

## HashMap

Generated by Doxygen 1.9.8



---

<b>1 File Index</b>	<b>1</b>
1.1 File List . . . . .	1
<b>2 File Documentation</b>	<b>3</b>
2.1 <code>hashset.h</code> File Reference . . . . .	3
2.1.1 Typedef Documentation . . . . .	4
2.1.1.1 <code>hash</code> . . . . .	4
2.1.1.2 <code>HashSet</code> . . . . .	4
2.1.2 Function Documentation . . . . .	4
2.1.2.1 <code>hs_add()</code> . . . . .	4
2.1.2.2 <code>hs_contains()</code> . . . . .	4
2.1.2.3 <code>hs_create()</code> . . . . .	5
2.1.2.4 <code>hs_destroy()</code> . . . . .	5
2.1.2.5 <code>hs_remove()</code> . . . . .	5
2.1.2.6 <code>hs_size()</code> . . . . .	6
2.2 <code>hashset.h</code> . . . . .	6
<b>Index</b>	<b>7</b>



# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">hashset.h</a> . . . . .	3
-------------------------------------	---



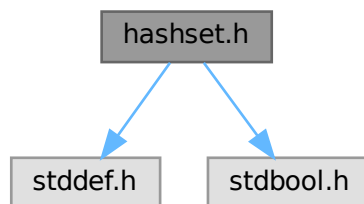
# Chapter 2

## File Documentation

### 2.1 `hashset.h` File Reference

```
#include <stddef.h>
#include <stdbool.h>
```

Include dependency graph for `hashset.h`:



#### Typedefs

- typedef struct `HashSet` \* `HashSet`
- typedef `size_t`(\* `hash`) (const void \*key, `size_t` key\_size)

#### Functions

- `HashSet` `hs_create` (`size_t` hs\_capacity, `size_t` key\_size, `hash` hash\_func)  
*Creates a new HashSet with a hash function.*
- int `hs_destroy` (`HashSet` hs)  
*Destroys the HashSet.*
- bool `hs_contains` (`HashSet` hs, const void \*key)  
*Tests if the HashSet contains the specified key.*
- int `hs_add` (`HashSet` hs, const void \*key)  
*Adds a new key to the HashSet.*
- `size_t` `hs_size` (`HashSet` hs)  
*Returns the size of the HashSet.*
- int `hs_remove` (`HashSet` hs, const void \*key)  
*Removes the key from the HashSet.*

## 2.1.1 Typedef Documentation

### 2.1.1.1 hash

```
typedef size_t(* hash) (const void *key, size_t key_size)
```

### 2.1.1.2 HashSet

```
typedef struct HashSet* HashSet
```

## 2.1.2 Function Documentation

### 2.1.2.1 hs\_add()

```
int hs_add (
    HashSet hs,
    const void * key )
```

Adds a new key to the HashSet.

#### Parameters

<i>hs</i>	The HashSet
<i>key</i>	The key

#### Returns

Success code

### 2.1.2.2 hs\_contains()

```
bool hs_contains (
    HashSet hs,
    const void * key )
```

Tests if the HashSet contains the specified key.

#### Parameters

<i>hs</i>	The HashSet
<i>key</i>	The key

#### Returns

True or Falsehood



### 2.1.2.3 `hs_create()`

```
HashSet hs_create (
    size_t hs_capacity,
    size_t key_size,
    hash hash_func )
```

Creates a new HashSet with a hash function.

#### Parameters

<i>hs_capacity</i>	The initial capacity of the HashSet
<i>key_size</i>	The sizeof value of the key
<i>hash_func</i>	a custom hash function. Pass NULL for generic hashing.

#### Returns

HashSet

### 2.1.2.4 `hs_destroy()`

```
int hs_destroy (
    HashSet hs )
```

Destroys the HashSet.

#### Parameters

<i>hs</i>	The HashSet
-----------	-------------

#### Returns

Success code

### 2.1.2.5 `hs_remove()`

```
int hs_remove (
    HashSet hs,
    const void * key )
```

Removes the key from the HashSet.

#### Parameters

<i>hs</i>	The HashSet
<i>key</i>	The key

**Returns**

Success code

**2.1.2.6 hs\_size()**

```
size_t hs_size (
    HashSet hs )
```

Returns the size of the HashSet.

**Parameters**

<i>hs</i>	The HashSet
-----------	-------------

**Returns**

The size

**2.2 hashset.h**

[Go to the documentation of this file.](#)

```
00001 //
00002 // Created by Emanuel on 03.09.2024.
00003 //
00004
00005 #ifndef HASHSET_H
00006 #define HASHSET_H
00007
00008 #include <stddef.h>
00009 #include <stdbool.h>
00010
00011 typedef struct HashSet *HashSet;
00012 typedef size_t (*hash)(const void *key, size_t key_size);
00013
00022 HashSet hs_create(size_t hs_capacity, size_t key_size, hash hash_func);
00023
00024
00031 int hs_destroy(HashSet hs);
00032
00033
00041 bool hs_contains(HashSet hs, const void *key);
00042
00050 int hs_add(HashSet hs, const void *key);
00051
00058 size_t hs_size(HashSet hs);
00059
00067 int hs_remove(HashSet hs, const void *key);
00068
00069 #endif //HASHSET_H
```

# Index

- hash
  - hashset.h, [4](#)
- HashSet
  - hashset.h, [4](#)
- hashset.h, [3](#)
  - hash, [4](#)
  - HashSet, [4](#)
  - hs\_add, [4](#)
  - hs\_contains, [4](#)
  - hs\_create, [4](#)
  - hs\_destroy, [5](#)
  - hs\_remove, [5](#)
  - hs\_size, [6](#)
- hs\_add
  - hashset.h, [4](#)
- hs\_contains
  - hashset.h, [4](#)
- hs\_create
  - hashset.h, [4](#)
- hs\_destroy
  - hashset.h, [5](#)
- hs\_remove
  - hashset.h, [5](#)
- hs\_size
  - hashset.h, [6](#)