

# libAHcutil Reference Manual

## 0.1

Generated by Doxygen 1.4.4

Fri Apr 7 01:18:06 2006



# Contents

<b>1</b>	<b>libAHcutil Directory Hierarchy</b>	<b>1</b>
1.1	libAHcutil Directories . . . . .	1
<b>2</b>	<b>libAHcutil Namespace Index</b>	<b>3</b>
2.1	libAHcutil Namespace List . . . . .	3
<b>3</b>	<b>libAHcutil Class Index</b>	<b>5</b>
3.1	libAHcutil Class List . . . . .	5
<b>4</b>	<b>libAHcutil File Index</b>	<b>7</b>
4.1	libAHcutil File List . . . . .	7
<b>5</b>	<b>libAHcutil Directory Documentation</b>	<b>9</b>
5.1	AH/ Directory Reference . . . . .	9
5.2	AH/CUtil/ Directory Reference . . . . .	10
<b>6</b>	<b>libAHcutil Namespace Documentation</b>	<b>11</b>
6.1	AH Namespace Reference . . . . .	11
<b>7</b>	<b>libAHcutil Class Documentation</b>	<b>13</b>
7.1	AH::CUtilSecureArray Class Reference . . . . .	13
7.2	AH::CUtilSecureString Class Reference . . . . .	15
<b>8</b>	<b>libAHcutil File Documentation</b>	<b>17</b>
8.1	AH/CUtil/SecureArray.h File Reference . . . . .	17
8.2	AH/CUtil/SecureString.h File Reference . . . . .	18



# Chapter 1

## libAHcutil Directory Hierarchy

### 1.1 libAHcutil Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

AH . . . . .	9
CUtil . . . . .	10



## Chapter 2

# libAHcutil Namespace Index

### 2.1 libAHcutil Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">AH</a>	.....	<a href="#">11</a>
--------------------	-------	--------------------





## Chapter 3

# libAHcutil Class Index

### 3.1 libAHcutil Class List

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

<a href="#">AH::CUtlSecureArray</a> . . . . .	13
<a href="#">AH::CUtlSecureString</a> . . . . .	15



# Chapter 4

## libAHcutil File Index

### 4.1 libAHcutil File List

Here is a list of all files with brief descriptions:

AH/CUtil/SecureArray.h . . . . .	17
AH/CUtil/SecureString.h . . . . .	18



## Chapter 5

# libAHcutil Directory Documentation

### 5.1 AH/ Directory Reference

#### Directories

- directory [CUtil](#)

## 5.2 AH/CUtil/ Directory Reference

### Files

- file [SecureArray.h](#)
- file [SecureString.h](#)

## Chapter 6

# libAHcutil Namespace Documentation

### 6.1 AH Namespace Reference

#### Classes

- class [CUtilSecureArray](#)
- class [CUtilSecureString](#)





## Chapter 7

# libAHcutil Class Documentation

### 7.1 AH::CUtilSecureArray Class Reference

```
#include <SecureArray.h>
```

#### Static Public Member Functions

- `template<class T> static size_t getNumEntries (const T &array)`
- `template<class Dest, class Source> static void setVal (Dest &array, const Source &val)`
- `template<class Dest, class Source, class Default> static int copyAll (Dest &d, const Source &s, const Default &defval)`
- `template<class Dest, class Source> static int copy (Dest &d, const Source &s)`

#### 7.1.1 Detailed Description

utilities for secure handling of C arrays in C++ code.

Class [CUtilSecureArray](#) offers some functions to handle C style arrays (not pointers!) of arbitrary types in a secure way.

All functions are static and inline and can be used without object and without linking libCUtil.

#### 7.1.2 Member Function Documentation

**7.1.2.1** `template<class Dest, class Source> static int AH::CUtilSecureArray::copy (Dest & d, const Source & s) [inline, static]`

copy all array entries from other array.

[copy\(\)](#) copies all array entries from 's' to 'd', but no more entries than 'd' can store. If 's' contains less entries than 'd' can store, then the remaining entries of 'd' are not set and keep their previous state.

[copy\(\)](#) returns the difference of the entry numbers, that is the number of 's' array entries minus the number of 'd' array entries. So ideally [copy\(\)](#) returns 0 if the arrays have equal number of entries. If the return value is negative then the absolute of this number is the number of entries in 'd' that are not set. If the return value is positive then it is the number of entries in 's' that have not been copied.

**Parameters:***dest* the destination array*source* the source array**Returns:**

number of 's' array entries minus number of 'd' array entries

**7.1.2.2** `template<class Dest, class Source, class Default> static int AH::CUtilSecureArray::copyAll (Dest & d, const Source & s, const Default & defval) [inline, static]`

copy all array entries from other array.

`copyAll()` copies all array entries from 's' to 'd', but no more entries than 'd' can store. If 's' contains less entries than 'd' can store, then the remaining entries of 'd' are set to the default value 'defval'.

`copyAll()` returns the difference of the entry numbers, that is the number of 's' array entries minus the number of 'd' array entries. So ideally `copyAll()` returns 0 if the arrays have equal number of entries. If the return value is negative then the absolute of this number is the number of entries in 'd' that have been set to 'defval'. If the return value is positive then it is the number of entries in 's' that have not been copied.

**Parameters:***d* the destination array*s* the source array*defval* the default value**Returns:**

number of 's' array entries minus number of 'd' array entries

**7.1.2.3** `template<class T> static size_t AH::CUtilSecureArray::getNumEntries (const T & array) [inline, static]`

return number of array entries.

**Parameters:***array* the array**Returns:**

number of array entries

**7.1.2.4** `template<class Dest, class Source> static void AH::CUtilSecureArray::setVal (Dest & array, const Source & val) [inline, static]`

set all array entries to specified value.

**Parameters:***array* the array*val* the value each array entry is set to

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

- [AH/CUtil/SecureArray.h](#)

## 7.2 AH::CUtlSecureString Class Reference

```
#include <SecureString.h>
```

### Static Public Member Functions

- static char \* [strnCpy](#) (char \*dest, const char \*src, size\_t n)
- template<class CharArray> static char \* [strCpy](#) (CharArray d, const char \*src)

### 7.2.1 Detailed Description

utilities for secure handling of C character arrays in C++ code.

Class [CUtlSecureString](#) offers some functions to handle C style character arrays in a secure way.

All functions are static and inline and can be used without object.

### 7.2.2 Member Function Documentation

**7.2.2.1** `template<class CharArray> static char* AH::CUtlSecureString::strCpy (CharArray d, const char * src) [inline, static]`

copy character arrays.

The [strCpy\(\)](#) function copies the string pointed to by 'src' (including the terminating "" character) to the array 'dest'. It is guaranteed that no more characters are written into 'dest' than 'dest' can keep, and it is guaranteed that 'dest' is always NULL terminated. The character arrays may not overlap.

#### Parameters:

- dest* the destination character array (no pointer)  
*source* pointer to the source character array

#### Returns:

pointer to 'dest'

**7.2.2.2** `static char* AH::CUtlSecureString::strnCpy (char * dest, const char * src, size_t n) [static]`

copy character arrays.

The [strnCpy\(\)](#) function copies the string pointed to by 'src' (including the terminating "" character) to the array pointed to by 'dest'. No more than 'n'-1 bytes of 'src' are copied, since 'dest' is guaranteed to be always NULL terminated. The character arrays may not overlap.

#### Parameters:

- dest* pointer to the destination character array  
*source* pointer to the source character array  
*n* the size of the destination char array

#### Returns:

pointer to 'dest'

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

- AH/CUtil/[SecureString.h](#)

## Chapter 8

# libAHcutil File Documentation

### 8.1 AH/CUtil/SecureArray.h File Reference

```
#include <AH/Config/Platform.h>
#include <sys/types.h>
```

#### Namespaces

- namespace [AH](#)

#### Classes

- class [AH::CUtilSecureArray](#)

## 8.2 AH/CUtil/SecureString.h File Reference

```
#include <AH/Config/Platform.h>
#include <sys/types.h>
```

### Namespaces

- namespace [AH](#)

### Classes

- class [AH::CUtilSecureString](#)

# Index

- AH, [11](#)
- AH/ Directory Reference, [9](#)
- AH/CUtil/ Directory Reference, [10](#)
- AH/CUtil/SecureArray.h, [17](#)
- AH/CUtil/SecureString.h, [18](#)
- AH::CUtilSecureArray, [13](#)
- AH::CUtilSecureArray
  - copy, [13](#)
  - copyAll, [14](#)
  - getNumEntries, [14](#)
  - setVal, [14](#)
- AH::CUtilSecureString, [15](#)
- AH::CUtilSecureString
  - strCpy, [15](#)
  - strnCpy, [15](#)
  
- copy
  - AH::CUtilSecureArray, [13](#)
- copyAll
  - AH::CUtilSecureArray, [14](#)
  
- getNumEntries
  - AH::CUtilSecureArray, [14](#)
  
- setVal
  - AH::CUtilSecureArray, [14](#)
- strCpy
  - AH::CUtilSecureString, [15](#)
- strnCpy
  - AH::CUtilSecureString, [15](#)