.Net.RPC.Server and releases resources
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Finalize</a>	Calls <a href="#">Dispose</a> and deconstructs the Tungsten.Net.RPC.Server (Overrides <a href="#">ObjectFinalize</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Start</a>	Starts listening for client

---

		connections on the specified network interface and port
	<a href="#">Stop</a>	Stops listening for client connections
	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## ◀ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

---

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ◀ See Also

[Reference](#)

[Server Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ServerDispose Method

Disposes the Tungsten.Net.RPC.Server and releases resources

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements

[IDisposableDispose](#)

## ► See Also

[Reference](#)

[Server Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ServerFinalize Method

Calls Dispose and deconstructs the Tungsten.Net.RPC.Server

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
protected override void Finalize()
```

Implements

[ObjectFinalize](#)

## ► See Also

[Reference](#)

[Server Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ServerStart Method

Starts listening for client connections on the specified network interface and port

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

```
C#  
public void Start(  
    IPEndPoint ep,  
    Assembly rpcAssembly,  
    bool scanReferences = false  
)
```

[Copy](#)

## Parameters

*ep*

Type: [System.NetIPEndPoint](#)

The IP Endpoint on which to bind and listen for clients

*rpcAssembly*

Type: [System.ReflectionAssembly](#)

The root assembly in which to scan for RPC methods

*scanReferences (Optional)*

Type: [SystemBoolean](#)

If True, referenced assemblies will also be scanned for RPC methods

## ► See Also

[Reference](#)

[Server Class](#)

## [W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ServerStop Method

Stops listening for client connections

**Namespace:** [W.Net.RPC](#)

**Assembly:** Tungsten.Net (in Tungsten.Net.dll) Version: 2.0.2

## ► Syntax

C#

Copy

```
public void Stop()
```

## ► See Also

[Reference](#)

[Server Class](#)

[W.Net.RPC Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# W.Threading Namespace

[Missing <summary> documentation for "N:W.Threading"]

## ► Classes

	Class	Description
	<a href="#">Thread</a>	Functionality related to multi-threading
	<a href="#">ThreadMethod</a>	Task-based multi-threading

## ► Delegates

	Delegate	Description
	<a href="#">ThreadMethod</a> <a href="#">ThreadMethodDelegate</a>	Delegate type used by ThreadMethod

## ► Enumerations

	Enumeration	Description
	<a href="#">CPUProfileEnum</a>	The preferred level of CPU usage

## Tungsten

W<sup>74</sup>

# CPUProfileEnum Enumeration

The preferred level of CPU usage

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public enum CPUProfileEnum
```

## ► Members

Member name	Value	Description
SpinWait0	0	High CPU usage, but fastest execution. May be faster on single-core/cpu machines. May be slower on multi-core/cpu machines.
Sleep	1	Medium CPU usage. Uses Thread.Sleep or Task.Delay to block the current thread.
SpinWait1	2	Low CPU usage. Should be faster on multi-core/cpu machines as the load will be divided among cores/cpus. Slowest on single-core/cpu machines.

---

Yield	3	Only available for .Net Framework; uses Thread.Yield instead of Thread.Sleep.
SpinUntil	4	

---

## ▲ See Also

Reference

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Thread Class

Functionality related to multi-threading

## ► Inheritance Hierarchy

[SystemObject](#) [W.ThreadingThread](#)

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public class Thread
```

The [Thread](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">Thread</a>	Initializes a new instance of the <a href="#">Thread</a> class

[Top](#)

## ► Methods

	Name	Description
≡	<a href="#">Equals</a>	Determines whether the specified object is equal

		to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡 ⚡	<a href="#">Sleep(Int32)</a>	Blocks the calling thread for the specified time
💡 ⚡	<a href="#">Sleep(CPUProfileEnum)</a>	Attempts to free the CPU for other processes, based on the desired level. Consequences will vary depending on your hardware architecture. The more processors/cores you have, the better performance you will

have by selecting LowCPU. Likewise, on a single-core processor, you may wish to select HighCPU.

---

≡  	<a href="#">Sleep(Int32, Boolean)</a>	Blocks the calling thread for the specified time
≡  	<a href="#">Sleep(CPUProfileEnum, Int32)</a>	Attempts to free the CPU for other processes, based on the desired level. Consequences will vary depending on your hardware architecture. The more processors/cores you have, the better performance you will have by selecting SpinWait1. Likewise, on a single-core processor, you may wish to select SpinWait0.
≡ 	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## ► Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded.

Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---



### [IsDirty](#)

Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---



### [Lock](#)

Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)

---



### [MarkAsClean](#)

Scans each field and property of type IProperty and sets its IsDirty flag to false  
(Defined by [PropertyHostExtensions](#).)

---



### [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

---

[Top](#)

## ► See Also

Reference

[W.Threading Namespace](#)

## Tungsten

W<sup>74</sup>

# Thread Constructor

Initializes a new instance of the [Thread](#) class

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public Thread()
```

## ► See Also

[Reference](#)

[Thread Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Thread Methods

The [Thread](#) type exposes the following members.

## ▪ Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
S	<a href="#">Sleep(Int32)</a>	Blocks the calling thread for the specified

time

---

 	<a href="#">Sleep(CPUProfileEnum)</a>	Attempts to free the CPU for other processes, based on the desired level. Consequences will vary depending on your hardware architecture. The more processors/cores you have, the better performance you will have by selecting LowCPU. Likewise, on a single-core processor, you may wish to select HighCPU.
 	<a href="#">Sleep(Int32, Boolean)</a>	Blocks the calling thread for the specified time
 	<a href="#">Sleep(CPUProfileEnum, Int32)</a>	Attempts to free the CPU for other processes, based on the desired level. Consequences will vary depending on your hardware architecture. The more processors/cores you have, the better performance you will have by selecting SpinWait1. Likewise, on a single-core processor,

you may wish to select SpinWait0.



### [ToString](#)

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## ► Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in

		a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ◀ See Also

**Reference**

[Thread Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadSleep Method

## ▪ Overload List

Name	Description
  <a href="#">Sleep(Int32)</a>	Blocks the calling thread for the specified time
  <a href="#">Sleep(CPUProfileEnum)</a>	Attempts to free the CPU for other processes, based on the desired level. Consequences will vary depending on your hardware architecture. The more processors/cores you have, the better performance you will have by selecting LowCPU. Likewise, on a single-core processor, you may wish to select HighCPU.
  <a href="#">Sleep(Int32, Boolean)</a>	Blocks the calling thread for the specified time
  <a href="#">Sleep(CPUProfileEnum, Int32)</a>	Attempts to free the CPU for other processes, based on the desired level.

Consequences will vary depending on your hardware architecture. The more processors/cores you have, the better performance you will have by selecting SpinWait1. Likewise, on a single-core processor, you may wish to select SpinWait0.

---

[Top](#)

## ▲ See Also

[Reference](#)

[Thread Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Thread.Sleep Method (Int32)

Blocks the calling thread for the specified time

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static void Sleep(  
    int msDelay  
)
```

[Copy](#)

## Parameters

*msDelay*

Type: [System.Int32](#)

The number of milliseconds to block the thread

## ► See Also

[Reference](#)

[Thread Class](#)

[Sleep Overload](#)

[W.Threading Namespace](#)

## Tungsten

W<sup>74</sup>

# ThreadSleep Method (CPUProfileEnum)

Attempts to free the CPU for other processes, based on the desired level. Consequences will vary depending on your hardware architecture. The more processors/cores you have, the better performance you will have by selecting LowCPU. Likewise, on a single-core processor, you may wish to select HighCPU.

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static void Sleep(  
    CPUProfileEnum level  
)
```

### Parameters

*level*

Type: [W.ThreadingCPUProfileEnum](#)  
The desired level of CPU usage

## ► Remarks

Note results may vary. LowCPU will spread the load onto multiple cores and can actually yield faster results depending on your hardware architecture. This may not always be the case.

## ► See Also

[Reference](#)

[Thread Class](#)

[Sleep Overload](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Thread.Sleep Method (Int32, Boolean)

Blocks the calling thread for the specified time

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static void Sleep(  
    int msDelay,  
    bool useSpinWait  
)
```

[Copy](#)

## Parameters

*msDelay*

Type: [SystemInt32](#)

The number of milliseconds to block the thread

*useSpinWait*

Type: [SystemBoolean](#)

If True, a SpinWait.SpinUntil will be used instead of a call to Thread.Sleep (or Task.Delay). Note that SpinWait should only be used on multi-core/cpu machines.

## ► See Also

[Reference](#)

[Thread Class](#)

[Sleep Overload](#)

## [W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadSleep Method (CPUProfileEnum, Int32)

Attempts to free the CPU for other processes, based on the desired level. Consequences will vary depending on your hardware architecture. The more processors/cores you have, the better performance you will have by selecting SpinWait1. Likewise, on a single-core processor, you may wish to select SpinWait0.

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static void Sleep(  
    CPUProfileEnum level,  
    int msTimeout = 1  
)
```

## Parameters

*level*

Type: [W.ThreadingCPUProfileEnum](#)

The desired level of CPU usage

*msTimeout (Optional)*

Type: [SystemInt32](#)

Optional value for CPUProfileEnum.Sleep and CPUProfileEnum.SpinUntil. Ignored by other profiles.

## ► Remarks

Note results may vary. SpinWait1 will spread the load onto multiple cores and can actually yield faster results depending on your hardware architecture. This may not always be the case.

## ▲ See Also

Reference

[Thread Class](#)

[Sleep Overload](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethod Class

Task-based multi-threading

## ► Inheritance Hierarchy

[SystemObject](#) [W.Threading.ThreadMethod](#)

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public class ThreadMethod : IDisposable
```

The ThreadMethod type exposes the following members.

## ► Constructors

Name	Description
<a href="#">ThreadMethod(ActionCancellationToken)</a>	Creates a new instance of the ThreadMethod class.
<a href="#">ThreadMethod(ThreadMethodThreadMethodDelegate)</a>	Creates a new instance of the ThreadMethod class.

[Top](#)

## ► Properties

	Name	Description
	<a href="#">IsComplete</a>	True if the thread has completed, otherwise False
	<a href="#">Name</a>	A user-defined name for this object

[Top](#)

## ▲ Methods

	Name	Description
	<a href="#">Cancel</a>	Signals the method via Cancellation to stop run and waits for complete
 	<a href="#">Create(ActionCancellationToken)</a>	Constructs ThreadMethod
 	<a href="#">Create(ThreadMethodThreadMethodDelegate)</a>	Constructs ThreadMethod
	<a href="#">Dispose</a>	Disposes the ThreadMethod and releases resources
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object (Inherited from <a href="#">Object</a> .)

---

💡	<a href="#">Finalize</a>	Allows an object to try to free resources before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">GetType</a>	Gets the Type of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a copy of the current Object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Start</a>	Starts the timer.
💡	<a href="#">Start(Object)</a>	Starts the timer with the specified interval.
💡	<a href="#">ToString</a>	Returns a string that represents the current object.

(Inherited from [Object](#).)



## Wait

Waits a specified number of milliseconds for the thread to complete.

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded.

		Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 ↴	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 ↴	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 ↴	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 ↴	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 ↴	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 ↴	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by

[MonitorExtensions.\)](#)

---

[Top](#)

## ► See Also

Reference

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethod Constructor

## ► Overload List

Name	Description
 <a href="#">ThreadMethod(ActionCancellationToken)</a>	Creates a new instance of the ThreadMethod class.
 <a href="#">ThreadMethod(ThreadMethodThreadMethodDelegate)</a>	Creates a new instance of the ThreadMethod class.

[Top](#)

## ► See Also

Reference

[ThreadMethod Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethod Constructor (Action< CancellationToken >)

Constructs a new ThreadMethod

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public ThreadMethod(
    Action< CancellationToken > threadProc
)
```

## Parameters

*threadProc*

Type: [SystemAction< CancellationToken >](#)

The thread proc

## ► See Also

### Reference

[ThreadMethod Class](#)

[ThreadMethod Overload](#)

[W.Threading Namespace](#)

## Tungsten

W<sup>74</sup>

# ThreadMethod Constructor (ThreadMethodThreadMethodDeleg

Constructs a new ThreadMethod

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public ThreadMethod(
    ThreadMethodThreadMethodDelegate threadProc
)
```

### Parameters

*threadProc*

Type: [W.ThreadingThreadMethodThreadMethodDelegate](#)  
The thread proc

## ► See Also

Reference

[ThreadMethod Class](#)

[ThreadMethod Overload](#)

[W.Threading Namespace](#)

## Tungsten

W<sup>74</sup>

# ThreadMethod Properties

The [ThreadMethod](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">IsComplete</a>	True if the thread has completed, otherwise False
	<a href="#">Name</a>	A user-defined name for this object

[Top](#)

## ► See Also

Reference

[ThreadMethod Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethodIsComplete Property

True if the thread has completed, otherwise False

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public bool IsComplete { get; }
```

Property Value

Type: [Boolean](#)

## ► See Also

Reference

[ThreadMethod Class](#)

[W.Threading Namespace](#)

## Tungsten

W<sup>74</sup>

# ThreadMethodName Property

A user-defined name for this object

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public string Name { get; set; }
```

Property Value

Type: [String](#)

## ► See Also

Reference

[ThreadMethod Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethod Methods

The [ThreadMethod](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Cancel</a>	Signals the method via Cancellation to stop running and waits for complete
<a href="#">Create(ActionCancellationToken)</a>	Constructs ThreadMethod
<a href="#">Create(ThreadMethodThreadMethodDelegate)</a>	Constructs ThreadMethod
<a href="#">Dispose</a>	Disposes the ThreadMethod and releases resources
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to

to try to free resources perform other cleanup operations it is reclaim garbage collection.  
(Inherited from [Object](#).)

---

≡	<a href="#">GetHashCode</a>	Serves as default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the Type of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">MemberwiseClone</a>	Creates a copy of the current Object. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">Start</a>	Starts the timer.
≡	<a href="#">Start(Object)</a>	Starts the timer with the specified object.
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)



## Wait

Waits a specified number of milliseconds until the thread completes.

[Top](#)

## Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ◀ See Also

Reference

[ThreadMethod Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethodCancel Method

Signals the thread method via the CancellationToken to stop running and waits for it to complete

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

```
public void Cancel()
```

[Copy](#)

## ► See Also

Reference

[ThreadMethod Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethodCreate Method

## ► Overload List

Name	Description
  <a href="#">Create(ActionCancellationToken)</a>	Constructs new ThreadMethod using an Action<T> cancellation token.
  <a href="#">Create(ThreadMethodThreadMethodDelegate)</a>	Constructs new ThreadMethod using a ThreadMethodThreadMethodDelegate.

[Top](#)

## ► See Also

Reference

[ThreadMethod Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethodCreate Method (Action< CancellationToken >)

Constructs a new ThreadMethod

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static ThreadMethod Create(
    Action<CancellationToken> threadProc
)
```

## Parameters

*threadProc*

Type: [SystemAction< CancellationToken >](#)

The thread proc

## Return Value

Type: [ThreadMethod](#)

[Missing <returns> documentation for  
"M:W.Threading.ThreadMethod.Create(System.Action{System.Threading.CancellationT

## ► See Also

Reference

[ThreadMethod Class](#)

[Create Overload](#)

[W.Threading Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethodCreate Method (ThreadMethodThreadMethodDeleg

Constructs a new ThreadMethod

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static ThreadMethod Create(
    ThreadMethodThreadMethodDelegate threadP
)
◀ ▶
```

### Parameters

*threadProc*

Type: [W.ThreadingThreadMethodThreadMethodDelegate](#)  
The thread proc

### Return Value

Type: [ThreadMethod](#)

[Missing <returns> documentation for  
"M:W.Threading.ThreadMethod.Create(W.Threading.ThreadMethod.ThreadMethodDele

## ► See Also

Reference

[ThreadMethod Class](#)

Create Overload

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethodDispose Method

Disposes the ThreadMethod and releases resources

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements

[IDisposableDispose](#)

## ► See Also

[Reference](#)

[ThreadMethod Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethodStart Method

## ► Overload List

	Name	Description
≡	<a href="#">Start</a>	Starts the thread
≡	<a href="#">Start(Object)</a>	Starts the thread

[Top](#)

## ► See Also

[Reference](#)

[ThreadMethod Class](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethodStart Method

Starts the thread

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public void Start()
```

## ► See Also

Reference

[ThreadMethod Class](#)

[Start Overload](#)

[W.Threading Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ThreadMethodStart Method (Object)

Starts the thread

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

```
C#  
public void Start(  
    params Object[] args  
)
```

[Copy](#)

## Parameters

*args*

Type: [System.Object](#)

The arguments to pass into the thread procedure

## ► See Also

Reference

[ThreadMethod Class](#)

[Start Overload](#)

[W.Threading Namespace](#)

## Tungsten

W<sup>74</sup>

# ThreadMethodWait Method

Waits a specified number of milliseconds for the thread to complete

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public bool Wait(  
    int msTimeout = -1  
)
```

### Parameters

#### *msTimeout (Optional)*

Type: [SystemInt32](#)

The number of milliseconds to wait for the thread to complete. A value of -1 indicates an infinite wait period.

### Return Value

Type: [Boolean](#)

True if the thread completes within the timeout period, otherwise

False

## ► See Also

[Reference](#)

[ThreadMethod Class](#)

[W.Threading Namespace](#)



## Tungsten

W<sup>74</sup>

# ThreadMethodDelegate

Delegate type used by ThreadMethod

**Namespace:** [W.Threading](#)

**Assembly:** Tungsten.Threading (in Tungsten.Threading.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public delegate void ThreadMethodDelegate(
    CancellationToken token,
    params Object[] args
)
```

### Parameters

*token*

Type: [System.Threading.CancellationToken](#)

A CancellationToken which can be used to signal the threaded method to stop

*args*

Type: [System.Object](#)

Zero or more arguments to pass into the thread method

## ► See Also

Reference

[W.Threading Namespace](#)



## Tungsten

W<sup>74</sup>

# W.Threading.Lockers Namespace

[Missing <summary> documentation for "N:W.Threading.Lockers"]

## ► Classes

Class	Description
 <a href="#">Disposer</a>	Aids in implementing a clean Dispose method. Supports re-entrancy but only calls the cleanup Action once.
 <a href="#">MonitorExtensions</a>	Extensions to object to simplify locking with Monitor
 <a href="#">MonitorLocker</a>	Uses Monitor to provide resource locking
 <a href="#">MonitorLockerTState</a>	Extends MonitorLocker with an internal state variable
 <a href="#">ReaderWriterLocker</a>	Uses ReaderWriterLockSlim to provide resource locking
 <a href="#">ReaderWriterLockerTState</a>	Extends ReaderWriterLocker with an internal state

		variable
	<a href="#">ReaderWriterLockSlimExtensions</a>	Extensions to simplify locking with ReaderWriterLockSlim
	<a href="#">SemaphoreSlimExtensions</a>	Extensions to simplify locking with SemaphoreSlim
	<a href="#">SemaphoreSlimLocker</a>	Uses SemaphoreSlim to provide resource locking
	<a href="#">SemaphoreSlimLockerTState</a>	Extends SemaphoreSlimLocker with an internal state variable
	<a href="#">SpinLocker</a>	Uses SpinLock to provide resource locking
	<a href="#">SpinLockerTState</a>	Extends SpinLocker with an internal state variable
	<a href="#">SpinLockExtensions</a>	Extensions to simplify locking with SpinLock
	<a href="#">StateLockerTLocker, TState</a>	Extends a locker (SpinLocker, MonitorLocker, ReaderWriterLocker, SemaphoreSlimLocker) with an internal state value

## ▪ Interfaces

Interface	Description
<a href="#">ILocker</a>	The required implementation for a locking object
<a href="#">ILockerTLocker</a>	The required implementation for a locking object
<a href="#">IReaderWriterStateLockerTState</a>	Interface definition for a ReaderWriterLocker with a State variable
<a href="#">IStateLockerTLocker, TState</a>	The required implementation for a stateful locking object

## ▪ Delegates

Delegate	Description
 <a href="#">StateAssignmentDelegateTState</a>	Delegate which can be used to assign a new value to the internal state

## ▪ Enumerations

Enumeration	Description
-------------	-------------



[LockTypeEnum](#) Used by ReaderWriterLocker to specify the type of lock to obtain

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Disposer Class

Aids in implementing a clean Dispose method. Supports re-entrancy but only calls the cleanup Action once.

## ▪ Inheritance Hierarchy

[SystemObject](#) [W.Threading.LockersDisposer](#)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ▪ Syntax

C#

Copy

```
public class Disposer
```

The [Disposer](#) type exposes the following members.

## ▪ Constructors

	Name	Description
	<a href="#">Disposer</a>	Initializes a new instance of the <a href="#">Disposer</a> class

[Top](#)

## ▪ Properties

	Name	Description
	<a href="#">IsDisposed</a>	True if Cleanup has been called and

completed, otherwise False



[IsDisposing](#) True if the Disposer is in the process of disposing, otherwise False

[Top](#)

## ◀ Methods

	Name	Description
≡	<a href="#">Dispose(Action)</a>	Calls the action (should contain cleanup code)
≡	<a href="#">Dispose(Object, Action)</a>	Calls the action (should contain cleanup code)
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the

current [Object](#).  
(Inherited from [Object](#).)



### [ToString](#)

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## ► Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ◀ See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Disposer Constructor

Initializes a new instance of the [Disposer](#) class

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public Disposer()
```

## ► See Also

[Reference](#)

[Disposer Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Disposer Properties

The [Disposer](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">IsDisposed</a>	True if Cleanup has been called and completed, otherwise False
	<a href="#">IsDisposing</a>	True if the Disposer is in the process of disposing, otherwise False

[Top](#)

## ► See Also

[Reference](#)

[Disposer Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DisposerIsDisposed Property

True if Cleanup has been called and completed, otherwise False

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public bool IsDisposed { get; }
```

Property Value

Type: [Boolean](#)

## ► See Also

[Reference](#)

[Disposer Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DisposerIsDisposing Property

True if the Disposer is in the process of disposing, otherwise False

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public bool IsDisposing { get; }
```

Property Value

Type: [Boolean](#)

## ► See Also

[Reference](#)

[Disposer Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Disposer Methods

The [Disposer](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Dispose(Action)</a>	Calls the action (should contain cleanup code)
≡	<a href="#">Dispose(Object, Action)</a>	Calls the action (should contain cleanup code)
≡	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the

current [Object](#).  
(Inherited from [Object](#).)



### [ToString](#)

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## ► Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ► See Also

[Reference](#)

[Disposer Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# DisposerDispose Method

## ► Overload List

	Name	Description
≡	<a href="#">Dispose(Action)</a>	Calls the action (should contain cleanup code)
≡	<a href="#">Dispose(Object, Action)</a>	Calls the action (should contain cleanup code)

[Top](#)

## ► See Also

[Reference](#)

[Disposer Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# Disposer.Dispose Method (Action)

Calls the action (should contain cleanup code)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public void Dispose(  
    Action cleanupAction  
)
```

[Copy](#)

### Parameters

*cleanupAction*

Type: [SystemAction](#)

The action to call

## ► See Also

Reference

[Disposer Class](#)

[Dispose Overload](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# Disposer.Dispose Method (Object, Action)

Calls the action (should contain cleanup code)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public void Dispose(  
    Object objToSuppressFinalize,  
    Action cleanupAction  
)
```

### Parameters

*objToSuppressFinalize*

Type: [System.Object](#)

The object on which to suppress the finalizer call (usually the one currently being disposed)

*cleanupAction*

Type: [System.Action](#)

The action to call

## ► See Also

Reference

[Disposer Class](#)

[Dispose Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILocker Interface

The required implementation for a locking object

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public interface ILocker
```

## ► Methods

Name	Description
≡• InLock(Action)	Perform some action in a lock
≡• InLockTResult(FuncTResult)	Perform some function in a lock
≡• InLockAsync(Action)	
≡• InLockAsyncTResult(FuncTResult)	Asynchrонously perform some function in a lock

[Top](#)

## ► See Also

## Reference

### [W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILocker Methods

## ► Methods

Name	Description
 <a href="#">InLock(Action)</a>	Perform some action in a lock
 <a href="#">InLockTResult(FuncTResult)</a>	Perform some function in a lock
 <a href="#">InLockAsync(Action)</a>	
 <a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchronosuly perform some function in a lock

[Top](#)

## ► See Also

[Reference](#)

[ILocker Interface](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ILockerInLock Method

## ► Overload List

Name	Description
 <a href="#">InLock(Action)</a>	Perform some action in a lock
 <a href="#">InLockTResult(FuncTResult)</a>	Perform some function in a lock

[Top](#)

## ► See Also

[Reference](#)

[ILocker Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILockerInLock Method (Action)

Perform some action in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
void InLock(  
    Action action  
)
```

[Copy](#)

### Parameters

*action*

Type: [SystemAction](#)

The action to perform

## ► See Also

Reference

[ILocker Interface](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ILockerInLockTResult Method (FuncTResult)

Perform some function in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
TResult InLock<TResult>(  
    Func<TResult> func  
)
```

[Copy](#)

## Parameters

*func*

Type: [SystemFuncTResult](#)

The function to perform

## Type Parameters

*TResult*

The result Type

## Return Value

Type: [TResult](#)

The result of the function

## ► See Also

Reference

[ILocker Interface](#)  
[InLock Overload](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILockerInLockAsync Method

## ▪ Overload List

Name	Description
 <a href="#">InLockAsync(Action)</a>	
 <a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchrhonously perform some function in a lock

[Top](#)

## ▪ See Also

Reference

[ILocker Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILockerInLockAsync Method (Action)

[Missing <summary> documentation for "M:W.Threading.Lockers.ILocker.InLockAsync(System.Action)"]

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
Task InLockAsync(  
    Action action  
)
```

[Copy](#)

## Parameters

*action*

Type: [SystemAction](#)

[Missing <param name="action"/> documentation for "M:W.Threading.Lockers.ILocker.InLockAsync(System.Action)"]

## Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:W.Threading.Lockers.ILocker.InLockAsync(System.Action)"]

## ► See Also

[Reference](#)

[ILocker Interface](#)

InLockAsync Overload  
W.Threading.Lockers Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILockerInLockAsyncResult Method (Func<TResult>)

Asynchronous perform some function in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
Task<TResult> InLockAsync<TResult>(  
    Func<TResult> func  
)
```

### Parameters

*func*

Type: [System.Func<TResult>](#)

The function to perform

### Type Parameters

*TResult*

The result Type

### Return Value

Type: [Task<TResult>](#)

The result of the function

## ► See Also

Reference

[ILocker Interface](#)  
[InLockAsync Overload](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILocker<TLocker> Interface

The required implementation for a locking object

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public interface ILocker<TLocker> : ILocker
```

### Type Parameters

*TLocker*

The type of locker to use (SpinLock, Monitor, SemaphoreSlim,  
ReaderWriterLock)

The [ILocker<TLocker>](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The object used for locking

[Top](#)

## ► Methods

	Name	Description
	<a href="#">InLock(Action)</a>	Perform some

action in a lock  
(Inherited from  
[ILocker](#).)



### [InLockTResult\(FuncTResult\)](#)

Perform some  
function in a  
lock  
(Inherited from  
[ILocker](#).)



### [InLockAsync\(Action\)](#)

(Inherited from  
[ILocker](#).)



### [InLockAsyncTResult\(FuncTResult\)](#)

Asynchronosly  
perform some  
function in a  
lock  
(Inherited from  
[ILocker](#).)

[Top](#)

## ▲ See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILockerTLocker Properties

The [ILockerTLocker](#) generic type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The object used for locking

[Top](#)

## ► See Also

[Reference](#)

[ILockerTLocker Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILockerTLockerLocker Property

The object used for locking

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
TLocker Locker { get; }
```

Property Value

Type: [TLocker](#)

## ► See Also

Reference

[ILockerTLocker Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ILockerTLocker Methods

## ► Methods

Name	Description
<a href="#">InLock(Action)</a>	Perform some action in a lock (Inherited from <a href="#">ILocker</a> .)
<a href="#">InLockTResult(FuncTResult)</a>	Perform some function in a lock (Inherited from <a href="#">ILocker</a> .)
<a href="#">InLockAsync(Action)</a>	(Inherited from <a href="#">ILocker</a> .)
<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asyncrenously perform some function in a lock (Inherited from <a href="#">ILocker</a> .)

[Top](#)

## ► See Also

Reference

[ILockerTLocker Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLocker<TState> Interface

Interface definition for a ReaderWriterLocker with a State variable

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public interface IReaderWriterStateLocker<TState>  
{  
}
```

[Copy](#)

### Type Parameters

*TState*

The state Type

The [IReaderWriterStateLocker<TState>](#) type exposes the following members.

## ► Methods

Name	Description
<a href="#">GetState</a>	Retrieves the internal state from within a ReaderWriterLockSli
<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSli

≡	InLockTValue(LockTypeEnum, FuncTState, TValue)	Executes a function from within a ReaderWriterLockSli
≡	InLockAsync(LockTypeEnum, ActionTState)	Executes an action from within a ReaderWriterLockSli
≡	InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)	Executes a function from within a ReaderWriterLockSli
≡	SetState	Sets the internal stat from within a ReaderWriterLockSli

[Top](#)

## ▲ See Also

[Reference](#)

[W.Threading.Lockers Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLockerTState Methods

The [IReaderWriterStateLockerTState](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">GetState</a>	Retrieves the internal state from within a ReaderWriterLockSli
<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSli
<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSli
<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSli
<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSli
<a href="#">SetState</a>	Sets the internal state from within a ReaderWriterLockSli

[Top](#)

## ◀ See Also

Reference

[IReaderWriterStateLockerTState Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLocker{TState} Get Method

Retrieves the internal state from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
TState GetState()
```

**Return Value**

Type: [\*TState\*](#)

The current state

## ► See Also

[Reference](#)

[IReaderWriterStateLocker{TState} Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLockerTStateInLock Method

## ▪ Overload List

Name	Description
 <a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSlim
 <a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSlim

[Top](#)

## ▪ See Also

### Reference

[IReaderWriterStateLockerTState Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLocker{TState} InLock Method (LockTypeEnum, Action{TState})

Executes an action from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
void InLock(  
    LockTypeEnum lockType,  
    Action<TState> action  
)
```

### Parameters

*lockType*

Type: [W.Threading.Lockers.LockTypeEnum](#)

Specifies whether to use a Read or Write lock

*action*

Type: [SystemAction{TState}](#)

The action to run

## ► See Also

[Reference](#)

[IReaderWriterStateLocker{TState} Interface](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLocker{TState}InLock<TValue> Method (LockTypeEnum, Func<TState, TValue>) (Tungsten.Threading.Lockers)

Executes a function from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
Copy  
TValue InLock< TValue >(  
    LockTypeEnum lockType,  
    Func< TState, TValue > func  
)
```

### Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

Specifies whether to use a Read or Write lock

*func*

Type: [SystemFunc<TState, TValue>](#)

The function to run

### Type Parameters

*TValue*

The type of return value

### Return Value

Type: *TValue*

The result of the function call (a value of type TValue)

## ▲ See Also

### Reference

[IReaderWriterStateLockerTState Interface](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLockerTStateInLockMethod

## ▪ Overload List

Name	Description
 <a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSli
 <a href="#">InLockAsync TValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSli

[Top](#)

## ▪ See Also

### Reference

[IReaderWriterStateLockerTState Interface](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLocker`1.InLockAsync Method (LockTypeEnum, Action<TState>)

Executes an action from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
Task InLockAsync(  
    LockTypeEnum lockType,  
    Action<TState> action  
)
```

### Parameters

*lockType*

Type: [W.Threading.Lockers.LockTypeEnum](#)

Specifies whether to use a Read or Write lock

*action*

Type: [SystemAction<TState>](#)

The action to run

### Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:W.Threading.Lockers.IReaderWriterStateLocker`1.InLockAsync(W.Threading.Locke

## ► See Also

## Reference

[IReaderWriterStateLockerTState Interface](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLocker{TState} InLockAsync Method (LockTypeEnum, Func{TState, TValue})

Executes a function from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
Task<TValue> InLockAsync<TValue>(
    LockTypeEnum lockType,
    Func<TState, TValue> func
)
```

### Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

Specifies whether to use a Read or Write lock

*func*

Type: [SystemFunc{TState, TValue}](#)

The function to run

### Type Parameters

*TValue*

The type of return value

### Return Value

Type: [Task{TValue}](#)

The result of the function call (a value of type TValue)

## ▲ See Also

### Reference

[IReaderWriterStateLockerTState Interface](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IReaderWriterStateLocker{TState} SetMethod

Sets the internal state from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

```
void SetState(  
    TState newState  
)
```

[Copy](#)

### Parameters

*newState*

Type: [TState](#)

The new value

## ► See Also

### Reference

[IReaderWriterStateLocker{TState} Interface](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# IStateLocker<TLocker, TState> Interface

The required implementation for a stateful locking object

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public interface IStateLocker<TLocker, TState> :  
    ILocker
```

### Type Parameters

#### *TLocker*

[Missing <typeparam name="TLocker"/> documentation for  
"T:W.Threading.Lockers.IStateLocker`2"]

#### *TState*

[Missing <typeparam name="TState"/> documentation for  
"T:W.Threading.Lockers.IStateLocker`2"]

The **IStateLocker<TLocker, TState>** type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The object used for locking (Inherited from <a href="#">ILocker&lt;TLocker&gt;</a> .)

[Top](#)

## Methods

Name	Description
<a href="#">InLock(Action)</a>	Perform some action in a lock (Inherited from <a href="#">ILocker</a> .)
<a href="#">InLock(ActionTState)</a>	Perform some action in a lock
<a href="#">InLock(FuncTState, TState)</a>	Perform some function in a lock
<a href="#">InLockTResult(FuncTResult)</a>	Perform some function in a lock (Inherited from <a href="#">ILocker</a> .)
<a href="#">InLockAsync(Action)</a>	(Inherited from <a href="#">ILocker</a> .)
<a href="#">InLockAsync(ActionTState)</a>	
<a href="#">InLockAsync(FuncTState, TState)</a>	Asncrhonously perform some function in a lock
<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asncrhonously perform some function in a lock

(Inherited from  
[ILocker](#).)

---

[Top](#)

## ◀ See Also

[Reference](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IStateLockerTLocker, TState Properties

The [IStateLockerTLocker, TState](#) generic type exposes the following members.

## Properties

	Name	Description
	<a href="#">Locker</a>	The object used for locking (Inherited from <a href="#">ILockerTLocker</a> .)

[Top](#)

## See Also

### Reference

[IStateLockerTLocker, TState Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IStateLocker<TLocker, TState> Methods

## ◀ Methods

Name	Description
<a href="#">InLock(Action)</a>	Perform some action in a lock (Inherited from <a href="#">ILocker</a> .)
<a href="#">InLock(Action&lt;TState&gt;)</a>	Perform some action in a lock
<a href="#">InLock(Func&lt;TState, TState&gt;)</a>	Perform some function in a lock
<a href="#">InLockTResult(Func&lt;TResult&gt;)</a>	Perform some function in a lock (Inherited from <a href="#">ILocker</a> .)
<a href="#">InLockAsync(Action)</a>	(Inherited from <a href="#">ILocker</a> .)
<a href="#">InLockAsync(Action&lt;TState&gt;)</a>	
<a href="#">InLockAsync(Func&lt;TState, TState&gt;)</a>	Asynchronously perform some function in a lock



## InLockAsyncTResult(FuncTResult)

Asynchrонously perform some function in a lock  
(Inherited from [ILocker](#).)

---

[Top](#)

## ▲ See Also

### Reference

[IStateLockerTLocker, TState Interface](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IStateLockerTLocker, TStateInLock Method

## ► Overload List

Name	Description
 <a href="#">InLock(Action)</a>	Perform some action in a lock (Inherited from <a href="#">ILocker</a> .)
 <a href="#">InLockTResult(FuncTResult)</a>	Perform some function in a lock (Inherited from <a href="#">ILocker</a> .)
 <a href="#">InLock(ActionTState)</a>	Perform some action in a lock
 <a href="#">InLock(FuncTState, TState)</a>	Perform some function in a lock

[Top](#)

## ► See Also

### Reference

[IStateLockerTLocker, TState Interface](#)  
[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# IStateLockerTLocker, TStateInLock Method (Action<TState>)

Perform some action in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
void InLock(  
            Action<TState> action  
)
```

[Copy](#)

### Parameters

*action*

Type: [SystemAction<TState>](#)

The action to perform

## ► See Also

### Reference

[IStateLockerTLocker, TState Interface](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# IStateLockerTLocker, TStateInLock Method (Func<TState, TState>)

Perform some function in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
TState InLock(  
    Func<TState, TState> func  
)
```

[Copy](#)

### Parameters

*func*

Type: [System.Func<TState, TState>](#)

The function to perform

### Return Value

Type: [TState](#)

The result of the function

## ► See Also

Reference

[IStateLockerTLocker, TState Interface](#)

[InLock Overload](#)

## [W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IStateLockerTLocker, TStateInLockAsync Method

## ▪ Overload List

Name	Description
<a href="#">InLockAsync(Action)</a>	(Inherited from <a href="#">ILocker</a> .)
<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchrhonously perform some function in a lock (Inherited from <a href="#">ILocker</a> .)
<a href="#">InLockAsync(ActionTState)</a>	
<a href="#">InLockAsync(FuncTState, TState)</a>	Asynchrонously perform some function in a lock

[Top](#)

## ▪ See Also

[Reference](#)

[IStateLockerTLocker, TState Interface](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# IStateLocker{TLocker}, TStateInLockAsync Method (Action{TState})

[Missing <summary> documentation for  
"M:W.Threading.Lockers.IStateLocker`2.InLockAsync(System.Action{`1})"]

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
Task InLockAsync(  
    Action<TState> action  
)
```

### Parameters

*action*

Type: [SystemAction{TState}](#)

[Missing <param name="action"/> documentation for  
"M:W.Threading.Lockers.IStateLocker`2.InLockAsync(System.Action{`1})"]

### Return Value

Type: [Task](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.IStateLocker`2.InLockAsync(System.Action{`1})"]

## ► See Also

## Reference

[IStateLockerTLocker, TState Interface](#)  
[InLockAsync Overload](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# IStateLockerTLocker, TStateInLockAsync Method (Func<TState, TState>)

Asynchronous perform some function in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
Task<TState> InLockAsync(  
    Func<TState, TState> func  
)
```

## Parameters

*func*

Type: [System.Func<TState, TState>](#)

The function to perform

## Return Value

Type: [Task<TState>](#)

The result of the function

## ► See Also

### Reference

[IStateLockerTLocker, TState Interface](#)

[InLockAsync Overload](#)

## [W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# LockTypeEnum Enumeration

Used by ReaderWriterLocker to specify the type of lock to obtain

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public enum LockTypeEnum
```

## ► Members

Member name	Value	Description
Read	0	Obtain a read-lock
Write	1	Obtain a write-lock

## ► See Also

Reference

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# MonitorExtensions Class

Extensions to object to simplify locking with Monitor

## ► Inheritance Hierarchy

SystemObject W.Threading.LockersMonitorExtensions

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static class MonitorExtensions
```

The MonitorExtensions type exposes the following members.

## ► Methods

	Name	Description
≡ S	<a href="#">InLock(Object, Action)</a>	Performs the action in a Monitor lock
≡ S	<a href="#">InLockTType(Object, FuncTType)</a>	Performs the function in a Monitor lock
≡ S	<a href="#">InLockAsync(Object, Action)</a>	Asynchronously performs the action in a Monitor lock
≡ S	<a href="#">InLockAsyncTType(Object, FuncTType)</a>	Asynchronously performs the action

in a Monitor lock

---

  <b>Lock</b>	Performs a Monitor lock
  <b>Unlock</b>	Performs a Monitor unlock

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorExtensions Methods

The [MonitorExtensions](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">InLock(Object, Action)</a>	Performs the action in a Monitor lock
 	<a href="#">InLockTType(Object, FuncTType)</a>	Performs the function in a Monitor lock
 	<a href="#">InLockAsync(Object, Action)</a>	Asynchronously performs the action in a Monitor lock
 	<a href="#">InLockAsyncTType(Object, FuncTType)</a>	Asynchronously performs the action in a Monitor lock
 	<a href="#">Lock</a>	Performs a Monitor lock
 	<a href="#">Unlock</a>	Performs a Monitor unlock

[Top](#)

## ▪ See Also

[Reference](#)

[MonitorExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorExtensionsInLock Method

## ► Overload List

	Name	Description
 	<a href="#">InLock(Object, Action)</a>	Performs the action in a Monitor lock
 	<a href="#">InLockTType(Object, FuncTType)</a>	Performs the function in a Monitor lock

[Top](#)

## ► See Also

Reference

[MonitorExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorExtensionsInLock Method (Object, Action)

Performs the action in a Monitor lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static void InLock(  
    this Object this,  
    Action action  
)
```

[Copy](#)

### Parameters

*this*

Type: [SystemObject](#)

The object to provide resource locking

*action*

Type: [SystemAction](#)

The action to perform

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ▲ See Also

Reference

[MonitorExtensions Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorExtensionsInLock<TType> Method (Object, Func<TType>)

Performs the function in a Monitor lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
  
public static TType InLock<TType>(  
    this Object this,  
    Func<TType> func  
)
```

[Copy](#)

## Parameters

*this*

Type: [SystemObject](#)

The object to provide resource locking

*func*

Type: [SystemFunc<TType>](#)

The function to perform

## Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for  
"M:W.Threading.Lockers.MonitorExtensions.InLock`1(System.Object,System.Fur

## Return Value

Type: *TType*

[Missing <returns> documentation for  
"M:W.Threading.Lockers.MonitorExtensions.InLock`1(System.Object,System.Func{`0})

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[MonitorExtensions Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorExtensionsInLockAsync Method

## ▪ Overload List

	Name	Description
 S	<a href="#">InLockAsync(Object, Action)</a>	Asynchronously performs the action in a Monitor lock
 S	<a href="#">InLockAsyncTType(Object, FuncTType)</a>	Asynchronously performs the action in a Monitor lock

[Top](#)

## ▪ See Also

Reference

[MonitorExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorExtensionsInLockAsync Method (Object, Action)

Asynchronously performs the action in a Monitor lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static Task InLockAsync(  
    this Object this,  
    Action action  
)
```

[Copy](#)

## Parameters

*this*

Type: [SystemObject](#)

The object to provide resource locking

*action*

Type: [SystemAction](#)

The action to perform

## Return Value

Type: [Task](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.MonitorExtensions.InLockAsync(System.Object,System.Action)"

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance

method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[MonitorExtensions Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorExtensionsInLockAsync<TType> Method (Object, Func<TType>)

Asynchronously performs the action in a Monitor lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static Task<TType> InLockAsync<TType>(
    this Object this,
    Func<TType> func
)
```

## Parameters

*this*

Type: [System.Object](#)

The object to provide resource locking

*func*

Type: [System.Func<TType>](#)

The function to perform

## Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for  
"M:W.Threading.Lockers.MonitorExtensions.InLockAsync`1(System.Object,System.Func`1(TType))"]

## Return Value

Type: [Task<TType>](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.MonitorExtensions.InLockAsync`1(System.Object,System.Fu

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[MonitorExtensions Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorExtensionsLock Method

Performs a Monitor lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static void Lock(  
    this Object this,  
    Action action  
)
```

## Parameters

*this*

Type: [SystemObject](#)

The object to provide resource locking

*action*

Type: [SystemAction](#)

[Missing <param name="action"/> documentation for

"M:W.Threading.Lockers.MonitorExtensions.Lock(System.Object,System.Action)"

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ► See Also

## Reference

[MonitorExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorExtensionsUnlock Method

Performs a Monitor unlock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static void Unlock(  
    this Object this,  
    Action action  
)
```

[Copy](#)

### Parameters

*this*

Type: [SystemObject](#)

The object to provide resource locking

*action*

Type: [SystemAction](#)

[Missing <param name="action"/> documentation for

"M:W.Threading.Lockers.MonitorExtensions.Unlock(System.Object,System.Action)"

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Object](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ▲ See Also

Reference

[MonitorExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLocker Class

Uses Monitor to provide resource locking

## ► Inheritance Hierarchy

[SystemObject](#) [W.Threading.LockersMonitorLocker](#)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public class MonitorLocker : ILocker<Object>,  
    ILocker
```

The [MonitorLocker](#) type exposes the following members.

## ► Constructors

	Name	Description
	<a href="#">MonitorLocker</a>	Initializes a new instance of the MonitorLocker class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The object used to perform locks

[Top](#)

## ◀ Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 GetHashCode	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 GetType	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)

[Object.](#))

≡	<a href="#">InLock(Action)</a>	Executes an action from within a Monitor
≡	<a href="#">InLock TValue(FuncTValue)</a>	Executes a function from within a Monitor
≡	<a href="#">InLockAsync(Action)</a>	Executes an action from within a Monitor
≡	<a href="#">InLockAsync TValue(FuncTValue)</a>	Executes a function from within a Monitor
≡	<a href="#">Lock</a>	Locks the resource
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)



[Unlock](#)

Unlock the resource

[Top](#)

## Extension Methods

	Name	Description
	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

[Top](#)

## ► Remarks

Can be overridden to provide additional functionality

## ► See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLocker Constructor

Initializes a new instance of the [MonitorLocker](#) class

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public MonitorLocker()
```

## ► See Also

Reference

[MonitorLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLocker Properties

The [MonitorLocker](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The object used to perform locks

[Top](#)

## ► See Also

[Reference](#)

[MonitorLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerLocker Property

The object used to perform locks

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

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

Property Value

Type: [Object](#)

Implements

[ILockerTLockerLocker](#)

## ► See Also

Reference

[MonitorLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLocker Methods

The [MonitorLocker](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)

---

≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">InLock(Action)</a>	Executes an action from within a Monitor
≡	<a href="#">InLockTValue(FuncTValue)</a>	Executes a function from within a Monitor
≡	<a href="#">InLockAsync(Action)</a>	Executes an action from within a Monitor
≡	<a href="#">InLockAsyncTValue(FuncTValue)</a>	Executes a function from within a Monitor
≡	<a href="#">Lock</a>	Locks the resource
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns a

string that represents the current object.  
(Inherited from [Object](#).)



### [Unlock](#)

Unlock the resource

[Top](#)

## Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded.

Performs the function in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
---	---

---

 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
---	---

---

 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
---	---

---

 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
--	---

---

 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
---	---

---

 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by
--	--

[MonitorExtensions.\)](#)

---

[Top](#)

## ▲ See Also

Reference

[MonitorLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerInLock Method

## ► Overload List

	Name	Description
≡	<a href="#">InLock(Action)</a>	Executes an action from within a Monitor
≡	<a href="#">InLockTValue(FuncTValue)</a>	Executes a function from within a Monitor

[Top](#)

## ► See Also

Reference

[MonitorLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerInLock Method (Action)

Executes an action from within a Monitor

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public void InLock(  
    Action action  
)
```

[Copy](#)

## Parameters

*action*

Type: [SystemAction](#)

The action to run

Implements

[ILockerInLock\(Action\)](#)

## ► See Also

Reference

[MonitorLocker Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)



## Tungsten

W<sup>74</sup>

# MonitorLockerInLock*TValue* Method (Func*TValue*)

Executes a function from within a Monitor

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public TValue InLock<TValue>(  
    Func<TValue> func  
)
```

[Copy](#)

## Parameters

*func*

Type: [SystemFunc\*TValue\*](#)

The function to run

## Type Parameters

*TValue*

The type of return value

## Return Value

Type: *TValue*

The result of the function call (a value of type TValue)

## ► See Also

### Reference

MonitorLocker Class  
InLock Overload  
W.Threading.Lockers Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerInLockAsync Method

## ▪ Overload List

Name	Description
 <a href="#">InLockAsync(Action)</a>	Executes an action from within a Monitor
 <a href="#">InLockAsyncTValue(FuncTValue)</a>	Executes a function from within a Monitor

[Top](#)

## ▪ See Also

### Reference

[MonitorLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerInLockAsync Method (Action)

Executes an action from within a Monitor

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public Task InLockAsync(  
    Action action  
)
```

[Copy](#)

## Parameters

*action*

Type: [SystemAction](#)

The action to run

## Return Value

Type: [Task](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.MonitorLocker.InLockAsync(System.Action)"]

## Implements

[ILockerInLockAsync\(Action\)](#)

## ► See Also

### Reference

[MonitorLocker Class](#)

InLockAsync Overload  
W.Threading.Lockers Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerInLockAsync<TValue> Method (Func<TValue>)

Executes a function from within a Monitor

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public Task<TValue> InLockAsync<TValue>(
    Func<TValue> func
)
```

### Parameters

*func*  
Type: [SystemFunc<TValue>](#)  
The function to run

### Type Parameters

*TValue*  
The type of return value

**Return Value**  
Type: [Task<TValue>](#)  
The result of the function call (a value of type TValue)

## ► See Also

Reference

[MonitorLocker Class](#)  
[InLockAsync Overload](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerLock Method

Locks the resource

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public void Lock()
```

Return Value

Type:

Returns True

## ► See Also

Reference

[MonitorLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerUnlock Method

Unlock the resource

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public void Unlock()
```

## ► See Also

Reference

[MonitorLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerTState Class

Extends MonitorLocker with an internal state variable

## ► Inheritance Hierarchy

SystemObject [W.Threading.LockersStateLockerMonitorLocker](#),  
*TState*  
[W.Threading.LockersMonitorLockerTState](#)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public class MonitorLocker<TState> : StateLocker<
```

### Type Parameters

*TState*

The state Type

The [MonitorLockerTState](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">MonitorLockerTState</a>	Initializes a new instance of the <a href="#">MonitorLockerTState</a> class

[Top](#)

## ► Properties

Name	Description
	<a href="#">Locker</a> The locking mechanism (SpinLock, Monitor, SemaphoreSlim, ReaderWriterLock) (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

[Top](#)

## ► Methods

Name	Description
	<a href="#">Equals</a> Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a> Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a> Serves as the default hash function.

		(Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)	
 <a href="#">InLock(Action)</a>	Performs an action from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)	
 <a href="#">InLock(ActionTState)</a>	Performs an action from within a lock, passing in the current state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)	
 <a href="#">InLock(FuncTState, TState)</a>	Performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)	
 <a href="#">InLockTResult(FuncTResult)</a>	Performs a function from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)	

≡	InLockAsync(Action)	Asynchronously performs an action from within a lock (Inherited from StateLockerTLocker, TState.)
≡	InLockAsync(ActionTState)	Asynchronously performs an action from within a lock, passing in the current state (Inherited from StateLockerTLocker, TState.)
≡	InLockAsync(FuncTState, TState)	Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from StateLockerTLocker, TState.)
≡	InLockAsyncTResult(FuncTResult)	Asynchronously performs a function from within a lock (Inherited from StateLockerTLocker, TState.)
💡	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from

[Object.\)](#)

---

 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object.)</a>
--	--

---

[Top](#)

## Fields

Name	Description
 <a href="#">State</a>	The internal state (Inherited from <a href="#">StateLocker</a> , <a href="#">TLocker</a> , <a href="#">TState</a> .)

---

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by

[PropertyHostExtensions.\)](#)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	Scans each field and

[MarkAsClean](#) property of type `IProperty` and sets it's `IsDirty` flag to `false`  
(Defined by [PropertyHostExtensions](#).)



[Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions](#).)

[Top](#)

## ► Remarks

Same as `StateLocker<MonitorLocker<, TState>`

## ► See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerTState Constructor

Initializes a new instance of the [MonitorLockerTState](#) class

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public MonitorLocker()
```

## ► See Also

Reference

[MonitorLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerTState Properties

The [MonitorLockerTState](#) generic type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The locking mechanism (SpinLock, Monitor, SemaphoreSlim, ReaderWriterLock) (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

[Top](#)

## ► See Also

[Reference](#)

[MonitorLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerTState Methods

The [MonitorLockerTState](#) generic type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)

Object.)

---

≡	InLock(Action)	Performs an action from within a lock (Inherited from StateLockerTLocker, TState.)
≡	InLock(Action TState)	Performs an action from within a lock, passing in the current state (Inherited from StateLockerTLocker, TState.)
≡	InLock(FuncTState, TState)	Performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from StateLockerTLocker, TState.)
≡	InLockTResult(FuncTResult)	Performs a function from within a lock (Inherited from StateLockerTLocker, TState.)
≡	InLockAsync(Action)	Asynchronously performs an action from within a lock (Inherited from StateLockerTLocker, TState.)

---

≡	<a href="#">InLockAsync(ActionTState)</a>	Asynchronously performs an action from within a lock, passing in the current state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡	<a href="#">InLockAsync(FuncTState, TState)</a>	Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡	<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchronously performs a function from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from

[Object.\)](#)

---

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs

the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
---	---	---

---

	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
---	-------------------------	---

---

	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
--	----------------------	---

---

	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
---	-----------------------------	---

---

	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)
---	------------------------	---

[Top](#)

## ▲ See Also

Reference

MonitorLockerTState Class  
W.Threading.Lockers Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# MonitorLockerTState Fields

The [MonitorLockerTState](#) generic type exposes the following members.

## ► Fields

	Name	Description
	<a href="#">State</a>	The internal state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

[Top](#)

## ► See Also

[Reference](#)

[MonitorLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker Class

Uses ReaderWriterLockSlim to provide resource locking

## ► Inheritance Hierarchy

[SystemObject](#) [W.Threading.LockersReaderWriterLocker](#)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public class ReaderWriterLocker : IDisposable,  
    ILocker<ReaderWriterLockSlim>, ILocker
```

The [ReaderWriterLocker](#) type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">ReaderWriterLocker</a>	Constructs a new ReaderWriterLocker with a LockRecursionPolicy of NoRecursion.
 <a href="#">ReaderWriterLocker(LockRecursionPolicy)</a>	Constructs a new ReaderWriterLocker using the specified LockRecursionPolicy.

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The ReaderWriterLockSlim used to perform locks

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Dispose</a>	Disposes the instance and releases resources
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)

[Object.\)](#)

≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object.</a> )
≡	<a href="#">InLock(Action)</a>	Performs the action in a read lock
≡	<a href="#">InLock(LockTypeEnum, Action)</a>	Executes an action from within a <a href="#">ReaderWriterLockSli</a>
≡	<a href="#">InLockTResult(FuncTResult)</a>	Performs the function in a read lock
≡	<a href="#">InLock TValue(LockTypeEnum, FuncTValue)</a>	Executes a function from within a <a href="#">ReaderWriterLockSli</a>
≡	<a href="#">InLockAsync(Action)</a>	Asynchronously performs the action in a read lock
≡	<a href="#">InLockAsync(LockTypeEnum, Action)</a>	Executes an action from within a <a href="#">ReaderWriterLockSli</a>
≡	<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchronously performs the function in a read lock
≡	<a href="#">InLockAsync TValue(LockTypeEnum, FuncTValue)</a>	Executes a function from within a <a href="#">ReaderWriterLockSli</a>
≡	<a href="#">Lock</a>	Enters a read or writ

lock on the  
ReaderWriterLockSli

 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Unlock</a>	Exits a read or write lock on the ReaderWriterLockSli

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members

(Defined by  
[PropertyHostExtensions](#).)

 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)



### [MarkAsClean](#)

Scans each field and property of type `IProperty` and sets it's `IsDirty` flag to false  
(Defined by [PropertyHostExtensions.](#))



### [Unlock](#)

Performs a Monitor unlock  
(Defined by [MonitorExtensions.](#))

---

[Top](#)

## ► Remarks

Can be overridden to provide additional functionality

## ► See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker Constructor

## ▪ Overload List

Name	Description
 <a href="#">ReaderWriterLocker</a>	Constructs a new ReaderWriterLock with a LockRecursion of NoRecursion.
 <a href="#">ReaderWriterLocker(LockRecursionPolicy)</a>	Constructs a new ReaderWriterLock using the specified LockRecursion.

[Top](#)

## ▪ See Also

Reference

[ReaderWriterLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker Constructor

Constructs a new ReaderWriterLocker with a LockRecursionPolicy of NoRecursion

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public ReaderWriterLocker()
```

## ► See Also

Reference

[ReaderWriterLocker Class](#)

[ReaderWriterLocker Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker Constructor (LockRecursionPolicy)

Constructs a new ReaderWriterLocker using the specified LockRecursionPolicy

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public ReaderWriterLocker(  
    LockRecursionPolicy lockPolicy  
)
```

### Parameters

*lockPolicy*

Type: [System.ThreadingLockRecursionPolicy](#)

The lock recursion policy to use

## ► See Also

[Reference](#)

[ReaderWriterLocker Class](#)

[ReaderWriterLocker Overload](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker Properties

The [ReaderWriterLocker](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The ReaderWriterLockSlim used to perform locks

[Top](#)

## ► See Also

Reference

[ReaderWriterLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerLocker Property

The ReaderWriterLockSlim used to perform locks

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public ReaderWriterLockSlim Locker { get; }
```

Property Value

Type: [ReaderWriterLockSlim](#)

Implements

[ILockerTLockerLocker](#)

## ► See Also

Reference

[ReaderWriterLocker Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker Methods

The [ReaderWriterLocker](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disposes the instance and releases resources
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance.

		(Inherited from Object.)
≡♪	<a href="#">InLock(Action)</a>	Performs the action i a read lock
≡♪	<a href="#">InLock(LockTypeEnum, Action)</a>	Executes an action from within a ReaderWriterLockSli
≡♪	<a href="#">InLockTResult(FuncTResult)</a>	Performs the function in a read lock
≡♪	<a href="#">InLock TValue(LockTypeEnum, FuncTValue)</a>	Executes a function from within a ReaderWriterLockSli
≡♪	<a href="#">InLockAsync(Action)</a>	Asynchronously performs the action i a read lock
≡♪	<a href="#">InLockAsync(LockTypeEnum, Action)</a>	Executes an action from within a ReaderWriterLockSli
≡♪	<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchronously performs the function in a read lock
≡♪	<a href="#">InLockAsync TValue(LockTypeEnum, FuncTValue)</a>	Executes a function from within a ReaderWriterLockSli
≡♪	<a href="#">Lock</a>	Enters a read or write lock on the ReaderWriterLockSli
≡♪	<a href="#">MemberwiseClone</a>	Creates a shallow

copy of the current [Object](#).  
(Inherited from [Object](#).)

 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Unlock</a>	Exits a read or write lock on the <a href="#">ReaderWriterLockSlim</a> .

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has <a href="#">IOwnedProperty</a> members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded.

Performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockTType\(FuncTType\)](#)  
Overloaded.  
Performs the function in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockAsync\(Action\)](#)  
Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [InLockAsyncTType\(FuncTType\)](#)  
Overloaded.  
Asynchronously performs the action in a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [IsDirty](#)  
Scans the IsDirty value of each field and property of type IProperty  
(Defined by [PropertyHostExtensions](#).)

---

⬇️ [Lock](#)  
Performs a Monitor lock  
(Defined by [MonitorExtensions](#).)

---

⬇️ [MarkAsClean](#)  
Scans each field and property of type IProperty and sets its IsDirty flag to false

(Defined by  
[PropertyHostExtensions](#).)



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[ReaderWriterLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerDispose Method

Disposes the instance and releases resources

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

```
public void Dispose()
```

[Copy](#)

Implements  
[IDisposableDispose](#)

## ► See Also

Reference

[ReaderWriterLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLock Method

## Overload List

Name	Description
<a href="#">InLock(Action)</a>	Performs the action in a read lock
<a href="#">InLockTResult(FuncTResult)</a>	Performs the function in a read lock
<a href="#">InLock(LockTypeEnum, Action)</a>	Executes an action from within a ReaderWriterLockSlim
<a href="#">InLock TValue(LockTypeEnum, FuncTValue)</a>	Executes a function from within a ReaderWriterLockSlim

[Top](#)

## See Also

Reference

[ReaderWriterLocker Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLock Method (Action)

Performs the action in a read lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public void InLock(  
    Action action  
)
```

[Copy](#)

## Parameters

*action*

Type: [SystemAction](#)

The action to perform

Implements

[ILockerInLock\(Action\)](#)

## ► See Also

Reference

[ReaderWriterLocker Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)



## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLock $TResult$ Method ( $\text{Func}TResult$ )

Performs the function in a read lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public TResult InLock<TResult>(
    Func<TResult> func
)
```

## Parameters

*func*

Type: [SystemFuncTResult](#)

The function to perform

## Type Parameters

*TResult*

[Missing <typeparam name="TResult"/> documentation for "M:W.Threading.Lockers.ReaderWriterLocker.InLock`1(System.Func{`0})"]

## Return Value

Type: *TResult*

[Missing <returns> documentation for "M:W.Threading.Lockers.ReaderWriterLocker.InLock`1(System.Func{`0})"]

## Implements

[ILockerInLockTResult\(FuncTResult\)](#)

## See Also

Reference

[ReaderWriterLocker Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLock Method (LockTypeEnum, Action)

Executes an action from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public void InLock(
    LockTypeEnum lockType,
    Action action
)
```

### Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)  
[Missing <param name="lockType"/> documentation for  
"M:W.Threading.Lockers.ReaderWriterLocker.InLock(W.Threading.LockT

*action*

Type: [SystemAction](#)  
The action to run

## ► See Also

[Reference](#)

[ReaderWriterLocker Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLock<TValue> Method (LockTypeEnum, Func<TValue>)

Executes a function from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public TValue InLock<TValue>(
    LockTypeEnum lockType,
    Func<TValue> func
)
```

## Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

[Missing <param name="lockType"/> documentation for

"M:W.Threading.Lockers.ReaderWriterLocker.InLock`1(W.Threading.Lockers.LockTypeEnum, SystemFunc`1[TValue])"

*func*

Type: [SystemFunc<TValue>](#)

The function to run

## Type Parameters

*TValue*

The type of return value

## Return Value

Type: *TValue*

The result of the function call (a value of type TValue)

## See Also

[Reference](#)

[ReaderWriterLocker Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLockAsync Method

## ► Overload List

Name	Description
<a href="#">InLockAsync(Action)</a>	Asynchronously performs the action in a read lock
<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchronously performs the function in a read lock
<a href="#">InLockAsync(LockTypeEnum, Action)</a>	Executes an action from within a ReaderWriterLockSli
<a href="#">InLockAsync TValue(LockTypeEnum, FuncTValue)</a>	Executes a function from within a ReaderWriterLockSli

[Top](#)

## ► See Also

### Reference

[ReaderWriterLocker Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLockAsync Method (Action)

Asynchronously performs the action in a read lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public Task InLockAsync(  
    Action action  
)
```

[Copy](#)

### Parameters

*action*

Type: [SystemAction](#)

The action to perform

### Return Value

Type: [Task](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.ReaderWriterLocker.InLockAsync(System.Action)"]

### Implements

[ILockerInLockAsync\(Action\)](#)

## ► See Also

### Reference

[ReaderWriterLocker Class](#)

InLockAsync Overload  
W.Threading.Lockers Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLockAsync<TResult> Method (Func<TResult>)

Asynchronously performs the function in a read lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public Task<TResult> InLockAsync<TResult>(
    Func<TResult> func
)
```

### Parameters

*func*

Type: [System.Func<TResult>](#)

The action to perform

### Type Parameters

*TResult*

[Missing <typeparam name="TResult"/> documentation for "M:W.Threading.Lockers.ReaderWriterLocker.InLockAsync`1(System.Func{`0})"]

### Return Value

Type: [Task<TResult>](#)

[Missing <returns> documentation for "M:W.Threading.Lockers.ReaderWriterLocker.InLockAsync`1(System.Func{`0})"]

### Implements

[ILockerInLockAsyncTResult\(Func<TResult>\)](#)

## See Also

Reference

[ReaderWriterLocker Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLockAsync Method (LockTypeEnum, Action)

Executes an action from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public Task InLockAsync(  
    LockTypeEnum lockType,  
    Action action  
)
```

### Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

[Missing <param name="lockType"/> documentation for  
"M:W.Threading.Lockers.ReaderWriterLocker.InLockAsync(W.Threading.Lockers.

*action*

Type: [SystemAction](#)

The action to run

### Return Value

Type: [Task](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.ReaderWriterLocker.InLockAsync(W.Threading.Lockers.Lock

## ► See Also

## Reference

[ReaderWriterLocker Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerInLockAsync<TValue> Method (LockTypeEnum, Func<TValue>)

Executes a function from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public Task<TValue> InLockAsync<TValue>(
    LockTypeEnum lockType,
    Func<TValue> func
)
```

## Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

[Missing <param name="lockType"/> documentation for "M:W.Threading.Lockers.ReaderWriterLocker.InLockAsync`1(W.Threading.Locke

*func*

Type: [SystemFunc<TValue>](#)

The function to run

## Type Parameters

*TValue*

The type of return value

## Return Value

Type: [Task<TValue>](#)

The result of the function call (a value of type TValue)

## ▲ See Also

Reference

[ReaderWriterLocker Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerLock Method

Enters a read or write lock on the ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public void Lock(  
    LockTypeEnum lockType  
)
```

### Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

The type of lock to enter

## ► See Also

Reference

[ReaderWriterLocker Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerUnlock Method

Exits a read or write lock on the ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public void Unlock(  
    LockTypeEnum lockType  
)
```

[Copy](#)

## Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

The type of lock to exit

## ► See Also

### Reference

[ReaderWriterLocker Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTState Class

Extends ReaderWriterLocker with an internal state variable

## ► Inheritance Hierarchy

SystemObject W.Threading.LockersReaderWriterLockerTState  
WLockableSlim TValue

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public class ReaderWriterLocker<TState> : IReader
    IDisposable
```

### Type Parameters

*TState*

The state Type

The [ReaderWriterLockerTState](#) type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">ReaderWriterLockerTState</a>	Constructs a new ReaderWriterLocker

[Top](#)

## ◀ Methods

Name	Description
Dispose	Disposes the ReaderWriterLocker and releases resources
Equals	Determines whether the specified object is equal to the current object. (Inherited from Object.)
Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
GetHashCode	Serves as the default hash function. (Inherited from Object.)
GetState	Retrieves the internal state from within a ReaderWriterLockSli
GetType	Gets the Type of the current instance. (Inherited from

[Object.\)](#)

≡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSli
≡	<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSli
≡	<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSli
≡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSli
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">SetState</a>	Sets the internal stat from within a ReaderWriterLockSli
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## ► Fields

	Name	Description
💡	<a href="#">Locker</a>	The ReaderWriterLocker used to access the State
💡	<a href="#">State</a>	The internal state

[Top](#)

## Extension Methods

	Name	Description
💡 ⚡	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
💡	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
💡	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock

(Defined by  
[MonitorExtensions](#).)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ► See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTState Constructor

Constructs a new ReaderWriterLocker

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public ReaderWriterLocker(
    LockRecursionPolicy lockRecursionPolicy
)
```

### Parameters

*lockRecursionPolicy*

Type: [System.ThreadingLockRecursionPolicy](#)

The lock recursion policy to use

## ► See Also

### Reference

[ReaderWriterLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTState Methods

The [ReaderWriterLockerTState](#) generic type exposes the following members.

## ▲ Methods

Name	Description
 <a href="#">Dispose</a>	Disposes the ReaderWriterLocker and releases resources
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)

[Object.\)](#)

≡	<a href="#">GetState</a>	Retrieves the internal state from within a ReaderWriterLockSli
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object.)</a>
≡	<a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSli
≡	<a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSli
≡	<a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSli
≡	<a href="#">InLockAsyncTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSli
≡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object.</a> (Inherited from <a href="#">Object.)</a>
≡	<a href="#">SetState</a>	Sets the internal stat from within a ReaderWriterLockSli
≡	<a href="#">ToString</a>	Returns a string that

represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

---

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## ▲ See Also

Reference

[ReaderWriterLockerTState Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTStateDispose Method

Disposes the ReaderWriterLocker and releases resources

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

```
public void Dispose()
```

[Copy](#)

Implements  
[IDisposable.Dispose](#)

## ► See Also

Reference

[ReaderWriterLockerTState Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTStateGetState Method

Retrieves the internal state from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public TState GetState()
```

Return Value

Type: [\*TState\*](#)

The current state

Implements

[IReaderWriterStateLockerTStateGetState](#)

## ► See Also

Reference

[ReaderWriterLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTStateInLock Method

## ► Overload List

Name	Description
 <a href="#">InLock(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSlim
 <a href="#">InLockTValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSlim

[Top](#)

## ► See Also

### Reference

[ReaderWriterLockerTState Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker{TState}InLock Method (LockTypeEnum, Action{TState})

Executes an action from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public void InLock(
    LockTypeEnum lockType,
    Action<TState> action
)
```

### Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

Specifies whether to use a Read or Write lock

*action*

Type: [SystemAction{TState}](#)

The action to run

### Implements

[IReaderWriterStateLocker{TState}InLock\(LockTypeEnum, Action{TState}\)](#)

## ► See Also

## Reference

[ReaderWriterLockerTState Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker{TState}InLock{TValue} Method (LockTypeEnum, Func{TState, TValue})

Executes a function from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public TValue InLock<TValue>(  
    LockTypeEnum lockType,  
    Func<TState, TValue> func  
)
```

[Copy](#)

### Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

Specifies whether to use a Read or Write lock

*func*

Type: [SystemFunc{TState, TValue}](#)

The function to run

### Type Parameters

*TValue*

The type of return value

### Return Value

Type: *TValue*

The result of the function call (a value of type *TValue*)

Implements

[IReaderWriterStateLockerTStateInLockTValue\(LockTypeEnum, FuncTState, TValue\)](#)

## ▲ See Also

Reference

[ReaderWriterLockerTState Class](#)

InLock Overload

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTStateInLockAs Method

## ► Overload List

Name	Description
 <a href="#">InLockAsync(LockTypeEnum, ActionTState)</a>	Executes an action from within a ReaderWriterLockSli
 <a href="#">InLockAsync TValue(LockTypeEnum, FuncTState, TValue)</a>	Executes a function from within a ReaderWriterLockSli

[Top](#)

## ► See Also

### Reference

[ReaderWriterLockerTState Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker{TState}InLockAs Method (LockTypeEnum, Action{TState})

Executes an action from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ◀ Syntax

C#

Copy

```
public Task InLockAsync(  
    LockTypeEnum lockType,  
    Action<TState> action  
)
```

### Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

Specifies whether to use a Read or Write lock

*action*

Type: [SystemAction{TState}](#)

The action to run

### Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:W.Threading.Lockers.ReaderWriterLocker`1.InLockAsync(W.Threading.Lockers.LockTypeEnum, SystemAction{TState})"]

### Implements

[IReaderWriterStateLocker{TState}InLockAsync\(LockTypeEnum, Action{TState}\)](#)

## ▲ See Also

### Reference

[ReaderWriterLockerTState Class](#)  
[InLockAsync Overload](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker{TState}InLockAs Method (LockTypeEnum, Func{TStat

Executes a function from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public Task<TValue> InLockAsync<TValue>(
    LockTypeEnum lockType,
    Func<TState, TValue> func
)
```

### Parameters

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

Specifies whether to use a Read or Write lock

*func*

Type: [SystemFunc{TState, TValue}](#)

The function to run

### Type Parameters

*TValue*

The type of return value

### Return Value

Type: [Task{TValue}](#)

The result of the function call (a value of type TValue)

Implements

[IReaderWriterStateLockerTStateInLockAsyncTValue\(LockTypeEnum, FuncTState, TValue\)](#)

## See Also

Reference

[ReaderWriterLockerTState Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTStateSetState Method

Sets the internal state from within a ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public void SetState(  
    TState newState  
)
```

[Copy](#)

## Parameters

*newState*

Type: [TState](#)

The new value

## Implements

[IReaderWriterStateLockerTStateSetState\(TState\)](#)

## ► See Also

### Reference

[ReaderWriterLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTState

## Fields

The [ReaderWriterLockerTState](#) generic type exposes the following members.

### ► Fields

	Name	Description
	<a href="#">Locker</a>	The ReaderWriterLocker used to access the State
	<a href="#">State</a>	The internal state

[Top](#)

### ► See Also

#### Reference

[ReaderWriterLockerTState Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockerTStateLocker Field

The ReaderWriterLocker used to access the State

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
protected ReaderWriterLocker Locker
```

[Copy](#)

Field Value

Type: [ReaderWriterLocker](#)

## ► See Also

Reference

[ReaderWriterLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLocker{TState} State Field

The internal state

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
protected TState State
```

Field Value

Type: [TState](#)

## ► See Also

Reference

[ReaderWriterLocker{TState} Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensions Class

Extensions to simplify locking with ReaderWriterLockSlim

## ► Inheritance Hierarchy

[SystemObject](#) [W.Threading.Lockers](#)[ReaderWriterLockSlimExtensions](#)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public static class ReaderWriterLockSlimExtensi
```

The [ReaderWriterLockSlimExtensions](#) type exposes the following members.

## ► Methods

	Name	Description
≡ S	<a href="#">InLock(ReaderWriterLockSlim, LockTypeEnum, Action)</a>	Performs the a lock
≡ S	<a href="#">InLockTType(ReaderWriterLockSlim, LockTypeEnum, FuncTType)</a>	Performs the f in a lock
≡ S	<a href="#">InLockAsync(ReaderWriterLockSlim, LockTypeEnum, Action)</a>	Asynchronous performs the a

a lock

---

≡ S [InLockAsyncTType\(ReaderWriterLockSlim, LockTypeEnum, FuncTType\)](#) Asynchronous performs the fi  
in a lock

---

≡ S [Lock](#) Enters a read lock on the ReaderWriterL

---

≡ S [Unlock](#) Exits a read or lock on the ReaderWriterL

[Top](#)

## ▲ See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensions Methods

The [ReaderWriterLockSlimExtensions](#) type exposes the following members.

## ▲ Methods

	Name	Description
≡  	<a href="#">InLock(ReaderWriterLockSlim, LockTypeEnum, Action)</a>	Performs the action in a lock
≡  	<a href="#">InLockTType(ReaderWriterLockSlim, LockTypeEnum, FuncTType)</a>	Performs the function in a lock
≡  	<a href="#">InLockAsync(ReaderWriterLockSlim, LockTypeEnum, Action)</a>	Asynchronous performs the action in a lock
≡  	<a href="#">InLockAsyncTType(ReaderWriterLockSlim, LockTypeEnum, FuncTType)</a>	Asynchronous performs the function in a lock
≡  	<a href="#">Lock</a>	Enters a read lock on the ReaderWriterLockSlim
≡  	<a href="#">Unlock</a>	Exits a read or write lock on the ReaderWriterLockSlim

[Top](#)

## ▲ See Also

### Reference

[ReaderWriterLockSlimExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensionsInL Method

## ► Overload List

Name	Description
  <a href="#">InLock(ReaderWriterLockSlim, LockTypeEnum, Action)</a>	Performs the action in a lock
  <a href="#">InLockTType(ReaderWriterLockSlim, LockTypeEnum, FuncTType)</a>	Performs the function in a lock

[Top](#)

## ► See Also

### Reference

[ReaderWriterLockSlimExtensions Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensionsInL Method (ReaderWriterLockSlim, LockTypeEnum, Action)

Performs the action in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static void InLock(
    this ReaderWriterLockSlim this,
    LockTypeEnum lockType,
    Action action
)
```

## Parameters

*this*

Type: [System.Threading.ReaderWriterLockSlim](#)

The ReaderWriterLockSlim to provide resource locking

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

The type of lock to obtain

*action*

Type: [SystemAction](#)

The action to perform

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [ReaderWriterLockSlim](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

Reference

[ReaderWriterLockSlimExtensions Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensionsInL Method (ReaderWriterLockSlim, LockTypeEnum, Func<TType>)

Performs the function in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static TType InLock<TType>(
    this ReaderWriterLockSlim this,
    LockTypeEnum lockType,
    Func<TType> func
)
```

## Parameters

*this*

Type: [System.Threading.ReaderWriterLockSlim](#)

The ReaderWriterLockSlim to provide resource locking

*lockType*

Type: [W.Threading.Lockers.LockTypeEnum](#)

The type of lock to obtain

*func*

Type: [SystemFunc<TType>](#)

The function to perform

## Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for  
"M:W.Threading.Lockers.ReaderWriterLockSlimExtensions.InLock`1(System.Thr

## Return Value

Type: *TType*

[Missing <returns> documentation for  
"M:W.Threading.Lockers.ReaderWriterLockSlimExtensions.InLock`1(System.Threading

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [ReaderWriterLockSlim](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[ReaderWriterLockSlimExtensions Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensionsInL Method

## ► Overload List

Name	Description
  <a href="#">InLockAsync(ReaderWriterLockSlim, LockTypeEnum, Action)</a>	Asynchronous performs the action in a lock
  <a href="#">InLockAsyncTType(ReaderWriterLockSlim, LockTypeEnum, FuncTType)</a>	Asynchronous performs the function in a lock

[Top](#)

## ► See Also

Reference

[ReaderWriterLockSlimExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensionsInL Method (ReaderWriterLockSlim, LockTypeEnum, Action)

Asynchronously performs the action in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static Task InLockAsync(
    this ReaderWriterLockSlim this,
    LockTypeEnum lockType,
    Action action
)
```

## Parameters

*this*

Type: [System.Threading.ReaderWriterLockSlim](#)

The ReaderWriterLockSlim to provide resource locking

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

The type of lock to obtain

*action*

Type: [SystemAction](#)

The action to perform

## Return Value

Type: Task

[Missing <returns> documentation for  
"M:W.Threading.Lockers.ReaderWriterLockSlimExtensions.InLockAsync(System.Threa

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [ReaderWriterLockSlim](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[ReaderWriterLockSlimExtensions Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensionsInL Method (ReaderWriterLockSlim, Loc Func<TType>)

Asynchronously performs the function in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static Task<TType> InLockAsync<TType>(
    this ReaderWriterLockSlim this,
    LockTypeEnum lockType,
    Func<TType> func
)
```

## Parameters

*this*

Type: [System.Threading.ReaderWriterLockSlim](#)

The ReaderWriterLockSlim to provide resource locking

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)

The type of lock to obtain

*func*

Type: [SystemFunc<TType>](#)

The function to perform

## Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for  
"M:W.Threading.Lockers.ReaderWriterLockSlimExtensions.InLockAsync`1(Syste

## Return Value

Type: [Task<TType>](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.ReaderWriterLockSlimExtensions.InLockAsync`1(System.Thi

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [ReaderWriterLockSlim](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[ReaderWriterLockSlimExtensions Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensionsLock Method

Enters a read or write lock on the ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static void Lock(
    this ReaderWriterLockSlim this,
    LockTypeEnum lockType
)
```

### Parameters

*this*

Type: [System.Threading.ReaderWriterLockSlim](#)  
The object to provide resource locking

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)  
The type of lock to enter

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [ReaderWriterLockSlim](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ▲ See Also

### Reference

[ReaderWriterLockSlimExtensions Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# ReaderWriterLockSlimExtensionsUnlock Method

Exits a read or write lock on the ReaderWriterLockSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static void Unlock(
    this ReaderWriterLockSlim this,
    LockTypeEnum lockType
)
```

### Parameters

*this*

Type: [System.Threading.ReaderWriterLockSlim](#)  
The object to provide resource locking

*lockType*

Type: [W.Threading.LockersLockTypeEnum](#)  
The type of lock to exit

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [ReaderWriterLockSlim](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ▲ See Also

### Reference

[ReaderWriterLockSlimExtensions Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimExtensions Class

Extensions to simplify locking with SemaphoreSlim

## ► Inheritance Hierarchy

[SystemObject](#) [W.Threading.LockersSemaphoreSlimExtensions](#)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static class SemaphoreSlimExtensions
```

## ► Methods

Name	Description
  <a href="#">InLock(SemaphoreSlim, Action)</a>	Performs the action in a lock
  <a href="#">InLockTType(SemaphoreSlim, FuncTType)</a>	Performs the action in a lock
  <a href="#">InLockAsync(SemaphoreSlim, Action)</a>	Asynchronously performs the action in a lock
  <a href="#">InLockAsyncTType(SemaphoreSlim,</a>	Asynchronously

`FuncTType)`

performs the  
function in a  
lock

---

[Top](#)

## ▲ See Also

[Reference](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimExtensions Methods

## ► Methods

Name	Description
  <a href="#">InLock(SemaphoreSlim, Action)</a>	Performs the action in a lock
  <a href="#">InLockTType(SemaphoreSlim, FuncTType)</a>	Performs the action in a lock
  <a href="#">InLockAsync(SemaphoreSlim, Action)</a>	Asynchronously performs the action in a lock
  <a href="#">InLockAsyncTType(SemaphoreSlim, FuncTType)</a>	Asynchronously performs the function in a lock

[Top](#)

## ► See Also

[Reference](#)

[SemaphoreSlimExtensions Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# SemaphoreSlimExtensionsInLock Method

## ▪ Overload List

	Name	Description
 	<a href="#">InLock(SemaphoreSlim, Action)</a>	Performs the action in a lock
 	<a href="#">InLockTType(SemaphoreSlim, FuncTType)</a>	Performs the action in a lock

[Top](#)

## ▪ See Also

Reference

[SemaphoreSlimExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimExtensionsInLock Method (SemaphoreSlim, Action)

Performs the action in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static void InLock(  
    this SemaphoreSlim this,  
    Action action  
)
```

[Copy](#)

### Parameters

*this*

Type: [System.ThreadingSemaphoreSlim](#)

The SemaphoreSlim to provide resource locking

*action*

Type: [SystemAction](#)

The action to perform

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SemaphoreSlim](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ▲ See Also

Reference

[SemaphoreSlimExtensions Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimExtensionsInLock{T} Method (SemaphoreSlim, Func{T}Type)

Performs the action in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static TType InLock<TType>(  
    this SemaphoreSlim this,  
    Func<TType> func  
)
```

[Copy](#)

## Parameters

*this*

Type: [System.ThreadingSemaphoreSlim](#)

The SemaphoreSlim to provide resource locking

*func*

Type: [SystemFunc{T}Type](#)

The function to perform

## Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for "M:W.Threading.Lockers.SemaphoreSlimExtensions.InLock`1(System.Threading

## Return Value

Type: *TType*

[Missing <returns> documentation for  
"M:W.Threading.Lockers.SemaphoreSlimExtensions.InLock`1(System.Threading.Semaphor

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SemaphoreSlim](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[SemaphoreSlimExtensions Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimExtensionsInLockAs Method

## ▪ Overload List

Name	Description
  <a href="#">InLockAsync(SemaphoreSlim, Action)</a>	Asynchronously performs the action in a lock
  <a href="#">InLockAsyncTType(SemaphoreSlim, FuncTType)</a>	Asynchronously performs the function in a lock

[Top](#)

## ▪ See Also

Reference

[SemaphoreSlimExtensions Class](#)  
[W.Threading.Lockers Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimExtensionsInLockAs Method (SemaphoreSlim, Action)

Asynchronously performs the action in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static Task InLockAsync(  
    this SemaphoreSlim this,  
    Action action  
)
```

[Copy](#)

### Parameters

*this*

Type: [System.ThreadingSemaphoreSlim](#)

The SemaphoreSlim to provide resource locking

*action*

Type: [SystemAction](#)

The action to perform

### Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:W.Threading.Lockers.SemaphoreSlimExtensions.InLockAsync(System.Threading.S

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SemaphoreSlim](#). When you use

instance method syntax to call this method, omit the first parameter.  
For more information, see [Extension Methods \(Visual Basic\)](#) or  
[Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

- [SemaphoreSlimExtensions Class](#)
- [InLockAsync Overload](#)
- [W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimExtensionsInLockAs Method (SemaphoreSlim, Func<TType>)

Asynchronously performs the function in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static Task<TType> InLockAsync<TType>(
    this SemaphoreSlim this,
    Func<TType> func
)
```

### Parameters

*this*

Type: [System.ThreadingSemaphoreSlim](#)

The SemaphoreSlim to provide resource locking

*func*

Type: [SystemFunc<TType>](#)

The function to perform

### Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for "M:W.Threading.Lockers.SemaphoreSlimExtensions.InLockAsync`1(System.ThreadingSemaphoreSlim, System.Func`1(TType))"]

### Return Value

Type: [Task<TType>](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.SemaphoreSlimExtensions.InLockAsync`1(System.Threading

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SemaphoreSlim](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[SemaphoreSlimExtensions Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLocker Class

Uses SemaphoreSlim to provide resource locking

## ► Inheritance Hierarchy

[SystemObject](#) [W.Threading.LockersSemaphoreSlimLocker](#)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public class SemaphoreSlimLocker : ILocker<SemaphoreSlim>,  
    ILocker, IDisposable
```

The [SemaphoreSlimLocker](#) type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">SemaphoreSlimLocker</a>	Constructs a new SemaphoreSlimLocker with an initial request count of 1 and maximum request count of 1
 <a href="#">SemaphoreSlimLocker(Int32)</a>	Constructs a new SemaphoreSlimLocker



[SemaphoreSlimLocker\(Int32, Int32\)](#) Constructs a new SemaphoreSlimLocker

[Top](#)

## Properties

	Name	Description
	<a href="#">Locker</a>	The SemaphoreSlim used to perform locks

[Top](#)

## Methods

	Name	Description
	<a href="#">Dispose</a>	Disposes the SemaphoreSlimLocker and releases resources
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)

---

≡	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">InLock(Action)</a>	Executes an action from within a <a href="#">SemaphoreSlim</a>
≡	<a href="#">InLockTValue(FuncTValue)</a>	Executes a function from within a <a href="#">SemaphoreSlim</a>
≡	<a href="#">InLockAsync(Action)</a>	Executes an action from within a <a href="#">SemaphoreSlim</a>
≡	<a href="#">InLockAsyncTValue(FuncTValue)</a>	Executes a function from within a <a href="#">SemaphoreSlim</a>
≡	<a href="#">Lock</a>	Locks the resource
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns a string that represents the current object.

---

(Inherited from  
[Object](#).)



## [Unlock](#)

Unlocks the resource

[Top](#)

# ◀ Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ▲ See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLocker Constructor

## ▪ Overload List

Name	Description
 <a href="#">SemaphoreSlimLocker</a>	Constructs a new SemaphoreSlimLocker with an initial request count of 1 and maximum request count of 1
 <a href="#">SemaphoreSlimLocker(Int32)</a>	Constructs a new SemaphoreSlimLocker
 <a href="#">SemaphoreSlimLocker(Int32, Int32)</a>	Constructs a new SemaphoreSlimLocker

[Top](#)

## ▪ See Also

Reference

[SemaphoreSlimLocker Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLocker Constructor

Constructs a new SemaphoreSlimLocker with an initial request count of 1 and maximum request count of 1

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public SemaphoreSlimLocker()
```

## ► See Also

Reference

[SemaphoreSlimLocker Class](#)

[SemaphoreSlimLocker Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLocker Constructor (Int32)

Constructs a new SemaphoreSlimLocker

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public SemaphoreSlimLocker(  
    int initialCount  
)
```

[Copy](#)

### Parameters

*initialCount*

Type: [System.Int32](#)

The initial number of requests that the semaphore can grant concurrently

## ► See Also

[Reference](#)

[SemaphoreSlimLocker Class](#)

[SemaphoreSlimLocker Overload](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLocker Constructor (Int32, Int32)

Constructs a new SemaphoreSlimLocker

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public SemaphoreSlimLocker(  
    int initialCount,  
    int maxCount  
)
```

[Copy](#)

## Parameters

*initialCount*

Type: [SystemInt32](#)

The initial number of requests that the semaphore can grant concurrently

*maxCount*

Type: [SystemInt32](#)

The maximum number of requests that can be granted concurrently

## ► See Also

[Reference](#)

[SemaphoreSlimLocker Class](#)

[SemaphoreSlimLocker Overload](#)

## [W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLocker Properties

The [SemaphoreSlimLocker](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">Locker</a>	The SemaphoreSlim used to perform locks

[Top](#)

## See Also

### Reference

[SemaphoreSlimLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerLocker Property

The SemaphoreSlim used to perform locks

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public SemaphoreSlim Locker { get; }
```

Property Value

Type: [SemaphoreSlim](#)

Implements

[ILocker](#)  
[TLocker](#)  
[Locker](#)

## ► See Also

Reference

[SemaphoreSlimLocker Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLocker Methods

The [SemaphoreSlimLocker](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Dispose</a>	Disposes the SemaphoreSlimLocker and releases resources
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the

current instance.  
(Inherited from  
[Object](#).)

≡	<a href="#">InLock(Action)</a>	Executes an action from within a SemaphoreSlim
≡	<a href="#">InLockTValue(FuncTValue)</a>	Executes a function from within a SemaphoreSlim
≡	<a href="#">InLockAsync(Action)</a>	Executes an action from within a SemaphoreSlim
≡	<a href="#">InLockAsyncTValue(FuncTValue)</a>	Executes a function from within a SemaphoreSlim
≡	<a href="#">Lock</a>	Locks the resource
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current Object. (Inherited from Object.)
≡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from Object.)
≡	<a href="#">Unlock</a>	Unlocks the resource

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

 <a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

[Reference](#)

[SemaphoreSlimLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLocker.Dispose Method

Disposes the SemaphoreSlimLocker and releases resources

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements  
[IDisposable.Dispose](#)

## ► See Also

Reference

[SemaphoreSlimLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerInLock Method

## ► Overload List

Name	Description
 <a href="#">InLock(Action)</a>	Executes an action from within a SemaphoreSlim
 <a href="#">InLockTValue(Func&lt;T&gt;Value)</a>	Executes a function from within a SemaphoreSlim

[Top](#)

## ► See Also

Reference

[SemaphoreSlimLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerInLock Method (Action)

Executes an action from within a SemaphoreSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public void InLock(  
    Action action  
)
```

[Copy](#)

### Parameters

*action*

Type: [SystemAction](#)

The action to run

Implements

[ILockerInLock\(Action\)](#)

## ► See Also

Reference

[SemaphoreSlimLocker Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)



## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerInLock<TValue> Method (Func<TValue>)

Executes a function from within a SemaphoreSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public TValue InLock<TValue>(  
    Func<TValue> func  
)
```

[Copy](#)

## Parameters

*func*

Type: [System.Func<TValue>](#)

The function to run

## Type Parameters

*TValue*

The type of return value

## Return Value

Type: *TValue*

The result of the function call (a value of type TValue)

## ► See Also

### Reference

SemaphoreSlimLocker Class  
InLock Overload  
W.Threading.Lockers Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerInLockAsync Method

## ▪ Overload List

Name	Description
 <a href="#">InLockAsync(Action)</a>	Executes an action from within a SemaphoreSlim
 <a href="#">InLockAsyncTValue(FuncTValue)</a>	Executes a function from within a SemaphoreSlim

[Top](#)

## ▪ See Also

[Reference](#)

[SemaphoreSlimLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerInLockAsync Method (Action)

Executes an action from within a SemaphoreSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public Task InLockAsync(  
    Action action  
)
```

[Copy](#)

### Parameters

*action*

Type: [SystemAction](#)

The action to run

### Return Value

Type: [Task](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.SemaphoreSlimLocker.InLockAsync(System.Action)"]

### Implements

[ILockerInLockAsync\(Action\)](#)

## ► See Also

### Reference

[SemaphoreSlimLocker Class](#)

InLockAsync Overload  
W.Threading.Lockers Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerInLockAsync` Method (Func<TValue>)

Executes a function from within a SemaphoreSlim

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public Task<TValue> InLockAsync<TValue>(  
    Func<TValue> func  
)
```

## Parameters

*func*

Type: [System.Func<TValue>](#)

The function to run

## Type Parameters

*TValue*

The type of return value

## Return Value

Type: [Task<TValue>](#)

The result of the function call (a value of type TValue)

## ► See Also

### Reference

SemaphoreSlimLocker Class  
InLockAsync Overload  
W.Threading.Lockers Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerLock Method

Locks the resource

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

```
public void Lock()
```

Copy

## ► See Also

[Reference](#)

[SemaphoreSlimLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerUnlock Method

Unlocks the resource

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

```
public void Unlock()
```

Copy

## ► See Also

Reference

[SemaphoreSlimLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerTState Class

Extends SemaphoreSlimLocker with an internal state variable

## ► Inheritance Hierarchy

```
SystemObject W.Threading.LockersStateLockerSemaphoreSlimLocke
TState
    W.Threading.LockersSemaphoreSlimLockerTState
```

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#
public class SemaphoreSlimLocker<TState> : StateL
```

[Copy](#)

### Type Parameters

*TState*

The state Type

The [SemaphoreSlimLockerTState](#) type exposes the following members.

## ► Constructors

	Name	Description
	<a href="#">SemaphoreSlimLockerTState</a>	Initializes a new instance of

the  
`SemaphoreSlimLockerTState`  
class

---

[Top](#)

## ► Properties

	Name	Description
	<a href="#"><code>Locker</code></a>	The locking mechanism (SpinLock, Monitor, SemaphoreSlim, ReaderWriterLock) (Inherited from <a href="#"><code>StateLockerTLocker</code></a> , <a href="#"><code>TState</code></a> .)

---

[Top](#)

## ► Methods

	Name	Description
	<a href="#"><code>Equals</code></a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#"><code>Object</code></a> .)
	<a href="#"><code>Finalize</code></a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#"><code>Object</code></a> .)

[Object.\)](#)

---

 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object.</a> )
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object.</a> )
 <a href="#">InLock(Action)</a>	Performs an action from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLock(ActionTState)</a>	Performs an action from within a lock, passing in the current state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLock(FuncTState, TState)</a>	Performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLockTResult(FuncTResult)</a>	Performs a function

from within a lock  
(Inherited from  
[StateLockerTLocker](#),  
[TState](#).)

---

≡	<a href="#">InLockAsync(Action)</a>	Asynchronously performs an action from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡	<a href="#">InLockAsync(ActionTState)</a>	Asynchronously performs an action from within a lock, passing in the current state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡	<a href="#">InLockAsync(FuncTState, TState)</a>	Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡	<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchronously performs a function from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

---

💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
💡	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## Fields

Name	Description
💡 <a href="#">State</a>	The internal state (Inherited from <a href="#">StateLocker</a> , <a href="#">TLocker</a> , <a href="#">TState</a> .)

---

[Top](#)

## Extension Methods

Name	Description
💡 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
💡 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method

should be called in the constructor of any class which has `IOwnedProperty` members  
(Defined by [PropertyHostExtensions](#).)

---

	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the <code>IsDirty</code> value of each field and property of type <code>IProperty</code> (Defined by <a href="#">PropertyHostExtensions</a> .)

 Lock	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 MarkAsClean	Scans each field and property of type <code>IProperty</code> and sets its <code>IsDirty</code> flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 Unlock	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

[Top](#)

## ► Remarks

Same as `StateLocker<SemaphoreSlimLocker<, TState>`

## ► See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerTState Constructor

Initializes a new instance of the [SemaphoreSlimLockerTState](#) class

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public SemaphoreSlimLocker()
```

## ► See Also

Reference

[SemaphoreSlimLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerTState Properties

The [SemaphoreSlimLockerTState](#) generic type exposes the following members.

## Properties

	Name	Description
	<a href="#">Locker</a>	The locking mechanism (SpinLock, Monitor, SemaphoreSlim, ReaderWriterLock) (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

[Top](#)

## See Also

### Reference

[SemaphoreSlimLockerTState Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerTState Methods

The [SemaphoreSlimLockerTState](#) generic type exposes the following members.

## Methods

Name	Description
<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the

current instance.  
(Inherited from  
[Object](#).)

---

 <a href="#">InLock(Action)</a>	Performs an action from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLock(ActionTState)</a>	Performs an action from within a lock, passing in the current state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLock(FuncTState, TState)</a>	Performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLockTResult(FuncTResult)</a>	Performs a function from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLockAsync(Action)</a>	Asynchronously performs an action from within a lock (Inherited from

[StateLockerTLocker](#),  
[TState](#).)

---

≡♥ [InLockAsync\(Action<TState>\)](#)

Asynchronously performs an action from within a lock, passing in the current state (Inherited from [StateLockerTLocker](#), [TState](#).)

---

≡♥ [InLockAsync\(Func<TState, TState>\)](#)

Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from [StateLockerTLocker](#), [TState](#).)

---

≡♥ [InLockAsync<TResult>\(Func<TResult>\)](#)

Asynchronously performs a function from within a lock (Inherited from [StateLockerTLocker](#), [TState](#).)

---

💡 [MemberwiseClone](#)

Creates a shallow copy of the current [Object](#). (Inherited from [Object](#).)

---

≡♥ [ToString](#)

Returns a string that represents the

current object.  
(Inherited from  
[Object](#).)

---

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">InLockAsyncTType(Func&lt;T&gt;)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

[Top](#)

## ◀ See Also

Reference

[SemaphoreSlimLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SemaphoreSlimLockerTState Fields

The [SemaphoreSlimLockerTState](#) generic type exposes the following members.

## ► Fields

	Name	Description
	<a href="#">State</a>	The internal state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

[Top](#)

## ► See Also

[Reference](#)

[SemaphoreSlimLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLocker Class

Uses SpinLock to provide resource locking

## ► Inheritance Hierarchy

[SystemObject](#) [W.Threading.LockersSpinLocker](#)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public class SpinLocker : ILocker<SpinLock>,  
    ILocker
```

The [SpinLocker](#) type exposes the following members.

## ► Constructors

	Name	Description
	<a href="#">SpinLocker</a>	Initializes a new instance of the SpinLocker class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The SpinLock used to perform locks

[Top](#)

## ◀ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)

≡	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
≡	<a href="#">InLock(Action)</a>	Performs an action from within a SpinLock
≡	<a href="#">InLockTResult(FuncTResult)</a>	Performs a function from within a SpinLock
≡	<a href="#">InLockAsync(Action)</a>	Performs an action from within a SpinLock
≡	<a href="#">InLockAsyncTResult(FuncTResult)</a>	Performs a function from within a SpinLock
≡	<a href="#">Lock</a>	Locks the resource
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString</a>	Returns a

string that represents the current object.  
(Inherited from [Object](#).)



### [Unlock](#)

Unlocks the resource

[Top](#)

## ► Extension Methods

Name	Description
<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)
<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)

 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock

(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► Remarks

Can be overridden to provide additional functionality

## ► See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLocker Constructor

Initializes a new instance of the [SpinLocker](#) class

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public SpinLocker()
```

## ► See Also

Reference

[SpinLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLocker Properties

The [SpinLocker](#) type exposes the following members.

## ► Properties

Name	Description
 <a href="#">Locker</a>	The SpinLock used to perform locks

[Top](#)

## ► See Also

[Reference](#)

[SpinLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerLocker Property

The SpinLock used to perform locks

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public SpinLock Locker { get; }
```

Property Value

Type: [SpinLock](#)

Implements

[ILockerTLockerLocker](#)

## ► See Also

Reference

[SpinLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLocker Methods

The [SpinLocker](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default

hash  
function.  
(Inherited  
from [Object](#).)

 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">InLock(Action)</a>	Performs an action from within a SpinLock
 <a href="#">InLockTResult(FuncTResult)</a>	Performs a function from within a SpinLock
 <a href="#">InLockAsync(Action)</a>	Performs an action from within a SpinLock
 <a href="#">InLockAsyncTResult(FuncTResult)</a>	Performs a function from within a SpinLock
 <a href="#">Lock</a>	Locks the resource
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current

[Object.](#)  
(Inherited  
from [Object.](#))

 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object.</a> )
 <a href="#">Unlock</a>	Unlocks the resource

[Top](#)

## ▪ Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded.

		Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
⬇️	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false

(Defined by  
[PropertyHostExtensions](#).)



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► See Also

[Reference](#)

[SpinLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerInLock Method

## ► Overload List

Name	Description
 <a href="#">InLock(Action)</a>	Performs an action from within a SpinLock
 <a href="#">InLockTResult(FuncTResult)</a>	Performs a function from within a SpinLock

[Top](#)

## ► See Also

Reference

[SpinLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerInLock Method (Action)

Performs an action from within a SpinLock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public void InLock(  
    Action action  
)
```

[Copy](#)

### Parameters

*action*

Type: [SystemAction](#)

The action to run

### Implements

[ILockerInLock\(Action\)](#)

## ► See Also

Reference

[SpinLocker Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)



## Tungsten

W<sup>74</sup>

# SpinLockerInLockTResult Method (FuncTResult)

Performs a function from within a SpinLock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public TResult InLock<TResult>(  
    Func<TResult> func  
)
```

[Copy](#)

## Parameters

*func*

Type: [SystemFuncTResult](#)

The function to run

## Type Parameters

*TResult*

The type of return value

## Return Value

Type: [TResult](#)

The result of the function call (a value of type TValue)

## Implements

[ILockerInLockTResult\(FuncTResult\)](#)

## ▲ See Also

Reference

[SpinLocker Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerInLockAsync Method

## ▪ Overload List

Name	Description
 <a href="#">InLockAsync(Action)</a>	Performs an action from within a SpinLock
 <a href="#">InLockAsyncTResult(FuncTResult)</a>	Performs a function from within a SpinLock

[Top](#)

## ▪ See Also

Reference

[SpinLocker Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerInLockAsync Method (Action)

Performs an action from within a SpinLock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public Task InLockAsync(  
    Action action  
)
```

[Copy](#)

### Parameters

*action*

Type: [SystemAction](#)

The action to run

### Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:W.Threading.Lockers.SpinLocker.InLockAsync(System.Action)"]

### Implements

[ILockerInLockAsync\(Action\)](#)

## ► See Also

### Reference

[SpinLocker Class](#)

InLockAsync Overload  
W.Threading.Lockers Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerInLockAsync<TResult> Method (Func<TResult>)

Performs a function from within a SpinLock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public Task<TResult> InLockAsync<TResult>(
    Func<TResult> func
)
```

### Parameters

*func*

Type: [System.Func<TResult>](#)

The function to run

### Type Parameters

*TResult*

The type of return value

### Return Value

Type: [Task<TResult>](#)

The result of the function call (a value of type TValue)

### Implements

[ILockerInLockAsync<TResult>\(Func<TResult>\)](#)

## ▲ See Also

Reference

[SpinLocker Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerLock Method

Locks the resource

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public void Lock(  
    ref bool lockTaken  
)
```

### Parameters

*lockTaken*

Type: [SystemBoolean](#)

[Missing <param name="lockTaken"/> documentation for  
"M:W.Threading.Lockers.SpinLocker.Lock(System.Boolean@)"]

## ► See Also

Reference

[SpinLocker Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# SpinLockerUnlock Method

Unlocks the resource

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public void Unlock(  
    bool useMemberBarrier = false  
)
```

## Parameters

### *useMemberBarrier* (Optional)

Type: [SystemBoolean](#)

[Missing <param name="useMemberBarrier"/> documentation for  
"M:W.Threading.Lockers.SpinLocker.Unlock(System.Boolean)"]

## ► See Also

### Reference

[SpinLocker Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# SpinLockerTState Class

Extends SpinLocker with an internal state variable

## ► Inheritance Hierarchy

SystemObject [W.Threading.LockersStateLockerSpinLocker](#), *TState*  
[W.Threading.LockersSpinLockerTState](#)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public class SpinLocker<TState> : StateLocker<Sp
```

### Type Parameters

*TState*

The state Type

The [SpinLockerTState](#) type exposes the following members.

## ► Constructors

	Name	Description
	<a href="#">SpinLockerTState</a>	Initializes a new instance of the <a href="#">SpinLockerTState</a> class

[Top](#)

## Properties

	Name	Description
	<a href="#">Locker</a>	The locking mechanism (SpinLock, Monitor, SemaphoreSlim, ReaderWriterLock) (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

[Top](#)

## Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)

[Object.\)](#)

---

 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object.</a> )
 <a href="#">InLock(Action)</a>	Performs an action from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLock(Action TState)</a>	Performs an action from within a lock, passing in the current state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLock(Func TState, TState)</a>	Performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLockTResult(Func TResult)</a>	Performs a function from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
 <a href="#">InLockAsync(Action)</a>	Asynchronously

performs an action from within a lock  
(Inherited from [StateLockerTLocker](#), [TState](#).)

---

≡ [InLockAsync\(Action<TState>\)](#)

Asynchronously performs an action from within a lock, passing in the current state  
(Inherited from [StateLockerTLocker](#), [TState](#).)

---

≡ [InLockAsync\(Func<TState, TState>\)](#)

Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result  
(Inherited from [StateLockerTLocker](#), [TState](#).)

---

≡ [InLockAsync<TResult>\(Func<TResult>\)](#)

Asynchronously performs a function from within a lock  
(Inherited from [StateLockerTLocker](#), [TState](#).)

---

💡 [MemberwiseClone](#)

Creates a shallow copy of the current [Object](#).  
(Inherited from [Object](#).)

---

 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)
--	---

---

[Top](#)

## Fields

Name	Description
 <a href="#">State</a>	The internal state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

---

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)

---

	<a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty

---

and sets it's IsDirty flag to  
false  
(Defined by  
[PropertyHostExtensions](#).)

---



### [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► Remarks

Same as StateLocker<SpinLocker;, TState>

## ► See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerTState Constructor

Initializes a new instance of the [SpinLockerTState](#) class

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public SpinLocker()
```

## ► See Also

Reference

[SpinLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerTState Properties

The [SpinLockerTState](#) generic type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The locking mechanism (SpinLock, Monitor, SemaphoreSlim, ReaderWriterLock) (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

[Top](#)

## ► See Also

Reference

[SpinLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerTState Methods

The [SpinLockerTState](#) generic type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 <a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)

---

≡	<a href="#">InLock(Action)</a>	Performs an action from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡	<a href="#">InLock(ActionTState)</a>	Performs an action from within a lock, passing in the current state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡	<a href="#">InLock(FuncTState, TState)</a>	Performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡	<a href="#">InLockTResult(FuncTResult)</a>	Performs a function from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡	<a href="#">InLockAsync(Action)</a>	Asynchronously performs an action from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

---

≡♪	<a href="#">InLockAsync(Action&lt;TState&gt;)</a>	Asynchronously performs an action from within a lock, passing in the current state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡♪	<a href="#">InLockAsync(Func&lt;TState, TState&gt;)</a>	Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
≡♪	<a href="#">InLockAsync&lt;TResult&gt;(Func&lt;TResult&gt;)</a>	Asynchronously performs a function from within a lock (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)
♪	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
≡♪	<a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

---

[Top](#)

## Extension Methods

Name	Description
  <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor

`lock`  
(Defined by  
[MonitorExtensions](#).)

---

 <a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
 <a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions</a> .)
 <a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions</a> .)

---

[Top](#)

## See Also

[Reference](#)

[SpinLockerTState Class](#)

## [W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockerTState Fields

The [SpinLockerTState](#) generic type exposes the following members.

## ► Fields

Name	Description
 <a href="#">State</a>	The internal state (Inherited from <a href="#">StateLockerTLocker</a> , <a href="#">TState</a> .)

[Top](#)

## ► See Also

Reference

[SpinLockerTState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockExtensions Class

Extensions to simplify locking with SpinLock

## ► Inheritance Hierarchy

SystemObject W.Threading.LockersSpinLockExtensions

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static class SpinLockExtensions
```

## ► Methods

	Name	Description
≡  S	<a href="#">InLock(SpinLock, Action)</a>	Performs the action in a lock
≡  S	<a href="#">InLockTType(SpinLock, FuncTType)</a>	Performs the function in a lock
≡  S	<a href="#">InLockAsync(SpinLock, Action)</a>	Asynchronously performs the action in a lock
≡  S	<a href="#">InLockAsyncTType(SpinLock, FuncTType)</a>	Asynchronously performs the function in a lock

[Top](#)

## ◀ See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockExtensions Methods

## ↳ Methods

	Name	Description
≡  S	<a href="#">InLock(SpinLock, Action)</a>	Performs the action in a lock
≡  S	<a href="#">InLockTType(SpinLock, FuncTType)</a>	Performs the function in a lock
≡  S	<a href="#">InLockAsync(SpinLock, Action)</a>	Asynchronously performs the action in a lock
≡  S	<a href="#">InLockAsyncTType(SpinLock, FuncTType)</a>	Asynchronously performs the function in a lock

[Top](#)

## ↳ See Also

[Reference](#)

[SpinLockExtensions Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# SpinLockExtensionsInLock

## Method

### ▪ Overload List

	Name	Description
 	<a href="#">InLock(SpinLock, Action)</a>	Performs the action in a lock
 	<a href="#">InLockTType(SpinLock, FuncTType)</a>	Performs the function in a lock

[Top](#)

### ▪ See Also

Reference

[SpinLockExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockExtensionsInLock Method (SpinLock, Action)

Performs the action in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static void InLock(  
    this SpinLock this,  
    Action action  
)
```

[Copy](#)

## Parameters

*this*

Type: [System.ThreadingSpinLock](#)

The SpinLock to provide resource locking

*action*

Type: [SystemAction](#)

The action to perform

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SpinLock](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ▲ See Also

Reference

[SpinLockExtensions Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockExtensionsInLock<TType> Method (SpinLock, Func<TType>)

Performs the function in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public static TType InLock<TType>(  
    this SpinLock this,  
    Func<TType> func  
)
```

## Parameters

*this*

Type: [System.ThreadingSpinLock](#)

The SpinLock to provide resource locking

*func*

Type: [SystemFunc<TType>](#)

The function to perform

## Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for  
"M:W.Threading.Lockers.SpinLockExtensions.InLock`1(System.Threading.SpinL

## Return Value

Type: *TType*

[Missing <returns> documentation for  
"M:W.Threading.Lockers.SpinLockExtensions.InLock`1(System.Threading.SpinLock,S

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SpinLock](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[SpinLockExtensions Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockExtensionsInLockAsync Method

## ▪ Overload List

	Name	Description
 	<a href="#">InLockAsync(SpinLock, Action)</a>	Asynchronously performs the action in a lock
 	<a href="#">InLockAsyncTType(SpinLock, FuncTType)</a>	Asynchronously performs the function in a lock

[Top](#)

## ▪ See Also

Reference

[SpinLockExtensions Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockExtensions.InLockAsync Method (SpinLock, Action)

Asynchronously performs the action in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public static Task InLockAsync(  
    this SpinLock this,  
    Action action  
)
```

[Copy](#)

## Parameters

*this*

Type: [System.ThreadingSpinLock](#)

The SpinLock to provide resource locking

*action*

Type: [SystemAction](#)

The action to perform

## Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:W.Threading.Lockers.SpinLockExtensions.InLockAsync(System.Threading.SpinLoc

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SpinLock](#). When you use instance

method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[SpinLockExtensions Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# SpinLockExtensionsInLockAsync<TType> Method (SpinLock, Func<TType>)

Asynchronously performs the function in a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public static Task<TType> InLockAsync<TType>(
    this SpinLock this,
    Func<TType> func
)
```

### Parameters

*this*

Type: [System.ThreadingSpinLock](#)

The SpinLock to provide resource locking

*func*

Type: [SystemFunc<TType>](#)

The function to perform

### Type Parameters

*TType*

[Missing <typeparam name="TType"/> documentation for "M:W.Threading.Lockers.SpinLockExtensions.InLockAsync`1(System.Threading

### Return Value

Type: [Task<TType>](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.SpinLockExtensions.InLockAsync``1(System.Threading.Spinl

## Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SpinLock](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## See Also

### Reference

[SpinLockExtensions Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# StateAssignmentDelegate<TState> Delegate

Delegate which can be used to assign a new value to the internal state

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public delegate TState StateAssignmentDelegate<TState>
    TState state
)
```

### Parameters

*state*

Type: [TState](#)

The current state

### Type Parameters

*TState*

[Missing <typeparam name="TState"/> documentation for  
"T:W.Threading.Lockers.StateAssignmentDelegate`1"]

### Return Value

Type: [TState](#)

The new value for the internal state

## ▲ See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# StateLocker<TLocker, TState> Class

Extends a locker (SpinLocker, MonitorLocker, ReaderWriterLocker, SemaphoreSlimLocker) with an internal state value

## ► Inheritance Hierarchy

```
SystemObject W.Threading.LockersStateLocker<TLocker, TState>
W.Threading.LockersMonitorLockerTState
W.Threading.LockersSemaphoreSlimLockerTState
W.Threading.LockersSpinLockerTState
```

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public abstract class StateLocker<TLocker, TState>
    : ILocker<TLocker>, ILocker
    where TLocker : new(), ILocker
```

### Type Parameters

#### *TLocker*

The Type of Locker to extend

#### *TState*

The Type of the internal state value

The `StateLocker<TLocker, TState>` type exposes the following members.

## ► Constructors

	Name	Description
	<a href="#">StateLockerTLocker(TState)</a>	Initializes a new instance of the <code>StateLockerTLocker</code> , <code>TState</code> class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The locking mechanism (SpinLock, Monitor, SemaphoreSlim, ReaderWriterLock)

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Equals</a>	Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code> .)
	<a href="#">Finalize</a>	Allows an object to try to free resources and perform other cleanup

operations before it is reclaimed by garbage collection.  
(Inherited from [Object](#).)

 <a href="#">GetHashCode</a>	Serves as the default hash function. (Inherited from <a href="#">Object</a> .)
 <a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
 <a href="#">InLock(Action)</a>	Performs an action from within a lock
 <a href="#">InLock(ActionTState)</a>	Performs an action from within a lock, passing in the current state
 <a href="#">InLock(FuncTState, TState)</a>	Performs a function from within a lock, passing in the current state and assigning the state to the function result

≡	<a href="#">InLockTResult(FuncTResult)</a>	Performs a function from within a lock
≡	<a href="#">InLockAsync(Action)</a>	Asynchronously performs an action from within a lock
≡	<a href="#">InLockAsync(ActionTState)</a>	Asynchronously performs an action from within a lock, passing in the current state
≡	<a href="#">InLockAsync(FuncTState, TState)</a>	Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result
≡	<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchronously performs a function from within a lock
💡	<a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)



## ToString

Returns a string that represents the current object.  
(Inherited from [Object](#).)

[Top](#)

## Fields

	Name	Description
	<a href="#">State</a>	The internal state

[Top](#)

## Extension Methods

	Name	Description
	<a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions</a> .)
	<a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions</a> .)



	InLock(Action)	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	InLockTType(FuncTType)	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	InLockAsync(Action)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	InLockAsyncTType(FuncTType)	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	IsDirty	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions</a> .)
⬇️	Lock	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions</a> .)
⬇️	MarkAsClean	Scans each field and property of type IProperty and sets its IsDirty flag to

false  
(Defined by  
[PropertyHostExtensions](#).)

---



## [Unlock](#)

Performs a Monitor  
unlock  
(Defined by  
[MonitorExtensions](#).)

---

[Top](#)

## ► Remarks

This class adds the state functionality by wrapping the TLocker and re-implementing the ILocker interface

## ► See Also

Reference

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *StateLockerTLocker, TState* Constructor

Initializes a new instance of the [StateLockerTLocker, TState](#) class

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
protected StateLocker()
```

## ► See Also

Reference

[StateLockerTLocker, TState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *StateLockerTLocker, TState* Properties

The [StateLockerTLocker, TState](#) generic type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Locker</a>	The locking mechanism (SpinLock, Monitor, SemaphoreSlim, ReaderWriterLock)

[Top](#)

## ► See Also

Reference

[StateLockerTLocker, TState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# StateLockerTLocker, TStateLocker Property

The locking mechanism (SpinLock, Monitor, SemaphoreSlim, ReaderWriterLock)

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

```
public TLocker Locker { get; }
```

[Copy](#)

Property Value

Type: [TLocker](#)

Implements

[ILocker](#)[TLocker](#)[Locker](#)

## ► See Also

Reference

[StateLockerTLocker, TState Class](#)

[W.Threading.Lockers Namespace](#)

## Tungsten

W<sup>74</sup>

# StateLocker`TLocker`, `TState` Methods

The `StateLocker``TLocker`, `TState` generic type exposes the following members.

## ▲ Methods

Name	Description
 Equals	Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a> .)
 Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
 GetHashCode	Serves as the default hash function.

		(Inherited from Object.)
≡♥	<a href="#">GetType</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from Object.)
≡♥	<a href="#">InLock(Action)</a>	Performs an action from within a lock
≡♥	<a href="#">InLock(Action&lt;TState&gt;)</a>	Performs an action from within a lock, passing in the current state
≡♥	<a href="#">InLock(Func&lt;TState, TState&gt;)</a>	Performs a function from within a lock, passing in the current state and assigning the state to the function result
≡♥	<a href="#">InLockTResult(Func&lt;TResult&gt;)</a>	Performs a function from within a lock
≡♥	<a href="#">InLockAsync(Action)</a>	Asynchronously performs an action from within a lock
≡♥	<a href="#">InLockAsync(Action&lt;TState&gt;)</a>	Asynchronously

performs an action from within a lock, passing in the current state

 <a href="#">InLockAsync(FuncTState, TState)</a>	Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result
 <a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchronously performs a function from within a lock
 <a href="#">MemberwiseClone</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
 <a href="#">ToString</a>	Returns a string that represents the current object. (Inherited from <a href="#">Object</a> .)

[Top](#)

## Extension Methods

Name	Description
 <a href="#">AsTType</a>	Use Generic syntax for the as operator. (Defined by <a href="#">AsExtensions.</a> )
 <a href="#">InitializeProperties</a>	Scans the fields and properties of "owner" and sets the member's Owner property to "owner". This method should be called in the constructor of any class which has IOwnedProperty members (Defined by <a href="#">PropertyHostExtensions.</a> )
 <a href="#">InLock(Action)</a>	Overloaded. Performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockTType(FuncTType)</a>	Overloaded. Performs the function in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
 <a href="#">InLockAsync(Action)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by

[MonitorExtensions.](#))

---

	<a href="#">InLockAsyncTType(FuncTType)</a>	Overloaded. Asynchronously performs the action in a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">IsDirty</a>	Scans the IsDirty value of each field and property of type IProperty (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Lock</a>	Performs a Monitor lock (Defined by <a href="#">MonitorExtensions.</a> )
	<a href="#">MarkAsClean</a>	Scans each field and property of type IProperty and sets its IsDirty flag to false (Defined by <a href="#">PropertyHostExtensions.</a> )
	<a href="#">Unlock</a>	Performs a Monitor unlock (Defined by <a href="#">MonitorExtensions.</a> )

---

[Top](#)

## ▲ See Also

### Reference

[StateLockerTLocker, TState Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# **StateLockerTLocker,** **TStateInLock Method**

## ▲ Overload List

Name	Description
 <a href="#">InLock(Action)</a>	Performs an action from within a lock
 <a href="#">InLock(ActionTState)</a>	Performs an action from within a lock, passing in the current state
 <a href="#">InLockTResult(FuncTResult)</a>	Performs a function from within a lock
 <a href="#">InLock(FuncTState, TState)</a>	Performs a function from within a lock, passing in the current state and assigning the state to the function result

[Top](#)

## ▲ See Also

### Reference

[StateLockerTLocker, TState Class](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# StateLockerTLocker, TStateInLock Method (Action)

Performs an action from within a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C#  
public void InLock(  
    Action action  
)
```

[Copy](#)

### Parameters

*action*

Type: [SystemAction](#)

The action to run

Implements

[ILockerInLock\(Action\)](#)

## ► See Also

Reference

[StateLockerTLocker, TState Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)



## Tungsten

W<sup>74</sup>

# *StateLockerTLocker*, *TStateInLock* Method (Action*TState*)

Performs an action from within a lock, passing in the current state

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public void InLock(
    Action<TState> action
)
```

### Parameters

*action*

Type: [SystemActionTState](#)

The action to run

### Implements

[IStateLockerTLocker](#), [TStateInLock\(ActionTState\)](#)

## ► See Also

### Reference

[StateLockerTLocker, TState Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# StateLocker $T$ Locker, $T$ StateInLock $T$ Result Method (Func $T$ Result)

Performs a function from within a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public TResult InLock<TResult>(  
    Func<TResult> func  
)
```

## Parameters

*func*

Type: [SystemFunc<TResult>](#)

The function to run

## Type Parameters

*TResult*

[Missing <typeparam name="TResult"/> documentation for  
"M:W.Threading.Lockers.StateLocker`2.InLock`1(System.Func{`0})"]

## Return Value

Type: *TResult*

[Missing <returns> documentation for  
"M:W.Threading.Lockers.StateLocker`2.InLock`1(System.Func{`0})"]

Implements  
[ILockerInLockTResult\(FuncTResult\)](#)

## ► See Also

Reference

[StateLockerTLocker, TState Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *StateLocker* *TLocker*, *TStateInLock* Method (*FuncTState*, *TState*)

Performs a function from within a lock, passing in the current state and assigning the state to the function result

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public TState InLock(  
    Func<TState, TState> func  
)
```

### Parameters

*func*

Type: [SystemFuncTState, TState](#)

The function to run

### Return Value

Type: [TState](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.StateLocker`2.InLock(System.Func`1,`1)"]

### Implements

[IStateLockerTLocker](#), [TStateInLock\(FuncTState, TState\)](#)

## ▲ See Also

Reference

[StateLockerTLocker, TState Class](#)

[InLock Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# **StateLocker***TLocker*, *TStateInLockAsync* Method

## ▪ Overload List

Name	Description
<a href="#">InLockAsync(Action)</a>	Asynchronously performs an action from within a lock
<a href="#">InLockAsync(ActionTState)</a>	Asynchronously performs an action from within a lock, passing in the current state
<a href="#">InLockAsyncTResult(FuncTResult)</a>	Asynchronously performs a function from within a lock
<a href="#">InLockAsync(FuncTState, TState)</a>	Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result

[Top](#)

## ◀ See Also

Reference

[StateLockerTLocker, TState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# StateLocker *TLocker*, *TStateInLockAsync* Method (Action)

Asynchronously performs an action from within a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

[Copy](#)

```
public Task InLockAsync(  
    Action action  
)
```

### Parameters

*action*

Type: [SystemAction](#)

The action to run

### Return Value

Type: [Task](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.StateLocker`2.InLockAsync(System.Action)"]

### Implements

[ILockerInLockAsync\(Action\)](#)

## ► See Also

## Reference

[StateLockerTLocker, TState Class](#)  
[InLockAsync Overload](#)  
[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *StateLocker* *TLocker*, *TStateInLockAsync* Method (Action*TState*)

Asynchronously performs an action from within a lock, passing in the current state

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public Task InLockAsync(  
    Action<TState> action  
)
```

### Parameters

*action*

Type: [SystemActionTState](#)

The action to run

### Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:W.Threading.Lockers.StateLocker`2.InLockAsync(System.Action{`1})"]

### Implements

[IStateLockerTLocker](#), [TStateInLockAsync\(ActionTState\)](#)

## ▲ See Also

Reference

[StateLockerTLocker, TState Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *StateLocker* *TLocker*, *TStateInLockAsync* *TResult* Method (*Func* *TResult*)

Asynchronously performs a function from within a lock

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

```
C# Copy
public Task<TResult> InLockAsync<TResult>(
    Func<TResult> func
)
```

## Parameters

*func*

Type: [SystemFunc](#) *TResult*

The function to run

## Type Parameters

*TResult*

[Missing <typeparam name="TResult"/> documentation for  
"M:W.Threading.Lockers.StateLocker`2.InLockAsync`1(System.Func{`0})"]

## Return Value

Type: [Task](#) *TResult*

[Missing <returns> documentation for  
"M:W.Threading.Lockers.StateLocker`2.InLockAsync`1(System.Func{`0})"]

Implements  
[ILockerInLockAsyncResult\(Func<TResult>\)](#)

## See Also

### Reference

[StateLockerTLocker, TState Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *StateLocker* *TLocker*, *TStateInLockAsync* Method (*FuncTState*, *TState*)

Asynchronously performs a function from within a lock, passing in the current state and assigning the state to the function result

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
public Task<TState> InLockAsync(  
    Func<TState, TState> func  
)
```

## Parameters

*func*

Type: [SystemFuncTState, TState](#)

The function to run

## Return Value

Type: [TaskTState](#)

[Missing <returns> documentation for  
"M:W.Threading.Lockers.StateLocker`2.InLockAsync(System.Func{`1,`1})"]

## Implements

[IStateLockerTLocker](#), [TStateInLockAsync\(FuncTState, TState\)](#)

## ▲ See Also

Reference

[StateLockerTLocker, TState Class](#)

[InLockAsync Overload](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# *StateLocker**TLocker*, *TState* Fields

The [StateLocker](#)[TLocker](#), [TState](#) generic type exposes the following members.

## ► Fields

	Name	Description
	<a href="#">State</a>	The internal state

[Top](#)

## ► See Also

Reference

[StateLocker](#)[TLocker](#), [TState](#) Class  
[W.Threading.Lockers](#) Namespace

---

Copyright @ 2018 Jordan Duerksen

## Tungsten

W<sup>74</sup>

# StateLockerTLocker, TState State Field

The internal state

**Namespace:** [W.Threading.Lockers](#)

**Assembly:** Tungsten.Threading.Lockers (in  
Tungsten.Threading.Lockers.dll) Version: 2.0.1

## ► Syntax

C#

Copy

```
protected TState State
```

Field Value

Type: [TState](#)

## ► See Also

Reference

[StateLockerTLocker, TState Class](#)

[W.Threading.Lockers Namespace](#)

---

Copyright @ 2018 Jordan Duerksen