

library ahbsd.lib

1.2.2

Generated by Doxygen 1.9.1



<b>1 LICENSE</b>	<b>1</b>
<b>2 ahbsd.lib</b>	<b>5</b>
2.1 About generic EventArgs	5
2.2 About generic ChangeEventArgs	5
<b>3 Namespace Index</b>	<b>7</b>
3.1 Packages	7
<b>4 Hierarchical Index</b>	<b>9</b>
4.1 Class Hierarchy	9
<b>5 Class Index</b>	<b>11</b>
5.1 Class List	11
<b>6 Namespace Documentation</b>	<b>13</b>
6.1 ahbsd Namespace Reference	13
6.2 ahbsd.lib Namespace Reference	13
6.2.1 Function Documentation	13
6.2.1.1 ChangeEventHandler< T >()	13
6.3 ahbsd.lib.ApiKey Namespace Reference	14
6.3.1 Function Documentation	14
6.3.1.1 ApiKeyEventHandler< T >()	14
6.4 ahbsd.lib.Exceptions Namespace Reference	15
6.5 ahbsd.lib.Tools Namespace Reference	15
6.6 Test Namespace Reference	15
<b>7 Class Documentation</b>	<b>17</b>
7.1 ahbsd.lib.ApiKey.ApiKeyEventArgs< T > Class Template Reference	17
7.1.1 Detailed Description	17
7.1.2 Constructor & Destructor Documentation	18
7.1.2.1 ApiKeyEventArgs() [1/2]	18
7.1.2.2 ApiKeyEventArgs() [2/2]	18
7.1.3 Property Documentation	18
7.1.3.1 Index	18
7.2 ahbsd.lib.ApiKey.ApiKeyHolder< T > Class Template Reference	19
7.2.1 Detailed Description	20
7.2.2 Constructor & Destructor Documentation	21
7.2.2.1 ApiKeyHolder() [1/2]	21
7.2.2.2 ApiKeyHolder() [2/2]	21
7.2.3 Member Function Documentation	21
7.2.3.1 Equals() [1/2]	21
7.2.3.2 Equals() [2/2]	22
7.2.3.3 FindApiKey()	22

7.2.3.4 GetApiKey()	23
7.2.3.5 GetHashCode()	23
7.2.3.6 operator!=()	23
7.2.3.7 operator==()	24
7.2.4 Member Data Documentation	24
7.2.4.1 KnownApiKeys	24
7.2.5 Property Documentation	24
7.2.5.1 ApiKey	24
7.2.6 Event Documentation	25
7.2.6.1 OnApiKeyAdded	25
7.3 ahbsd.lib.ChangeEventArgs< T > Class Template Reference	25
7.3.1 Detailed Description	26
7.3.2 Constructor & Destructor Documentation	26
7.3.2.1 ChangeEventArgs() [1/3]	26
7.3.2.2 ChangeEventArgs() [2/3]	26
7.3.2.3 ChangeEventArgs() [3/3]	27
7.3.3 Member Function Documentation	27
7.3.3.1 Equals() [1/2]	27
7.3.3.2 Equals() [2/2]	28
7.3.3.3 GetHashCode()	28
7.3.3.4 operator!=()	28
7.3.3.5 operator==()	29
7.3.3.6 SetNewValue()	29
7.3.3.7 ToString()	29
7.3.4 Property Documentation	30
7.3.4.1 NewValue	30
7.3.4.2 OldValue	30
7.4 ahbsd.lib.Tools.ConsolePrintTable Class Reference	30
7.4.1 Detailed Description	31
7.4.2 Member Function Documentation	31
7.4.2.1 Print()	31
7.5 ahbsd.lib.EventArgs< T > Class Template Reference	31
7.5.1 Detailed Description	32
7.5.2 Constructor & Destructor Documentation	32
7.5.2.1 EventArgs() [1/2]	32
7.5.2.2 EventArgs() [2/2]	32
7.5.3 Property Documentation	32
7.5.3.1 Value	33
7.6 ahbsd.lib.Exceptions.Exception< T > Class Template Reference	33
7.6.1 Detailed Description	33
7.6.2 Constructor & Destructor Documentation	34
7.6.2.1 Exception()	34

7.6.3 Property Documentation	35
7.6.3.1 Value	35
7.7 ahbsd.lib.ApiKey.IApiKeyEventArgs Interface Reference	35
7.7.1 Detailed Description	35
7.7.2 Property Documentation	36
7.7.2.1 Index	36
7.8 ahbsd.lib.IChangeEventArgs< T > Interface Template Reference	36
7.8.1 Detailed Description	36
7.8.2 Member Function Documentation	37
7.8.2.1 SetNewValue()	37
7.8.2.2 ToString()	37
7.8.3 Property Documentation	37
7.8.3.1 NewValue	38
7.8.3.2 OldValue	38
7.9 ahbsd.lib.IEventArgs< T > Interface Template Reference	38
7.9.1 Detailed Description	38
7.9.2 Property Documentation	39
7.9.2.1 Value	39
7.10 ahbsd.lib.Exceptions.IGenericException< T > Interface Template Reference	39
7.10.1 Detailed Description	39
7.10.2 Property Documentation	40
7.10.2.1 Value	40
7.11 Test.ITestClass< T, A > Interface Template Reference	40
7.11.1 Detailed Description	40
7.11.2 Property Documentation	41
7.11.2.1 Variable	41
7.11.3 Event Documentation	41
7.11.3.1 OnChange	41
7.12 Test.Program Class Reference	41
7.12.1 Detailed Description	41
7.13 Test.TestClass< T, A > Class Template Reference	42
7.13.1 Detailed Description	42
7.13.2 Constructor & Destructor Documentation	43
7.13.2.1 TestClass() [1/4]	43
7.13.2.2 TestClass() [2/4]	43
7.13.2.3 TestClass() [3/4]	43
7.13.2.4 TestClass() [4/4]	43
7.13.3 Property Documentation	44
7.13.3.1 Variable	44
7.13.4 Event Documentation	44
7.13.4.1 OnChange	44



# Chapter 1

## LICENSE

Apache License Version 2.0, January 2004 <http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.



9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.



## Chapter 2

# ahbsd.lib

Classes with functionality I miss

- For example generic EventArgs or EventArgs for changing values. These two are in version 1.0.
- In Version 1.0.1 some classes for generic API-Keys were added.
- In Version 1.2.0 some code beautification was done, the license changes to Apache 2.0 and a generic Exception was added.

### 2.1 About generic EventArgs

The generic EventArgs have a generic Value that could be set with the constructor.

### 2.2 About generic ChangeEventArgs

The generic ChangeEventArgs have two generic values. One for the old value and one for the new value.



## Chapter 3

# Namespace Index

### 3.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">ahbsd</a>	13
<a href="#">ahbsd.lib</a>	13
<a href="#">ahbsd.lib.ApiKey</a>	14
<a href="#">ahbsd.lib.Exceptions</a>	15
<a href="#">ahbsd.lib.Tools</a>	15
<a href="#">Test</a>	15



## Chapter 4

# Hierarchical Index

### 4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ahbsd.lib.ApiKey.ApiKeyHolder< T > . . . . .	19
ahbsd.lib.ApiKey.ApiKeyHolder< A > . . . . .	19
Test.TestClass< T, A > . . . . .	42
ahbsd.lib.Tools.ConsolePrintTable . . . . .	30
ahbsd.lib.ApiKey.IApiKeyEventArgs . . . . .	35
ahbsd.lib.ApiKey.ApiKeyEventArgs< T > . . . . .	17
ahbsd.lib.IChangeEventArgs< T > . . . . .	36
ahbsd.lib.ChangeEventArgs< T > . . . . .	25
ahbsd.lib.IEventArgs< T > . . . . .	38
ahbsd.lib.EventArgs< T > . . . . .	31
ahbsd.lib.ApiKey.ApiKeyEventArgs< T > . . . . .	17
ahbsd.lib.ChangeEventArgs< T > . . . . .	25
ahbsd.lib.Exceptions.IGenericException< T > . . . . .	39
ahbsd.lib.Exceptions.Exception< T > . . . . .	33
Test.ITestClass< T, A > . . . . .	40
Test.TestClass< T, A > . . . . .	42
Test.Program . . . . .	41





## Chapter 5

# Class Index

### 5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">ahbsd.lib.ApiKey.ApiKeyEventArgs&lt; T &gt;</a>	
Specialized generic <a href="#">EventArgs</a> for API-Keys	17
<a href="#">ahbsd.lib.ApiKey.ApiKeyHolder&lt; T &gt;</a>	
Class for generic API-Keys	19
<a href="#">ahbsd.lib.ChangeEventArgs&lt; T &gt;</a>	
Generic <a href="#">EventArgs</a> for changing values	25
<a href="#">ahbsd.lib.Tools.ConsolePrintTable</a>	
Static class to print a DataTable to console	30
<a href="#">ahbsd.lib.EventArgs&lt; T &gt;</a>	
Generic <a href="#">EventArgs</a>	31
<a href="#">ahbsd.lib.Exceptions.Exception&lt; T &gt;</a>	
Class for a generic <a href="#">Exception</a> , which additionally holds a value of T	33
<a href="#">ahbsd.lib.ApiKey.IApiKeyEventArgs</a>	
Interface for ApiKeyEventArgs<T>	35
<a href="#">ahbsd.lib.IChangeEventArgs&lt; T &gt;</a>	
Interface for generic <a href="#">EventArgs</a> for changing values	36
<a href="#">ahbsd.lib.IEventArgs&lt; T &gt;</a>	
Interface for generic <a href="#">EventArgs</a>	38
<a href="#">ahbsd.lib.Exceptions.IGenericException&lt; T &gt;</a>	
Interface for generic exceptions	39
<a href="#">Test.ITestClass&lt; T, A &gt;</a>	
An interface for a class to demonstrate ChangeEventArgs<T> and ChangeEventHandler<T>	40
<a href="#">Test.Program</a>	
The test program	41
<a href="#">Test.TestClass&lt; T, A &gt;</a>	
A class to demonstrate ChangeEventArgs<T> and ChangeEventHandler<T>	42



## Chapter 6

# Namespace Documentation

### 6.1 ahbsd Namespace Reference

### 6.2 ahbsd.lib Namespace Reference

#### Classes

- class [ChangeEventArgs](#)  
*Generic [EventArgs](#) for changing values.*
- class [EventArgs](#)  
*Generic [EventArgs](#).*
- interface [IChangeEventArgs](#)  
*Interface for generic [EventArgs](#) for changing values.*
- interface [IEventArgs](#)  
*Interface for generic [EventArgs](#).*

#### Functions

- delegate void [ChangeEventHandler< T >](#) (object sender, [ChangeEventArgs< T >](#) e)  
*A delegate for change events.*

#### 6.2.1 Function Documentation

##### 6.2.1.1 [ChangeEventHandler< T >\(\)](#)

```
delegate void ahbsd.lib.ChangeEventHandler< T > (
    object sender,
    ChangeEventArgs< T > e )
```

A delegate for change events.

## Template Parameters

<i>T</i>	The type of changing values.
----------	------------------------------

## Parameters

<i>sender</i>	Sending object.
<i>e</i>	The changing <a href="#">EventArgs</a> .

## 6.3 ahbsd.lib.ApiKey Namespace Reference

### Classes

- class [ApiKeyEventArgs](#)  
*Specialized generic [EventArgs](#) for API-Keys.*
- class [ApiKeyHolder](#)  
*Class for generic API-Keys.*
- interface [IApiKeyEventArgs](#)  
*Interface for [ApiKeyEventArgs](#)< T >.*

### Functions

- delegate void [ApiKeyEventHandler](#)< T > (object sender, [ApiKeyEventArgs](#)< T > e)  
*Delegate for events with generic API-Keys.*

### 6.3.1 Function Documentation

#### 6.3.1.1 ApiKeyEventHandler< T >()

```
delegate void ahbsd.lib.ApiKey.ApiKeyEventHandler< T > (
    object sender,
    ApiKeyEventArgs< T > e )
```

Delegate for events with generic API-Keys.

## Template Parameters

<i>T</i>	The Type of the API-Key.
----------	--------------------------

## Parameters

<i>sender</i>	The sending object.
---------------	---------------------

## Parameters

<i>e</i>	The event arguments with a generic API-Key.
----------	---

## 6.4 ahbsd.lib.Exceptions Namespace Reference

### Classes

- class [Exception](#)  
*Class for a generic [Exception](#), which additionally holds a value of *T**
- interface [IGenericException](#)  
*Interface for generic exceptions.*

## 6.5 ahbsd.lib.Tools Namespace Reference

### Classes

- class [ConsolePrintTable](#)  
*Static class to print a *DataTable* to console.*

## 6.6 Test Namespace Reference

### Classes

- interface [ITestClass](#)  
*An interface for a class to demonstrate *ChangeEventArgs<T>* and *ChangeEventHandler<T>*.*
- class [Program](#)  
*The test program.*
- class [TestClass](#)  
*A class to demonstrate *ChangeEventArgs<T>* and *ChangeEventHandler<T>*.*



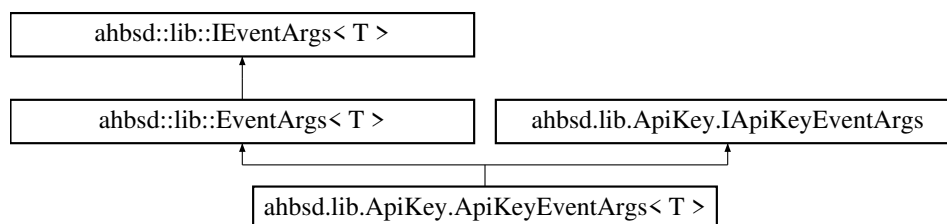
## Chapter 7

# Class Documentation

### 7.1 ahbsd.lib.ApiKey.ApiKeyEventArgs< T > Class Template Reference

Specialized generic [EventArgs](#) for API-Keys.

Inheritance diagram for ahbsd.lib.ApiKey.ApiKeyEventArgs< T >:



#### Public Member Functions

- [ApiKeyEventArgs](#) (T apiKey, int idx)  
*Constructor with the API-Key and the index of the API-Key.*
- [ApiKeyEventArgs](#) (T apiKey)  
*Constructor with the API-Key.*

#### Properties

- int? [Index](#) [get]  
*Gets the index of the API-Key from the ApiKeyHolder< T >.*

#### 7.1.1 Detailed Description

Specialized generic [EventArgs](#) for API-Keys.

### Template Parameters

<i>T</i>	Type of API-Key.
----------	------------------

### See also

`ApiKeyHolder<T>`

## 7.1.2 Constructor & Destructor Documentation

### 7.1.2.1 `ApiKeyEventArgs()` [1/2]

```
ahbsd.lib.ApiKey.ApiKeyEventArgs< T >.ApiKeyEventArgs (
    T apiKey,
    int idx )
```

Constructor with the API-Key and the index of the API-Key.

#### Parameters

<i>apiKey</i>	The API-Key.
<i>idx</i>	The index of the API-Key.

### 7.1.2.2 `ApiKeyEventArgs()` [2/2]

```
ahbsd.lib.ApiKey.ApiKeyEventArgs< T >.ApiKeyEventArgs (
    T apiKey )
```

Constructor with the API-Key.

#### Parameters

<i>apiKey</i>	The API-Key.
---------------	--------------

## 7.1.3 Property Documentation

### 7.1.3.1 Index

```
int? ahbsd.lib.ApiKey.ApiKeyEventArgs< T >.Index [get]
```



Gets the index of the API-Key from the ApiKeyHolder<T>.

The index.

The documentation for this class was generated from the following file:

- ahbsd.lib/ApiKey/ApiKeyEventArgs.cs

## 7.2 ahbsd.lib.ApiKey.ApiKeyHolder< T > Class Template Reference

Class for generic API-Keys.

Inherits IEquatable< ApiKeyHolder< T >>.

### Public Member Functions

- [ApiKeyHolder](#) (T apiKey)  
*Constructor with a given API-Key.*
- [ApiKeyHolder](#) ()  
*Constructor without parameters.*
- override bool [Equals](#) (object obj)  
*Find out, if this object equals another given object.*
- override int [GetHashCode](#) ()  
*Gets the HashCode.*
- bool [Equals](#) (ApiKeyHolder< T > other)  
*Find out, if this object equals another given object.*

### Static Public Member Functions

- static ? int [FindApiKey](#) (T apiKey)  
*Looks for a given API-Key.*
- static T [GetApiKey](#) (int idx)  
*Returns an API-Key from a defined index number.*
- static bool [operator==](#) (ApiKeyHolder< T > left, ApiKeyHolder< T > right)  
*Find out if two objects equals.*
- static bool [operator!=](#) (ApiKeyHolder< T > left, ApiKeyHolder< T > right)  
*Find out if two objects do not equals.*

### Static Package Attributes

- static List< T > [KnownApiKeys](#)  
*A list of all known API-Keys.*

### Properties

- T [ApiKey](#) [get]  
*Gets the API-Key.*

## Events

- static ApiKeyEventHandler< T > [OnApiKeyAdded](#)  
*Happens if a new API-Key was added to the static list [KnownApiKeys](#).*

### 7.2.1 Detailed Description

Class for generic API-Keys.

## Template Parameters

<i>T</i>	Type of API-Key
----------	-----------------

## 7.2.2 Constructor &amp; Destructor Documentation

## 7.2.2.1 ApiKeyHolder() [1/2]

```
ahbsd.lib.ApiKey.ApiKeyHolder< T >.ApiKeyHolder (
    T apiKey )
```

Constructor with a given API-Key.

## Parameters

<i>apiKey</i>	The API-Key.
---------------	--------------

## 7.2.2.2 ApiKeyHolder() [2/2]

```
ahbsd.lib.ApiKey.ApiKeyHolder< T >.ApiKeyHolder ( )
```

Constructor without parameters.

If before an object was created, the last API-Key will be used. Otherwise the default of *T* will be used.

## Exceptions

<i>ArgumentNullException</i>	If <a href="#">KnownApiKeys</a> is <code>null</code> or something similar.
<i>InvalidOperationException</i>	If anything regarding <a href="#">KnownApiKeys</a> is an invalid operation.

References [ahbsd.lib.ApiKey.ApiKeyHolder< T >.ApiKey](#), and [ahbsd.lib.ApiKey.ApiKeyHolder< T >.KnownApiKeys](#).

## 7.2.3 Member Function Documentation

## 7.2.3.1 Equals() [1/2]

```
bool ahbsd.lib.ApiKey.ApiKeyHolder< T >.Equals (
    ApiKeyHolder< T > other )
```

Find out, if this object equals another given object.

**Parameters**

<i>other</i>	The other object.
--------------	-------------------

**Returns**

If both objects equals `TRUE`, otherwise `FALSE`.

References [ahbsd.lib.ApiKey.ApiKeyHolder< T >.ApiKey](#), and [ahbsd.lib.ApiKey.ApiKeyHolder< T >.Equals\(\)](#).

**7.2.3.2 Equals()** [2/2]

```
override bool ahbsd.lib.ApiKey.ApiKeyHolder< T >.Equals (
    object obj )
```

Find out, if this object equals another given object.

**Parameters**

<i>obj</i>	The other object.
------------	-------------------

**Returns**

If both objects equals `TRUE`, otherwise `FALSE`.

Referenced by [ahbsd.lib.ApiKey.ApiKeyHolder< T >.Equals\(\)](#).

**7.2.3.3 FindApiKey()**

```
static ? int ahbsd.lib.ApiKey.ApiKeyHolder< T >.FindApiKey (
    T apiKey ) [static]
```

Looks for a given API-Key.

**Parameters**

<i>apiKey</i>	The given API-Key.
---------------	--------------------

**Returns**

If found it returns the index, if not `null` is returned.

References [ahbsd.lib.ApiKey.ApiKeyHolder< T >.KnownApiKeys](#).

#### 7.2.3.4 GetApiKey()

```
static T ahbsd.lib.ApiKey.ApiKeyHolder< T >.GetApiKey (
    int idx ) [static]
```

Returns an API-Key from a defined index number.

##### Parameters

<i>idx</i>	The defined index number.
------------	---------------------------

##### Returns

An API-Key.

References [ahbsd.lib.ApiKey.ApiKeyHolder< T >.KnownApiKeys](#).

#### 7.2.3.5 GetHashCode()

```
override int ahbsd.lib.ApiKey.ApiKeyHolder< T >.GetHashCode ( )
```

Gets the HashCode.

##### Returns

The HashCode.

References [ahbsd.lib.ApiKey.ApiKeyHolder< T >.ApiKey](#).

#### 7.2.3.6 operator"!="()

```
static bool ahbsd.lib.ApiKey.ApiKeyHolder< T >.operator!= (
    ApiKeyHolder< T > left,
    ApiKeyHolder< T > right ) [static]
```

Find out if two objects do not equals.

##### Parameters

<i>left</i>	The object on the left side.
<i>right</i>	The object on the right side.

##### Returns

If both objects do not equals TRUE, otherwise FALSE.

### 7.2.3.7 operator==()

```
static bool ahbsd.lib.ApiKey.ApiKeyHolder< T >.operator== (
    ApiKeyHolder< T > left,
    ApiKeyHolder< T > right ) [static]
```

Find out if two objects equals.

#### Parameters

<i>left</i>	The object on the left side.
<i>right</i>	The object on the right side.

#### Returns

If both objects equals `TRUE`, otherwise `FALSE`.

## 7.2.4 Member Data Documentation

### 7.2.4.1 KnownApiKeys

```
List<T> ahbsd.lib.ApiKey.ApiKeyHolder< T >.KnownApiKeys [static], [package]
```

A list of all known API-Keys.

Of current instances. Is eg needed for construction without api-key etc.

Referenced by [ahbsd.lib.ApiKey.ApiKeyHolder< T >.ApiKeyHolder\(\)](#), [ahbsd.lib.ApiKey.ApiKeyHolder< T >.FindApiKey\(\)](#), and [ahbsd.lib.ApiKey.ApiKeyHolder< T >.GetApiKey\(\)](#).

## 7.2.5 Property Documentation

### 7.2.5.1 ApiKey

```
T ahbsd.lib.ApiKey.ApiKeyHolder< T >.ApiKey [get], [protected]
```

Gets the API-Key.

The API-Key.

Referenced by [ahbsd.lib.ApiKey.ApiKeyHolder< T >.ApiKeyHolder\(\)](#), [ahbsd.lib.ApiKey.ApiKeyHolder< T >.Equals\(\)](#), and [ahbsd.lib.ApiKey.ApiKeyHolder< T >.GetHashCode\(\)](#).

## 7.2.6 Event Documentation

### 7.2.6.1 OnApiKeyAdded

ApiKeyEventHandler<T> [ahbsd.lib.ApiKey.ApiKeyHolder](#)< T >.OnApiKeyAdded [static], [package]

Happens if a new API-Key was added to the static list [KnownApiKeys](#).

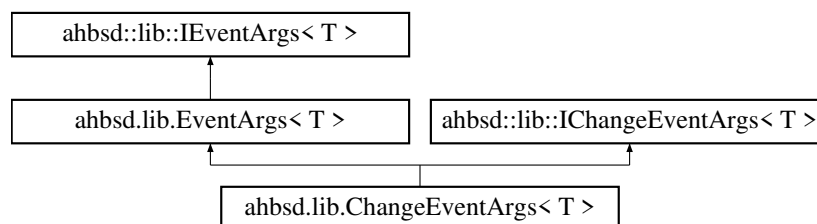
The documentation for this class was generated from the following file:

- ahbsd.lib/ApiKey/ApiKeyHolder.cs

## 7.3 ahbsd.lib.ChangeEventArgs< T > Class Template Reference

Generic [EventArgs](#) for changing values.

Inheritance diagram for ahbsd.lib.ChangeEventArgs< T >:



### Public Member Functions

- [ChangeEventArgs](#) ()  
*Constructor without any parameters.*
- [ChangeEventArgs](#) (T oldValue, T newValue)  
*Constructor with the old and the new value.*
- [ChangeEventArgs](#) (T oldValue)  
*Constructor with the old value.*
- void [SetNewValue](#) (T newValue)  
*Sets the new value.*
- override string [ToString](#) ()  
*Gets a string representation of the changed value.*
- bool [Equals](#) (IChangeEventArgs< T > other)  
*Finds out wheather an other object of type IChangeEventArgs<T> equals this object.*
- override bool [Equals](#) (object obj)  
*Finds out wheather an other object equals this object.*
- override int [GetHashCode](#) ()  
*Gets the HashCode of this object.*

## Static Public Member Functions

- static bool `operator==` (`ChangeEventArgs< T >` left, `ChangeEventArgs< T >` right)  
*Finds out, if two objects of type `ChangeEventArgs< T >` equals each other.*
- static bool `operator!=` (`ChangeEventArgs< T >` left, `ChangeEventArgs< T >` right)  
*Finds out, if two objects of type `ChangeEventArgs< T >` do not equals each other.*

## Properties

- T `OldValue` [get]  
*Gets the old value.*
- T `NewValue` [get]  
*Gets the new value.*

### 7.3.1 Detailed Description

Generic `EventArgs` for changing values.

Template Parameters

<code>T</code>	The type of the changing Values.
----------------	----------------------------------

### 7.3.2 Constructor & Destructor Documentation

#### 7.3.2.1 `ChangeEventArgs()` [1/3]

```
ahbsd.lib.ChangeEventArgs< T >.ChangeEventArgs ( )
```

Constructor without any parameters.

References `ahbsd.lib.ChangeEventArgs< T >.NewValue`, and `ahbsd.lib.ChangeEventArgs< T >.OldValue`.

#### 7.3.2.2 `ChangeEventArgs()` [2/3]

```
ahbsd.lib.ChangeEventArgs< T >.ChangeEventArgs (
    T oldValue,
    T newValue )
```

Constructor with the old and the new value.



## Parameters

<i>oldValue</i>	The old value.
<i>newValue</i>	The new value.

References [ahbsd.lib.ChangeEventArgs< T >.NewValue](#), and [ahbsd.lib.ChangeEventArgs< T >.OldValue](#).

**7.3.2.3 ChangeEventArgs()** [3/3]

```
ahbsd.lib.ChangeEventArgs< T >.ChangeEventArgs (
    T oldValue )
```

Constructor with the old value.

## Parameters

<i>oldValue</i>	The old value.
-----------------	----------------

References [ahbsd.lib.ChangeEventArgs< T >.NewValue](#), and [ahbsd.lib.ChangeEventArgs< T >.OldValue](#).

**7.3.3 Member Function Documentation****7.3.3.1 Equals()** [1/2]

```
bool ahbsd.lib.ChangeEventArgs< T >.Equals (
    IChangeEventArgs< T > other )
```

Finds out wheather an other object of type IChangeEventArgs<T> equals this object.

## Parameters

<i>other</i>	The other object.
--------------	-------------------

## Returns

TRUE if the other object equals this object, otherwise FALSE.

References [ahbsd.lib.ChangeEventArgs< T >.NewValue](#), [ahbsd.lib.IChangeEventArgs< T >.NewValue](#), [ahbsd.lib.ChangeEventArgs< T >.OldValue](#), and [ahbsd.lib.IChangeEventArgs< T >.OldValue](#).

Referenced by [ahbsd.lib.ChangeEventArgs< T >.Equals\(\)](#).

### 7.3.3.2 Equals() [2/2]

```
override bool ahbsd.lib.ChangeEventArgs< T >.Equals (
    object obj )
```

Finds out wheather an other object equals this object.

#### Parameters

<i>obj</i>	The other object.
------------	-------------------

#### Returns

TRUE if the other object equals this object, otherwise FALSE.

References [ahbsd.lib.ChangeEventArgs< T >.Equals\(\)](#).

### 7.3.3.3 GetHashCode()

```
override int ahbsd.lib.ChangeEventArgs< T >.GetHashCode ( )
```

Gets the GetHashCode of this object.

#### Returns

The GetHashCode.

References [ahbsd.lib.ChangeEventArgs< T >.NewValue](#), and [ahbsd.lib.ChangeEventArgs< T >.OldValue](#).

### 7.3.3.4 operator"!=()"

```
static bool ahbsd.lib.ChangeEventArgs< T >.operator!= (
    ChangeEventArgs< T > left,
    ChangeEventArgs< T > right ) [static]
```

Finds out, if two objects of type [ChangeEventArgs<T>](#) do not eaquals each other.

#### Parameters

<i>left</i>	The object on the left side.
<i>right</i>	The object on the right side.

#### Returns

TRUE if both objects are not eaqual, otherwise FALSE.

### 7.3.3.5 operator==()

```
static bool ahbsd.lib.ChangeEventArgs< T >.operator== (
    ChangeEventArgs< T > left,
    ChangeEventArgs< T > right ) [static]
```

Finds out, if two objects of type [ChangeEventArgs<T>](#) equals each other.

#### Parameters

<i>left</i>	The object on the left side.
<i>right</i>	The object on the right side.

#### Returns

TRUE if both objects are equal, otherwise FALSE.

### 7.3.3.6 SetNewValue()

```
void ahbsd.lib.ChangeEventArgs< T >.SetNewValue (
    T newValue )
```

Sets the new value.

#### Parameters

<i>newValue</i>	The new value.
-----------------	----------------

#### Exceptions

<i>Exception</i>	If the <a href="#">NewValue</a> was already set.
------------------	--

Implements [ahbsd.lib.IChangeEventArgs< T >](#).

References [ahbsd.lib.ChangeEventArgs< T >.NewValue](#).

### 7.3.3.7 ToString()

```
override string ahbsd.lib.ChangeEventArgs< T >.ToString ( )
```

Gets a string representation of the changed value.

**Returns**

A string representation of the changed value.

Implements [ahbsd.lib.IChangeEventArgs< T >](#).

References [ahbsd.lib.ChangeEventArgs< T >.NewValue](#), and [ahbsd.lib.ChangeEventArgs< T >.OldValue](#).

## 7.3.4 Property Documentation

### 7.3.4.1 NewValue

```
T ahbsd.lib.ChangeEventArgs< T >.NewValue [get]
```

Gets the new value.

The new value.

Referenced by [ahbsd.lib.ChangeEventArgs< T >.ChangeEventArgs\(\)](#), [ahbsd.lib.ChangeEventArgs< T >.Equals\(\)](#), [ahbsd.lib.ChangeEventArgs< T >.GetHashCode\(\)](#), [ahbsd.lib.ChangeEventArgs< T >.SetNewValue\(\)](#), and [ahbsd.lib.ChangeEventArgs< T >.ToString\(\)](#).

### 7.3.4.2 OldValue

```
T ahbsd.lib.ChangeEventArgs< T >.OldValue [get]
```

Gets the old value.

The old value.

Referenced by [ahbsd.lib.ChangeEventArgs< T >.ChangeEventArgs\(\)](#), [ahbsd.lib.ChangeEventArgs< T >.Equals\(\)](#), [ahbsd.lib.ChangeEventArgs< T >.GetHashCode\(\)](#), and [ahbsd.lib.ChangeEventArgs< T >.ToString\(\)](#).

The documentation for this class was generated from the following file:

- [ahbsd.lib/ChangeEventArgs.cs](#)

## 7.4 ahbsd.lib.Tools.ConsolePrintTable Class Reference

Static class to print a DataTable to console.

### Static Public Member Functions

- static void [Print](#) (DataTable table)  
*Prints a table on console.*

### 7.4.1 Detailed Description

Static class to print a DataTable to console.

### 7.4.2 Member Function Documentation

#### 7.4.2.1 Print()

```
static void ahbsd.lib.Tools.ConsolePrintTable.Print (
    DataTable table ) [static]
```

Prints a table on console.

#### Parameters

<i>table</i>	The table to print.
--------------	---------------------

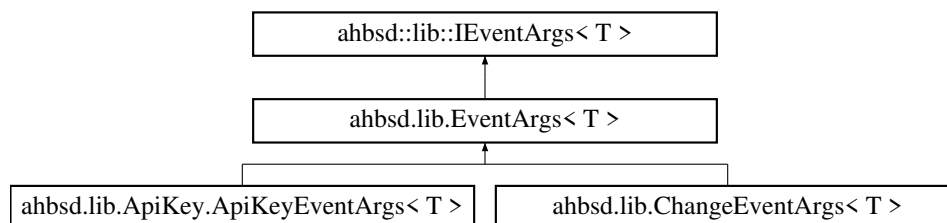
The documentation for this class was generated from the following file:

- ahbsd.lib/Tools/ConsolePrintTable.cs

## 7.5 ahbsd.lib.EventArgs< T > Class Template Reference

Generic [EventArgs](#).

Inheritance diagram for ahbsd.lib.EventArgs< T >:



### Public Member Functions

- [EventArgs](#) ()  
*Constructor without parameters.*
- [EventArgs](#) (T val)  
*Constructor with a value.*

## Properties

- `T Value` [get]  
*Gets a value.*

### 7.5.1 Detailed Description

Generic `EventArgs`.

Template Parameters

<code>T</code>	Type of <code>Value</code> .
----------------	------------------------------

### 7.5.2 Constructor & Destructor Documentation

#### 7.5.2.1 EventArgs() [1/2]

```
ahbsd.lib.EventArgs< T >.EventArgs ( )
```

Constructor without parameters.

References `ahbsd.lib.EventArgs< T >.Value`.

#### 7.5.2.2 EventArgs() [2/2]

```
ahbsd.lib.EventArgs< T >.EventArgs (  
    T val )
```

Constructor with a value.

Parameters

<code>val</code>	A value.
------------------	----------

References `ahbsd.lib.EventArgs< T >.Value`.

### 7.5.3 Property Documentation

### 7.5.3.1 Value

`T ahbsd.lib.EventArgs< T >.Value` [get]

Gets a value.

A value.

Referenced by `ahbsd.lib.EventArgs< T >.EventArgs()`.

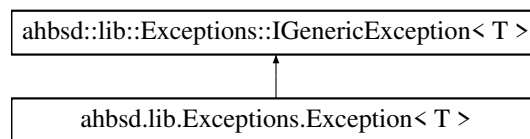
The documentation for this class was generated from the following file:

- `ahbsd.lib/EventArgs.cs`

## 7.6 ahbsd.lib.Exceptions.Exception< T > Class Template Reference

Class for a generic `Exception`, which additionally holds a value of `T`

Inheritance diagram for `ahbsd.lib.Exceptions.Exception< T >`:



### Protected Member Functions

- `Exception()` `Exception(T value)` `Exception(string message)` `Exception(string message, T value)` `Exception(string message, Exception innerException)` `Exception(SerializationInfo info, StreamingContext context)`

*Constructor without any parameters.*

### Properties

- `Exception(string message, T value, Exception innerException)` `T Value` [get]

*Constructor with a message, a value and an inner exception.*

### 7.6.1 Detailed Description

Class for a generic `Exception`, which additionally holds a value of `T`

Template Parameters

<code>T</code>	The type.
----------------	-----------

## 7.6.2 Constructor & Destructor Documentation

### 7.6.2.1 Exception()

```
Exception () Exception (T value) Exception (string message) Exception (string message, T value)
Exception (string message, Exception innerException) ahbsd.lib.Exceptions.Exception< T >.Exception
(
    SerializationInfo info,
    StreamingContext context ) [protected]
```

Constructor without any parameters.

Constructor with a value.

#### Parameters

<i>value</i>	The value.
--------------	------------

Constructor with a message.

#### Parameters

<i>message</i>	The Message.
----------------	--------------

Constructor with a message and a value.

#### Parameters

<i>message</i>	The Message.
<i>value</i>	The value.

Constructor with a message and an inner exception.

#### Parameters

<i>message</i>	The Message.
<i>innerException</i>	The inner <a href="#">Exception</a> .

Constructor with serialized data.

#### Parameters

<i>info</i>	The serialization info.
<i>context</i>	The straming content.

References [ahbsd.lib.Exceptions.Exception< T >.Value](#), and [ahbsd.lib.Exceptions.IGenericException< T >.Value](#).



### 7.6.3 Property Documentation

#### 7.6.3.1 Value

```
Exception (string message, T value, Exception innerException) T ahbsd.lib.Exceptions.Exception<
T >.Value [get]
```

Constructor with a message, a value and an inner exception.

##### Parameters

<i>message</i>	The Message.
<i>value</i>	The value.
<i>innerException</i>	The inner <a href="#">Exception</a> .

Gets the value of type T.

The value of type T.

Referenced by [ahbsd.lib.Exceptions.Exception< T >.Exception\(\)](#).

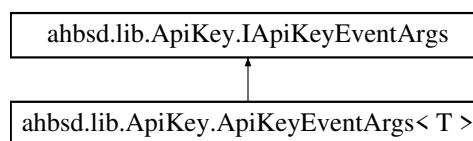
The documentation for this class was generated from the following file:

- [ahbsd.lib/Exceptions/Exception.cs](#)

## 7.7 ahbsd.lib.ApiKey.IApiKeyEventArgs Interface Reference

Interface for [ApiKeyEventArgs<T>](#).

Inheritance diagram for [ahbsd.lib.ApiKey.IApiKeyEventArgs](#):



### Properties

- int? [Index](#) [get]  
*Gets the index of the API-Key.*

#### 7.7.1 Detailed Description

Interface for [ApiKeyEventArgs<T>](#).

## 7.7.2 Property Documentation

### 7.7.2.1 Index

```
int? ahbsd.lib.ApiKey.IApiKeyEventArgs.Index [get]
```

Gets the index of the API-Key.

The index.

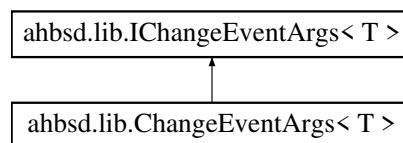
The documentation for this interface was generated from the following file:

- ahbsd.lib/ApiKey/IApiKeyEventArgs.cs

## 7.8 ahbsd.lib.IChangeEventArgs< T > Interface Template Reference

Interface for generic [EventArgs](#) for changing values.

Inheritance diagram for ahbsd.lib.IChangeEventArgs< T >:



### Public Member Functions

- void [SetNewValue](#) (T newValue)  
*Sets the new value.*
- string [ToString](#) ()  
*Gets a string representation of the changed value.*

### Properties

- T [OldValue](#) [get]  
*Gets the old value.*
- T [NewValue](#) [get]  
*Gets the new value.*

### 7.8.1 Detailed Description

Interface for generic [EventArgs](#) for changing values.

## Template Parameters

<i>T</i>	The type of the changing Values.
----------	----------------------------------

## 7.8.2 Member Function Documentation

### 7.8.2.1 SetNewValue()

```
void ahbsd.lib.IChangeEventArgs< T >.SetNewValue (
    T newValue )
```

Sets the new value.

## Parameters

<i>newValue</i>	The new value.
-----------------	----------------

## Exceptions

<i>Exception</i>	If the <a href="#">NewValue</a> was already set.
------------------	--

Implemented in [ahbsd.lib.ChangeEventArgs< T >](#).

### 7.8.2.2 ToString()

```
string ahbsd.lib.IChangeEventArgs< T >.ToString ( )
```

Gets a string representation of the changed value.

## Returns

A string representation of the changed value.

Implemented in [ahbsd.lib.ChangeEventArgs< T >](#).

## 7.8.3 Property Documentation

### 7.8.3.1 NewValue

`T ahbsd.lib.IChangeEventArgs< T >.NewValue [get]`

Gets the new value.

The new value.

Referenced by [ahbsd.lib.ChangeEventArgs< T >.Equals\(\)](#).

### 7.8.3.2 OldValue

`T ahbsd.lib.IChangeEventArgs< T >.OldValue [get]`

Gets the old value.

The old value.

Referenced by [ahbsd.lib.ChangeEventArgs< T >.Equals\(\)](#).

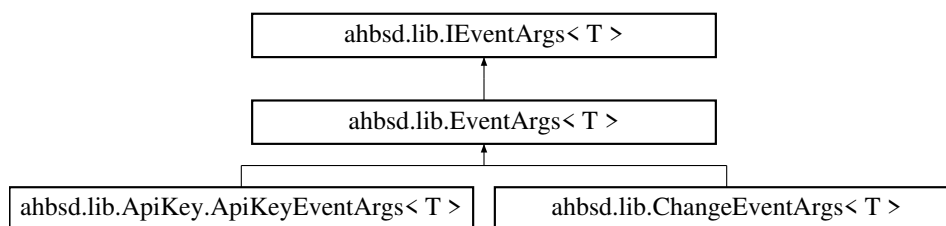
The documentation for this interface was generated from the following file:

- [ahbsd.lib/IChangeEventArgs.cs](#)

## 7.9 ahbsd.lib.IEventArgs< T > Interface Template Reference

Interface for generic [EventArgs](#).

Inheritance diagram for `ahbsd.lib.IEventArgs< T >`:



### Properties

- `T Value [get]`  
*Gets a value.*

### 7.9.1 Detailed Description

Interface for generic [EventArgs](#).

## Template Parameters

<i>T</i>	Type of Value.
----------	----------------

The difference to usual [EventArgs](#) is, that a generic value is added.

## 7.9.2 Property Documentation

### 7.9.2.1 Value

```
T ahbsd.lib.IEventArgs< T >.Value [get]
```

Gets a value.

A value.

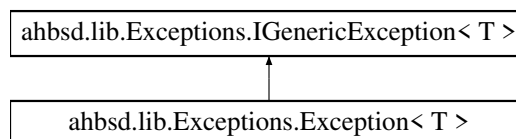
The documentation for this interface was generated from the following file:

- ahbsd.lib/IEventArgs.cs

## 7.10 ahbsd.lib.Exceptions.IGenericException< T > Interface Template Reference

Interface for generic exceptions.

Inheritance diagram for ahbsd.lib.Exceptions.IGenericException< T >:



### Properties

- T [Value](#) [get]  
*Gets the value of type T.*

### 7.10.1 Detailed Description

Interface for generic exceptions.

## Template Parameters

<i>T</i>	Selectable type.
----------	------------------

## 7.10.2 Property Documentation

### 7.10.2.1 Value

`T ahbsd.lib.Exceptions.IGenericException< T >.Value [get]`

Gets the value of type T.

The value of type T.

Referenced by `ahbsd.lib.Exceptions.Exception< T >.Exception()`.

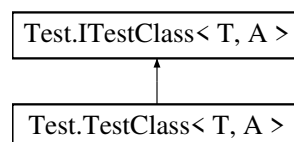
The documentation for this interface was generated from the following file:

- `ahbsd.lib/Exceptions/IGenericException.cs`

## 7.11 Test.ITestClass< T, A > Interface Template Reference

An interface for a class to demonstrate `ChangeEventArgs<T>` and `ChangeEventHandler<T>`.

Inheritance diagram for `Test.ITestClass< T, A >`:



### Properties

- `T Variable [get, set]`  
*Gets or sets a variable.*

### Events

- `ChangeEventHandler< T > OnChange`  
*Happenes when `Variable` has changed.*

### 7.11.1 Detailed Description

An interface for a class to demonstrate `ChangeEventArgs<T>` and `ChangeEventHandler<T>`.

## Template Parameters

<i>T</i>	Type of <a href="#">Variable</a> .
<i>A</i>	Type of API-Key.

## 7.11.2 Property Documentation

### 7.11.2.1 Variable

`T` [Test.ITestClass](#)< `T`, `A` >.Variable [get], [set]

Gets or sets a variable.

## 7.11.3 Event Documentation

### 7.11.3.1 OnChange

`ChangeEventHandler<T>` [Test.ITestClass](#)< `T`, `A` >.OnChange

Happenes when [Variable](#) has changed.

The documentation for this interface was generated from the following file:

- Test/ITestClass.cs

## 7.12 Test.Program Class Reference

The test program.

### 7.12.1 Detailed Description

The test program.

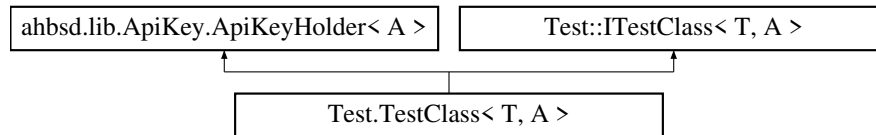
The documentation for this class was generated from the following file:

- Test/Program.cs

## 7.13 Test.TestClass< T, A > Class Template Reference

A class to demonstrate ChangeEventArgs<T> and ChangeEventHandler<T>.

Inheritance diagram for Test.TestClass< T, A >:



### Public Member Functions

- [TestClass](#) ()  
*Constructor without parameters.*
- [TestClass](#) (T v)  
*Constructor with a given variable.*
- [TestClass](#) (A apiKey)  
*Constructor with a given API-Key.*
- [TestClass](#) (T v, A apiKey)  
*Constructor with a given value and a given API-Key.*

### Properties

- T?? [Variable](#) [get, set]  
*Gets or sets a variable.*

### Events

- ChangeEventHandler< T > [OnChange](#)  
*Happenes when [Variable](#) has changed.*

### Additional Inherited Members

#### 7.13.1 Detailed Description

A class to demonstrate ChangeEventArgs<T> and ChangeEventHandler<T>.

##### Template Parameters

<i>T</i>	Type of <a href="#">Variable</a> .
<i>A</i>	Type of API-Key.



## 7.13.2 Constructor & Destructor Documentation

### 7.13.2.1 TestClass() [1/4]

```
Test.TestClass< T, A >.TestClass ( )
```

Constructor without parameters.

References [ahbsd.lib.ApiKey.ApiKeyHolder< A >.OnApiKeyAdded](#).

### 7.13.2.2 TestClass() [2/4]

```
Test.TestClass< T, A >.TestClass (
    T v )
```

Constructor with a given variable.

#### Parameters

<i>v</i>	The given variable.
----------	---------------------

References [ahbsd.lib.ApiKey.ApiKeyHolder< A >.OnApiKeyAdded](#).

### 7.13.2.3 TestClass() [3/4]

```
Test.TestClass< T, A >.TestClass (
    A apiKey )
```

Constructor with a given API-Key.

#### Parameters

<i>apiKey</i>	The given API-Key.
---------------	--------------------

References [ahbsd.lib.ApiKey.ApiKeyHolder< A >.OnApiKeyAdded](#).

### 7.13.2.4 TestClass() [4/4]

```
Test.TestClass< T, A >.TestClass (
    T v,
    A apiKey )
```

Constructor with a given value and a given API-Key.

#### Parameters

<i>v</i>	The given variable.
<i>apiKey</i>	The given API-Key.

References [ahbsd.lib.ApiKey.ApiKeyHolder< A >.OnApiKeyAdded](#).

## 7.13.3 Property Documentation

### 7.13.3.1 Variable

```
T?? Test.TestClass< T, A >.Variable [get], [set]
```

Gets or sets a variable.

## 7.13.4 Event Documentation

### 7.13.4.1 OnChange

```
ChangeEventHandler<T> Test.TestClass< T, A >.OnChange
```

Happenes when [Variable](#) has changed.

The documentation for this class was generated from the following file:

- Test/TestClass.cs

# Index

- ahbsd, [13](#)
- ahbsd.lib, [13](#)
  - ChangeEventHandler< T >, [13](#)
- ahbsd.lib.ApiKey, [14](#)
  - ApiKeyEventHandler< T >, [14](#)
- ahbsd.lib.ApiKey.ApiKeyEventArgs< T >, [17](#)
  - ApiKeyEventArgs, [18](#)
  - Index, [18](#)
- ahbsd.lib.ApiKey.ApiKeyHolder< T >, [19](#)
  - ApiKey, [24](#)
  - ApiKeyHolder, [21](#)
  - Equals, [21](#), [22](#)
  - FindApiKey, [22](#)
  - GetApiKey, [22](#)
  - GetHashCode, [23](#)
  - KnownApiKeys, [24](#)
  - OnApiKeyAdded, [25](#)
  - operator!=, [23](#)
  - operator==, [24](#)
- ahbsd.lib.ApiKey.IApiKeyEventArgs, [35](#)
  - Index, [36](#)
- ahbsd.lib.ChangeEventArgs< T >, [25](#)
  - ChangeEventArgs, [26](#), [27](#)
  - Equals, [27](#)
  - GetHashCode, [28](#)
  - NewValue, [30](#)
  - OldValue, [30](#)
  - operator!=, [28](#)
  - operator==, [29](#)
  - SetNewValue, [29](#)
  - ToString, [29](#)
- ahbsd.lib.EventArgs< T >, [31](#)
  - EventArgs, [32](#)
  - Value, [32](#)
- ahbsd.lib.Exceptions, [15](#)
- ahbsd.lib.Exceptions.Exception< T >, [33](#)
  - Exception, [34](#)
  - Value, [35](#)
- ahbsd.lib.Exceptions.IGenericException< T >, [39](#)
  - Value, [40](#)
- ahbsd.lib.IChangeEventArgs< T >, [36](#)
  - NewValue, [37](#)
  - OldValue, [38](#)
  - SetNewValue, [37](#)
  - ToString, [37](#)
- ahbsd.lib.IEventArgs< T >, [38](#)
  - Value, [39](#)
- ahbsd.lib.Tools, [15](#)
- ahbsd.lib.Tools.ConsolePrintTable, [30](#)
- Print, [31](#)
- ApiKey
  - ahbsd.lib.ApiKey.ApiKeyHolder< T >, [24](#)
- ApiKeyEventArgs
  - ahbsd.lib.ApiKey.ApiKeyEventArgs< T >, [18](#)
- ApiKeyEventHandler< T >
  - ahbsd.lib.ApiKey, [14](#)
- ApiKeyHolder
  - ahbsd.lib.ApiKey.ApiKeyHolder< T >, [21](#)
- ChangeEventArgs
  - ahbsd.lib.ChangeEventArgs< T >, [26](#), [27](#)
- ChangeEventHandler< T >
  - ahbsd.lib, [13](#)
- Equals
  - ahbsd.lib.ApiKey.ApiKeyHolder< T >, [21](#), [22](#)
  - ahbsd.lib.ChangeEventArgs< T >, [27](#)
- EventArgs
  - ahbsd.lib.EventArgs< T >, [32](#)
- Exception
  - ahbsd.lib.Exceptions.Exception< T >, [34](#)
- FindApiKey
  - ahbsd.lib.ApiKey.ApiKeyHolder< T >, [22](#)
- GetApiKey
  - ahbsd.lib.ApiKey.ApiKeyHolder< T >, [22](#)
- GetHashCode
  - ahbsd.lib.ApiKey.ApiKeyHolder< T >, [23](#)
  - ahbsd.lib.ChangeEventArgs< T >, [28](#)
- Index
  - ahbsd.lib.ApiKey.ApiKeyEventArgs< T >, [18](#)
  - ahbsd.lib.ApiKey.IApiKeyEventArgs, [36](#)
- KnownApiKeys
  - ahbsd.lib.ApiKey.ApiKeyHolder< T >, [24](#)
- NewValue
  - ahbsd.lib.ChangeEventArgs< T >, [30](#)
  - ahbsd.lib.IChangeEventArgs< T >, [37](#)
- OldValue
  - ahbsd.lib.ChangeEventArgs< T >, [30](#)
  - ahbsd.lib.IChangeEventArgs< T >, [38](#)
- OnApiKeyAdded
  - ahbsd.lib.ApiKey.ApiKeyHolder< T >, [25](#)
- OnChange
  - Test.ITestClass< T, A >, [41](#)
  - Test.TestClass< T, A >, [44](#)

operator!=  
    ahbsd.lib.ApiKey.ApiKeyHolder< T >, [23](#)  
    ahbsd.lib.ChangeEventArgs< T >, [28](#)  
operator==  
    ahbsd.lib.ApiKey.ApiKeyHolder< T >, [24](#)  
    ahbsd.lib.ChangeEventArgs< T >, [29](#)

Print  
    ahbsd.lib.Tools.ConsolePrintTable, [31](#)

SetNewValue  
    ahbsd.lib.ChangeEventArgs< T >, [29](#)  
    ahbsd.lib.IChangeEventArgs< T >, [37](#)

Test, [15](#)  
Test.ITestClass< T, A >, [40](#)  
    OnChange, [41](#)  
    Variable, [41](#)  
Test.Program, [41](#)  
Test.TestClass< T, A >, [42](#)  
    OnChange, [44](#)  
    TestClass, [43](#)  
    Variable, [44](#)

TestClass  
    Test.TestClass< T, A >, [43](#)

ToString  
    ahbsd.lib.ChangeEventArgs< T >, [29](#)  
    ahbsd.lib.IChangeEventArgs< T >, [37](#)

Value  
    ahbsd.lib.EventArgs< T >, [32](#)  
    ahbsd.lib.Exceptions.Exception< T >, [35](#)  
    ahbsd.lib.Exceptions.IGenericException< T >, [40](#)  
    ahbsd.lib.IEventArgs< T >, [39](#)

Variable  
    Test.ITestClass< T, A >, [41](#)  
    Test.TestClass< T, A >, [44](#)