SampleClassLibrary Reference

Table of Contents

SampleClassLibrary Help

Overview

This is documentation for our sample SampleClassLibrary class library. It was generated with VSdocman to show its capabilities. With VSdocman you can comment and generate documentation for your .NET code.

This topic and other topics and chapters were created as custom topics. The API topics in SampleClassLibrary Reference chapter were generated from the source code and XML comments.

See the API documentation for the main MainClass class.

See also string class.

See Also

MainClass class[23]

SampleClassLibrary Usage

Basic Functionality

Here we describe basic functionality of SampleClassLibrary library.

See Also

Advanced Usage[5]

Advanced

To fully use the SampleClassLibrary, you need to understand some additional concepts. They are described below.

Remarks

Our remarks here.

See Also

Basic Functionality[5]

SampleClassLibrary Reference

Here you can find class reference for SampleClassLibrary.

See the API documentation for its main SampleClassLibrary namespace.

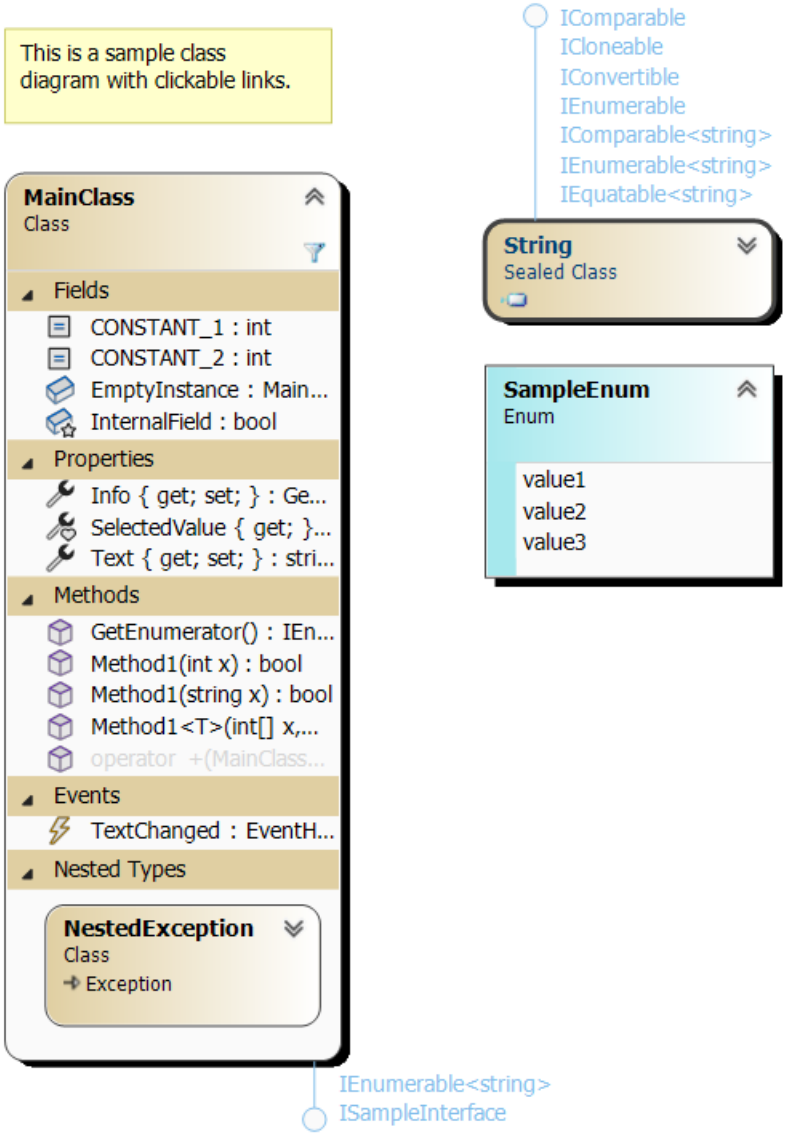
Namespaces

SampleClassLibrary Namespace

This is main namespace. See an example of a class diagram in the Remarks.

Remarks

You can navigate to code elements using clickable class diagram below. The diagram can be placed anywhere, e.g. in summary or Remarks section or in its own section as shown here.



Classes

ClassWithGenericMethods[8], DependencyAndAttachedProps[10], GenericClass2<T>[15], GenericClass<T1, T2>[16], MainClass[23], TextChangedEventArgs[38]

Interfaces

ISampleInterface[41]

Structures

GeneralInfo[43]

Enumerations

SampleEnum[45]

ClassWithGenericMethods Class

A non-generic class with generic methods.

System.Object
  SampleClassLibrary.ClassWithGenericMethods

VB

```
Public Class ClassWithGenericMethods
```

C#

```
public class ClassWithGenericMethods
```

```
[F#]
type ClassWithGenericMethods = class end
```

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Methods

AddHierarchy[8], Equals (inherited from Object), Finalize (inherited from Object), GetHashCode (inherited from Object), GetType (inherited from Object), MemberwiseClone (inherited from Object), Mm5<T3>[9], ToString (inherited from Object)

ClassWithGenericMethods.AddHierarchy Method

A method with a parameter of constructed type (bound generic type).

VB

```
Public Sub AddHierarchy( _
   ByVal objCol As ReadOnlyCollection(Of ClassWithGenericMethods) _
)
```

C#

```
public void AddHierarchy(
   ReadOnlyCollection<ClassWithGenericMethods> objCol
```

```
)
```

```
[F#]
member AddHierarchy :
  objCol:ReadOnlyCollection<ClassWithGenericMethods>
-> unit
```

Parameters

objCol

A parameter of constructed type (bound generic type)

Remarks

The link to this member has the following syntax:

<see cref="M:SampleClassLibrary.ClassWithGenericMethods.AddHierarchy(System.Collections.ObjectModel.ReadOnlyCollection{SampleClassLibrary.ClassWithGenericMethods})"/>

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: ClassWithGenericMethods[8]

ClassWithGenericMethods.Mm5<T3> Method

A generic method with a parameter of constructed type (bound generic type).

VB

```
Public Sub Mm5(Of  _
   T3)( _
   ByVal x As List(Of Integer) _
)
```

C#

```
public void Mm5<T3>(
   List<int> x
)
```

```
[F#]
member Mm5<'T3> :
  x:List<int>
-> unit
```

Type Parameters

T3

Method's own type parameter.

Parameters

x

A parameter of constructed type (bound generic type)

Remarks

The link to this member has the following syntax:

<see cref="M:SampleClassLibrary.ClassWithGenericMethods.Mm5``1(System.Collections.Generic.List{System.Int32})"/>

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: ClassWithGenericMethods[8]

DependencyAndAttachedProps Class

This class serves as a container for demonstrating dependency and attached properties.

System.Object
  System.Windows.Threading.DispatcherObject
    System.Windows.DependencyObject
      SampleClassLibrary.DependencyAndAttachedProps

VB

```
Public Class DependencyAndAttachedProps
    Inherits DependencyObject
```

C#

```
public class DependencyAndAttachedProps : DependencyObject
```

```
 [F#]
 type DependencyAndAttachedProps = class
   inherit DependencyObject
end
```

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Properties

DependencyObjectType (inherited from DependencyObject), Dispatcher (inherited from DispatcherObject), IsSealed (inherited from DependencyObject), LabelDp[12]

Attached Properties

BlinkIntervalAp[11]

Methods

CheckAccess (inherited from DispatcherObject), ClearValue (inherited from DependencyObject), CoerceValue (inherited from DependencyObject), Equals (inherited from DependencyObject), Finalize (inherited from Object), GetBlinkIntervalAp[12], GetHashCode (inherited from DependencyObject), GetLocalValueEnumerator (inherited from DependencyObject), GetType (inherited from Object), GetValue (inherited from DependencyObject), InvalidateProperty (inherited from DependencyObject), MemberwiseClone (inherited from Object), OnPropertyChanged (inherited from DependencyObject), ReadLocalValue (inherited from DependencyObject), SetBlinkIntervalAp[13], SetCurrentValue (inherited from DependencyObject), SetValue (inherited from DependencyObject), ShouldSerializeProperty (inherited from DependencyObject), ToString (inherited from Object), VerifyAccess (inherited from DispatcherObject)

Fields

BlinkIntervalApProperty[14], LabelDpProperty[14]

DependencyAndAttachedProps.BlinkIntervalAp Property

A sample attached property. Gets or sets a blinking interval in ms.

VB

```
See GetBlinkIntervalAp, SetBlinkIntervalAp
```

C#

```
See GetBlinkIntervalAp, SetBlinkIntervalAp
```

```
[F#]
See GetBlinkIntervalAp, SetBlinkIntervalAp
```

Remarks

If you have pure **attached property** (that is not also a dependency property with a "wrapper" property), there is no place in the code you can add the XML comments. There's only attached property identifier field and getter/setter methods. All of them have their own XML comments

used in their own documentation topics. VSdocman solves this problem by taking XML comments from **external XML files**. You need to create these files manually. See *Using VSdocman - Tips & Tricks - Documenting Attached Properties* in VSdocman help for more information.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: DependencyAndAttachedProps[10]

DependencyAndAttachedProps.LabelDp Property

A sample dependency property. Gets or sets a textual label.

VB

```
Public Property LabelDp() As String
```

C#

```
public string LabelDp {get; set;}
```

```
[F#]
member LabelDp : string with get, set
```

Remarks

Since **dependency properties** have their "wrapper" property, you can document them like normal properties. You just place your XML comments before property code.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: DependencyAndAttachedProps[10]

DependencyAndAttachedProps.GetBlinkIntervalAp Method

A getter for BlinkIntervalAp attached property.

VB

```
Public Shared Function GetBlinkIntervalAp( _
   ByVal element As UIElement _
```

```
) As Integer
```

C#

```
public static int GetBlinkIntervalAp(
   UIElement element
)
```

```
[F#]
static member GetBlinkIntervalAp :
  element:UIElement
-> int
```

Parameters

element

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: DependencyAndAttachedProps[10]

DependencyAndAttachedProps.SetBlinkIntervalAp Method

A setter for BlinkIntervalAp attached property.

VB

```
Public Shared Sub SetBlinkIntervalAp( _
   ByVal element As UIElement, _
   ByVal value As Integer _
)
```

C#

```
public static void SetBlinkIntervalAp(
   UIElement element,
   int value
)
```

```
[F#]
static member SetBlinkIntervalAp :
  element:UIElement *
```

```
  value:int
-> unit
```

Parameters

element

value

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: DependencyAndAttachedProps[10]

BlinkIntervalApProperty Field

The identifier field for BlinkIntervalAp attached property.

VB

```
Public Shared Shadows ReadOnly BlinkIntervalApProperty As DependencyProperty
```

C#

```
new public static readonly DependencyProperty BlinkIntervalApProperty
```

```
[F#]
static val  BlinkIntervalApProperty : DependencyProperty
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: DependencyAndAttachedProps[10]

LabelDpProperty Field

The identifier field for LabelDp dependency property.

VB

```
Public Shared Shadows ReadOnly LabelDpProperty As DependencyProperty
```

C#

```
new public static readonly DependencyProperty LabelDpProperty
```

```
[F#]
static val  LabelDpProperty : DependencyProperty
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: DependencyAndAttachedProps[10]

GenericClass2<T> Class

A simple generic class (unbound generic type).

System.Object
  SampleClassLibrary.GenericClass2<T>

VB

```
Public Class GenericClass2(Of  _
   T)
```

C#

```
public class GenericClass2<T>
```

```
[F#]
type GenericClass2<'T> = class end
```

Type Parameters

T

Remarks

The link to this member has the following syntax:

<see cref="T:SampleClassLibrary.GenericClass2`1"/>

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Methods

Equals (inherited from Object), Finalize (inherited from Object), GetHashCode (inherited from Object), GetType (inherited from Object), MemberwiseClone (inherited from Object), ToString (inherited from Object)

GenericClass<T1, T2> Class

More complicated generic class (unbound generic type with constraint) with generic methods. It is used to demonstrate how to create cref links for generic members.

System.Object
  SampleClassLibrary.GenericClass<T1, T2>

VB

```
Public Class GenericClass(Of  _
   T1 As IComparer, _
   T2)
```

C#

```
public class GenericClass<T1, T2>
   where T1 : IComparer
```

```
 [F#]
 type GenericClass<'T1, 'T2
   when 'T1 :> IComparer> = class end
```

Type Parameters

T1

The first type parameter.

T2

The second type parameter.

Remarks

The link to this member has the following syntax:

<see cref="T:SampleClassLibrary.GenericClass`2"/>

Generics nomenclature

Generic types introduce and use the following terms:

- **Unbound Generic Type** - A definition of a class, structure, interface, procedure, or delegate for which you supply at least one data type when you declare it. An unbound generic type is not itself a type, and cannot be used as the type of a variable, argument or return value, or as a base type. The only construct in which an unbound generic type can be referenced is the typeof expression.
  public class List<T>
- **Type Parameter** - In an unbound generic type definition, a placeholder for a data type you supply when you declare the type.
  public class List<**T**>
- **Type Argument** - A specific data type that replaces a type parameter when you declare a constructed type from an unbound generic type.
  public class List<**int**>
- **Constraint** - A condition on a type parameter that restricts the type argument you can supply for it. A constraint can require that the type argument must implement a particular interface, be or inherit from a particular class, have an accessible parameterless constructor, or be a reference type or a value type. You can combine these constraints, but you can specify at most one class.
  public class genericClass **where T : System.Collections.IComparer**
- **Constructed Type (bound generic type)** - A class, structure, interface, procedure, or delegate declared from an unbound generic type by supplying type arguments for its type parameters.
  public class List<int>

Rules for the cref syntax:

- **Type parameters in unbound types** are replaced by **`number_of_params**
- **Type parameters in unbound non-types** (methods, props, ...) are replaced by **``number_of_params**
- **Type arguments** are enclosed inside **{ }**. Moreover, if type argument is a type parameter of containing class(es), it is replaced by its **index starting from zero** prepended by **`**. This index is agregated from outermost class, it isn't set to zero in nested generic class.
- **Constraints** are removed.

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Methods

Equals (inherited from Object), Finalize (inherited from Object), GetHashCode (inherited from Object), GetType (inherited from Object), MemberwiseClone (inherited from Object), Mm1[18], Mm2[18], Mm3[19], Mm4<T3>[20], ToString (inherited from Object)

Classes

NestedGenericClass<T3>[21]

GenericClass<T1, T2>.Mm1 Method

A method with a parameter of constructed type (bound generic type).

VB

```
Public Sub Mm1( _
   ByVal x As List(Of Integer)() _
)
```

C#

```
public void Mm1(
   List<int>[] x
)
```

```
[F#]
member Mm1 :
  x:List<int>[]
-> unit
```

Parameters

x

A parameter of constructed type (bound generic type)

Remarks

The link to this member has the following syntax:

<see cref="M:SampleClassLibrary.GenericClass`2.Mm1(System.Collections.Generic.List{System.Int32}[])"/>

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: GenericClass<T1, T2>[16]

GenericClass<T1, T2>.Mm2 Method

A method with a parameter of generic type from the owner class.

VB

```
Public Sub Mm2( _
   ByVal x As List(Of T1) _
)
```

C#

```
public void Mm2(
   List<T1> x
)
```

```
[F#]
member Mm2 :
  x:List<'T1>
-> unit
```

Parameters

x

A parameter of generic type from the owner class.

Remarks

The link to this member has the following syntax:

<see cref="M:SampleClassLibrary.GenericClass`2.Mm2(System.Collections.Generic.List{`0})"/>

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: GenericClass<T1, T2>[16]

GenericClass<T1, T2>.Mm3 Method

A method with a parameter of generic type from the owner class.

VB

```
Public Sub Mm3( _
   ByVal x As List(Of T2) _
)
```

C#

```
public void Mm3(
   List<T2> x
)
```

```
[F#]
member Mm3 :
  x:List<'T2>
```

```
-> unit
```

Parameters

x

A parameter of generic type from the owner class.

Remarks

The link to this member has the following syntax:

<see cref="M:SampleClassLibrary.GenericClass`2.Mm3(System.Collections.Generic.List{`1})"/>

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: GenericClass<T1, T2>[16]

GenericClass<T1, T2>.Mm4<T3> Method

A generic method with a parameter of generic type from the owner class.

VB

```
Public Sub Mm4(Of  _
   T3)( _
   ByVal x As List(Of T2) _
)
```

C#

```
public void Mm4<T3>(
   List<T2> x
)
```

```
[F#]
member Mm4<'T3> :
  x:List<'T2>
-> unit
```

Type Parameters

T3

Method's own type parameter.

Parameters

x

A parameter of generic type from the owner class.

Remarks

The link to this member has the following syntax:

<see cref="M:SampleClassLibrary.GenericClass`2.Mm4``1(System.Collections.Generic.List{`1})"/>

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: GenericClass<T1, T2>[16]

NestedGenericClass<T3> Class

A nested generic class (unbound generic type).

System.Object
SampleClassLibrary.GenericClass<T1, T2>.NestedGenericClass<T3>

VB

```
Public Class NestedGenericClass(Of _
   T3)
```

C#

```
public class NestedGenericClass<T3>
```

```
[F#]
type NestedGenericClass<'T3> = class end
```

Type Parameters

T3

Nested type parameter.

Remarks

The link to this member has the following syntax:

<see cref="T:SampleClassLibrary.GenericClass`2.NestedGenericClass`1"/>

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Methods

Equals (inherited from Object), Finalize (inherited from Object), GetHashCode (inherited from Object), GetType (inherited from Object), MemberwiseClone (inherited from Object), Mm1<T4>[22], ToString (inherited from Object)

See Also

Applies to: GenericClass<T1, T2>[16]

NestedGenericClass<T3>.Mm1<T4> Method

A generic method with parameters of generic types from direct owner class and from its parent class.

VB

```
Public Sub Mm1(Of  _
   T4)( _
   ByVal x As T1, _
   ByVal y As T2, _
   ByVal z As T3, _
   ByVal v As T4 _
)
```

C#

```
public void Mm1<T4>(
   T1 x,
   T2 y,
   T3 z,
   T4 v
)
```

```
[F#]
member Mm1<'T4> :
  x:'T1 *
  y:'T2 *
  z:'T3 *
  v:'T4
-> unit
```

Type Parameters

T4

Method's own type parameter.

Parameters

x

A parameter of generic type from parent class of direct owner class.

y

A parameter of generic type from parent class of direct owner class.

z

A parameter of generic type from direct owner class.

v

A parameter of generic type from this method.

Remarks

The link to this member has the following syntax:

<see cref="M:SampleClassLibrary.GenericClass`2.NestedGenericClass`1.Mm1``1(`0,`1,`2,``0)"/>

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: NestedGenericClass<T3>[21]

MainClass Class

The main class of SampleClassLibrary.

System.Object
  SampleClassLibrary.MainClass

VB

```
Public Class MainClass
    Implements IEnumerable(Of String), _
               ISampleInterface
```

C#

```
public class MainClass : IEnumerable<string>,
   ISampleInterface
```

```
 [F#]
 type MainClass = class
   interface IEnumerable<string>
   interface ISampleInterface
end
```

Remarks

Here we show how to use XML comments. As a thing of interest, this class has HelperFunctions.AsPrettyString extension method defined in HelperFunctions class.

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Thread Safety

Any static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Properties

Info[24], SelectedValue[25], Text[25]

Methods

Addition[27], Equals (inherited from Object), Finalize (inherited from Object), GetEnumerator[28], GetEnumerator1[29], GetHashCode (inherited from Object), GetType (inherited from Object), ISampleInterface.TestMethod[29], MemberwiseClone (inherited from Object), Method1[30], Method1<T>[31], System.Collections.IEnumerable.GetEnumerator[32], ToString (inherited from Object)

Events

TextChanged[33]

Classes

NestedException[34]

Fields

CONSTANT_1[37], CONSTANT_2[37], EmptyInstance[35], InternalField[35], _SelectedValue[36], _Text[36]

MainClass.Info Property

Gets or sets information about this object.

VB

```
Public Property Info() As GeneralInfo
```

C#

```
public GeneralInfo Info {get; set;}
```

```
[F#]
member Info : GeneralInfo with get, set
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

MainClass.SelectedValue Property

Gets the selected value.

VB

```
Friend ReadOnly Property SelectedValue() As SampleEnum
```

C#

```
internal SampleEnum SelectedValue {get;}
```

```
[F#]
member internal SelectedValue : SampleEnum with get
```

Property Value

Only allowed values.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

Our Text Property[25], SampleEnum[45]

MainClass.Text Property

Our sample property.

VB

```
Public Property Text() As String
```

C#

```
public string Text {get; set;}
```

```
[F#]
member Text : string with get, set
```

Property Value

Some nice text.

Remarks

This property is really interesting. We can use:

1. ***font formatting*** or `code`
2. bulleted or numbered lists
3. tables, pictures, links

Here is a table:

Column1	Column2
abc	123
def	456

Example

This is an example how to use **Text** and SelectedValue properties:

C#

```
try {
   if (this.Text != "hello") {
     return SelectedValue;
   }
 } catch (nestedException ex) {
   return SampleEnum.value1;
 }
```

VB

```
Try
      If Me.Text <> "hello" Then
          Return SelectedValue
      End If
 Catch ex As nestedException
      Return SampleEnum.value1
 End Try
```

Author

Peter Macej

Version

3.0

Revision

29

Source code

```
public string Text
{
    get { return _Text; }
    set
    {
```

```
        if (value != _Text)
        {
            string oldVal = _Text;
            _Text = value;
            if (TextChanged != null)
            {
                TextChanged(this, new TextChangedEventArgs(oldVal, value));
            }
        }
    }
}
```

.NET Framework Security

PermissionSet

Everyone can access this member.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

Another interesting property[25]

MainClass.Addition Method

Returns an instance whose Text property is a concatenation of Text properties of *c1* and *c2*.

VB

```
Public Shared Operator +( _
   ByVal c1 As MainClass, _
   ByVal c2 As MainClass _
) As MainClass
```

C#

```
public static MainClass operator +(
   MainClass c1,
   MainClass c2
)
```

```
 [F#]
 static member (+)(
   c1:MainClass *
   c2:MainClass
)
```

Parameters

c1

c2

Remarks

This is an example of operator overloading.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

MainClass.GetEnumerator Method

Returns an enumerator that iterates through the collection.

VB

```
Public Function GetEnumerator() As IEnumerator(Of String) Implements  _
   String).GetEnumerator
```

C#

```
public IEnumerator<string> GetEnumerator()
```

```
[F#]
member GetEnumerator : unit -> IEnumerator<string>
```

Returns

A IEnumerator<T> that can be used to iterate through the collection.

Implements

String>.GetEnumerator

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

MainClass.GetEnumerator1 Method

Returns an enumerator that iterates through a collection.

VB

```
Private Function GetEnumerator1() As IEnumerator
```

C#

```
private IEnumerator GetEnumerator1()
```

```
[F#]
member private GetEnumerator1 : unit -> IEnumerator
```

Returns

An IEnumerator object that can be used to iterate through the collection.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

MainClass.ISampleInterface.TestMethod Method

A method with an optional parameter.

VB

```
Private Sub ISampleInterface.TestMethod( _
   Optional ByVal x As Integer = 3 _
) Implements  _
   ISampleInterface.TestMethod
```

C#

```
private void ISampleInterface.TestMethod(
   int x = 3
)
```

```
[F#]
member private ISampleInterface.TestMethod :
  ?x:int
-> unit
```

Parameters

x

Implements

ISampleInterface.TestMethod[42]

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

MainClass.Method1 (Int32) Method

Our sample method.

VB

```
Public Function Method1( _
   ByVal x As Integer _
) As Boolean
```

C#

```
public bool Method1(
   int x
)
```

```
[F#]
member Method1 :
  x:int
-> bool
```

Parameters

x

The first parameter.

Returns

true if no error occurs; **false** otherwise.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

MainClass.Method1 (String) Method

Sample method with one String argument.

VB

```
Public Function Method1( _
   ByVal x As String _
) As Boolean
```

C#

```
public bool Method1(
   string x
)
```

```
[F#]
member Method1 :
  x:string
-> bool
```

Parameters

x

The first parameter.

Returns

true if no error occurs; **false** otherwise.

Remarks

This remarks section was included from external XML file using <include> comment tag.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

MainClass.Method1<T> Method

Sample generic method with two arguments.

VB

```
Public Function Method1(Of  _
   T)( _
```

```
   ByVal x As Integer(), _
   ByVal y As String _
) As Boolean
```

C#

```
public bool Method1<T>(
   int[] x,
   string y
)
```

```
[F#]
member Method1<'T> :
  x:int[] *
  y:string
-> bool
```

Type Parameters

T

The type used in this generic method.

Parameters

x

The first parameter.

y

The second parameter.

Returns

true if no error occurs; **false** otherwise.

Exceptions

Exception type	Condition
NestedException[34]	If something horrible happens.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

MainClass.System.Collections.IEnumerable.GetEnumerator Method

Returns an enumerator that iterates through a collection.

VB

```
Private Function System.Collections.IEnumerable.GetEnumerator() As IEnumerator
Implements _
   IEnumerable.GetEnumerator
```

C#

```
private IEnumerator System.Collections.IEnumerable.GetEnumerator()
```

```
 [F#]
 member private System.Collections.IEnumerable.GetEnumerator : unit ->
IEnumerator
```

Returns

An IEnumerator object that can be used to iterate through the collection.

Implements

IEnumerable.GetEnumerator

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

TextChanged Event

Occurs when the Text property value has changed.

VB

```
Public Event TextChanged As EventHandler(Of TextChangedEventArgs)
```

C#

```
public event EventHandler<TextChangedEventArgs> TextChanged
```

```
 [F#]
 member TextChanged : IEvent<EventHandler<TextChangedEventArgs>, _>
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

NestedException Class

NOTE: This member is now obsolete.

Use other exception type instead.

This is our custom exception.

System.Object
  System.Exception
    SampleClassLibrary.MainClass.NestedException

VB

```
<Obsolete("Use other exception type instead.")> _
Public Class NestedException
    Inherits Exception
```

C#

```
[Obsolete("Use other exception type instead.")]
public class NestedException : Exception
```

```
 [F#]
 [<Obsolete("Use other exception type instead.")>]
type NestedException = class
   inherit Exception
end
```

Remarks

This is also an example of a nested class and ObsoleteAttribute.

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Properties

Data (inherited from Exception), HResult (inherited from Exception), HelpLink (inherited from Exception), InnerException (inherited from Exception), Message (inherited from Exception), Source (inherited from Exception), StackTrace (inherited from Exception), TargetSite (inherited from Exception)

Methods

Equals (inherited from Object), Finalize (inherited from Object), GetBaseException (inherited from Exception), GetHashCode (inherited from Object), GetObjectData (inherited from Exception), GetType (inherited from Exception), MemberwiseClone (inherited from Object), ToString (inherited from Exception)

Events

SerializeObjectState (inherited from Exception)

See Also

Applies to: MainClass[23]

EmptyInstance Field

An instance with uninitialized property values.

VB

```
Public Shared Shadows ReadOnly EmptyInstance As MainClass
```

C#

```
new public static readonly MainClass EmptyInstance
```

```
[F#]
static val  EmptyInstance : MainClass
```

Remarks

This is an example of a field.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

InternalField Field

An internal field with logical value.

VB

```
Protected InternalField As Boolean
```

C#

```
protected bool InternalField
```

```
[F#]
val protected  InternalField : bool
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

_SelectedValue Field

VB

```
Private _SelectedValue As SampleEnum
```

C#

```
private SampleEnum _SelectedValue
```

```
[F#]
val private  _SelectedValue : SampleEnum
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

_Text Field

VB

```
Private _Text As String
```

C#

```
private string _Text
```

```
[F#]
val private  _Text : string
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

CONSTANT_1 Field

Sample constant #1.

VB

```
Public Const CONSTANT_1 As Integer = 1
```

C#

```
public const int CONSTANT_1 = 1
```

```
[F#]
static val mutable  CONSTANT_1 : int
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

CONSTANT_2 Field

Sample constant #2.

```
[F#]
static val mutable  CONSTANT_2 : int
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: MainClass[23]

TextChangedEventArgs Class

Provides data for the MainClass.TextChanged event.

System.Object
  System.EventArgs
    SampleClassLibrary.TextChangedEventArgs

VB

```
Public Class TextChangedEventArgs
    Inherits EventArgs
```

C#

```
public class TextChangedEventArgs : EventArgs
```

```
 [F#]
 type TextChangedEventArgs = class
   inherit EventArgs
end
```

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Constructors

TextChangedEventArgs[39]

Properties

NewValue[39], OldValue[40]

Methods

Equals (inherited from Object), Finalize (inherited from Object), GetHashCode (inherited from Object), GetType (inherited from Object), MemberwiseClone (inherited from Object), ToString (inherited from Object)

Fields

_NewValue[40], _OldValue[41]

TextChangedEventArgs Constructor

Initializes a new instance of the TextChangedEventArgs class.

VB

```
Friend Sub New( _
   ByVal oldVal As String, _
   ByVal newVal As String _
)
```

C#

```
internal TextChangedEventArgs(
   string oldVal,
   string newVal
)
```

```
[F#]
internal new :
  oldVal:string *
  newVal:string
-> unit
```

Parameters

oldVal

The old value of MainClass.Text property.

newVal

The new value of MainClass.Text property.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: TextChangedEventArgs[38]

TextChangedEventArgs.NewValue Property

Gets the new value of MainClass.Text property.

VB

```
Public ReadOnly Property NewValue() As String
```

C#

```
public string NewValue {get;}
```

```
[F#]
member  NewValue : string with get
```

Property Value

New string value.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: TextChangedEventArgs[38]

TextChangedEventArgs.OldValue Property

Gets the old value of MainClass.Text property.

VB

```
Public ReadOnly Property OldValue() As String
```

C#

```
public string OldValue {get;}
```

```
[F#]
member  OldValue : string with get
```

Property Value

Old string value.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: TextChangedEventArgs[38]

_NewValue Field

VB

```
Private _NewValue As String
```

C#

```
private string _NewValue
```

```
[F#]
val private  _NewValue : string
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: TextChangedEventArgs[38]

_OldValue Field

VB

```
Private _OldValue As String
```

C#

```
private string _OldValue
```

```
[F#]
val private  _OldValue : string
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: TextChangedEventArgs[38]

ISampleInterface Interface

Example of an interface.

VB

```
Public Interface ISampleInterface
```

C#

```
public interface ISampleInterface
```

```
[F#]
type ISampleInterface = interface end
```

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Methods

TestMethod[42]

ISampleInterface.TestMethod Method

A method with an optional parameter.

VB

```
Sub TestMethod( _
   Optional ByVal x As Integer = 3 _
)
```

C#

```
void TestMethod(
   int x = 3
)
```

```
[F#]
member TestMethod :
  ?x:int
-> unit
```

Parameters

x

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: ISampleInterface[41]

GeneralInfo Structure

Represents information about any object.

VB

```
Public Structure GeneralInfo
```

C#

```
public struct GeneralInfo
```

```
[F#]
type GeneralInfo = struct end
```

Remarks

This is en example of a structure.

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Fields

Description[43], Name[44], Tag[44]

Description Field

A description of the object.

VB

```
Public Description As String
```

C#

```
public string Description
```

```
[F#]
```

```
val  Description : string
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: GeneralInfo[43]

Name Field

A name of the object.

VB

```
Public Name As String
```

C#

```
public string Name
```

```
[F#]
val  Name : string
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: GeneralInfo[43]

Tag Field

Additional info related to the object.

VB

```
Public Tag As Object
```

C#

```
public object Tag
```

```
[F#]
val  Tag : Object
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: GeneralInfo[43]

SampleEnum Enumeration

Sample enumeration.

This enumeration has a FlagsAttribute attribute that allows a bitwise combination of its member values.

Constant	Value	Description
value1	1	First value.
value2	2	Second value
value3	4	Third value.

Requirements

Namespace: SampleClassLibrary[7]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

SampleClassLibrary.Utils Namespace

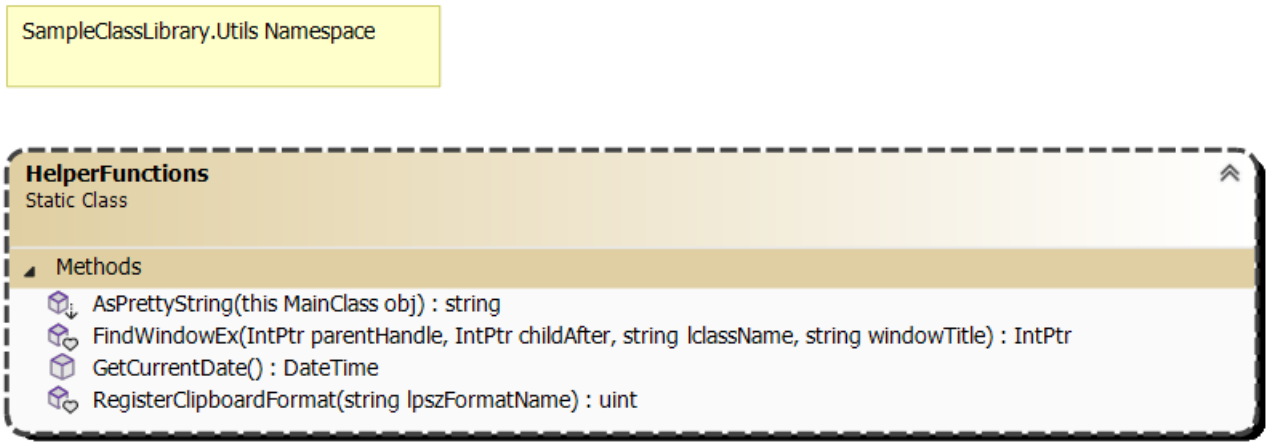
Contains general-purpose helper classes.

Remarks

To document a namespace, you do this directly in the source code. Just select any file that contains the namespace and add an XML comment in front of the **namespace** keyword. You can use VSdocman's "Add XML comment" from context menu or the comment editor.

You can ignore the VS warning that XML comment will be ignored for a namespace. If you add an XML comment for the same namespace on multiple places, VSdocman will show a warning during the compilation and a random comment will be used.

This is the class diagram of this namespace:



Classes

HelperFunctions[46]

HelperFunctions Class

This is an example of a module in VB which is represented as a sealed class with static members in C#.

System.Object
  SampleClassLibrary.Utils.HelperFunctions

VB

```
Friend Class HelperFunctions
```

C#

```
internal static class HelperFunctions
```

```
[F#]
type internal HelperFunctions = class end
```

Remarks

In VB, you can use included methods directly without specifying HelperFunctions name.

Requirements

Namespace: SampleClassLibrary.Utils[46]

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

Assembly: SampleClassLibrary (in SampleClassLibrary.dll)

Methods

AsPrettyString[47], Equals (inherited from Object), Finalize (inherited from Object), FindWindowEx[48], GetCurrentDate[49], GetHashCode (inherited from Object), GetType (inherited from Object), MemberwiseClone (inherited from Object), RegisterClipboardFormat[49], ToString (inherited from Object)

HelperFunctions.AsPrettyString Method

An example of an **extension method**. Returns a human readable string representing an instance of MainClass.

VB

```
<Extension()> _
Public Shared Function AsPrettyString( _
   ByVal obj As MainClass _
) As String
```

C#

```
public static string AsPrettyString(
   this MainClass obj
)
```

```
[F#]
static member AsPrettyString :
  obj:MainClass
-> string
```

Parameters

obj

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: HelperFunctions[46]

HelperFunctions.FindWindowEx Method

An example of an **external** function. Retrieves a handle to a window whose class name and window name match the specified strings.

VB

```
<DllImport("user32.dll", SetLastError = true, CharSet = CharSet.Auto)> _
Friend Shared Function FindWindowEx( _
   ByVal parentHandle As IntPtr, _
   ByVal childAfter As IntPtr, _
   ByVal lclassName As String, _
   ByVal windowTitle As String _
) As IntPtr
```

C#

```
[DllImport("user32.dll", SetLastError = true, CharSet = CharSet.Auto)]
internal static extern IntPtr FindWindowEx(
   IntPtr parentHandle,
   IntPtr childAfter,
   string lclassName,
   string windowTitle
)
```

```
 [F#]
 [<DllImport("user32.dll", SetLastError = true, CharSet = CharSet.Auto)>]
extern IntPtr FindWindowEx(
   parentHandle:IntPtr *
   childAfter:IntPtr *
   lclassName:string *
   windowTitle:string
)
```

Parameters

parentHandle

A handle to the parent window whose child windows are to be searched.

childAfter

A handle to a child window. The search begins with the next child window in the Z order. The child window must be a direct child window of hwndParent, not just a descendant window.

lclassName

The class name or a class atom created by a previous call to the **RegisterClass** or **RegisterClassEx** function.

windowTitle

The window name (the window's title). If this parameter is null reference (**Nothing** in Visual Basic), all window names match.

Returns

If the function succeeds, the return value is a handle to the window that has the specified class and window names.

If the function fails, the return value is null reference (**Nothing** in Visual Basic).

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: HelperFunctions[46]

HelperFunctions.GetCurrentDate Method

Gets the current date and time.

VB

```
Public Shared Function GetCurrentDate() As Date
```

C#

```
public static DateTime GetCurrentDate()
```

```
[F#]
static member GetCurrentDate : unit -> DateTime
```

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: HelperFunctions[46]

HelperFunctions.RegisterClipboardFormat Method

An example of an **external** function. Registers a new clipboard format. This format can then be used as a valid clipboard format.

VB

```
<DllImport("user32.dll")> _
Friend Shared Function RegisterClipboardFormat( _
   ByVal lpszFormatName As String _
) As UInt32
```

C#

```
[DllImport("user32.dll")]
internal static extern UInt32 RegisterClipboardFormat(
   string lpszFormatName
)
```

```
 [F#]
 [<DllImport("user32.dll")>]
extern UInt32 RegisterClipboardFormat(
   lpszFormatName:string
)
```

Parameters

lpszFormatName

The name of the new format.

Returns

If the function succeeds, the return value identifies the registered clipboard format.

If the function fails, the return value is zero.

Requirements

Platforms: Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Vista SP2

See Also

Applies to: HelperFunctions[46]

License Agreement

The license agreement.

About

SampleClassLibrary version 1.0

Home page and registration info: http://www.helixoft.com/

Index