| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/HashMap.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/GregorianCalendar.html)   [**NEXT CLASS**](http://docs.google.com/java/util/HashSet.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/HashMap.html)    [**NO FRAMES**](http://docs.google.com/HashMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#1ksv4uv) |

## **java.util**

Class HashMap<K,V>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.util.AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<K,V>  
 **java.util.HashMap<K,V>**

**Type Parameters:**K - the type of keys maintained by this mapV - the type of mapped values **All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Map](http://docs.google.com/java/util/Map.html)<K,V> **Direct Known Subclasses:** [LinkedHashMap](http://docs.google.com/java/util/LinkedHashMap.html), [PrinterStateReasons](http://docs.google.com/javax/print/attribute/standard/PrinterStateReasons.html)

public class **HashMap<K,V>**extends [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<K,V>implements [Map](http://docs.google.com/java/util/Map.html)<K,V>, [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

Hash table based implementation of the Map interface. This implementation provides all of the optional map operations, and permits null values and the null key. (The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls.) This class makes no guarantees as to the order of the map; in particular, it does not guarantee that the order will remain constant over time.

This implementation provides constant-time performance for the basic operations (get and put), assuming the hash function disperses the elements properly among the buckets. Iteration over collection views requires time proportional to the "capacity" of the HashMap instance (the number of buckets) plus its size (the number of key-value mappings). Thus, it's very important not to set the initial capacity too high (or the load factor too low) if iteration performance is important.

An instance of HashMap has two parameters that affect its performance: *initial capacity* and *load factor*. The *capacity* is the number of buckets in the hash table, and the initial capacity is simply the capacity at the time the hash table is created. The *load factor* is a measure of how full the hash table is allowed to get before its capacity is automatically increased. When the number of entries in the hash table exceeds the product of the load factor and the current capacity, the hash table is *rehashed* (that is, internal data structures are rebuilt) so that the hash table has approximately twice the number of buckets.

As a general rule, the default load factor (.75) offers a good tradeoff between time and space costs. Higher values decrease the space overhead but increase the lookup cost (reflected in most of the operations of the HashMap class, including get and put). The expected number of entries in the map and its load factor should be taken into account when setting its initial capacity, so as to minimize the number of rehash operations. If the initial capacity is greater than the maximum number of entries divided by the load factor, no rehash operations will ever occur.

If many mappings are to be stored in a HashMap instance, creating it with a sufficiently large capacity will allow the mappings to be stored more efficiently than letting it perform automatic rehashing as needed to grow the table.

**Note that this implementation is not synchronized.** If multiple threads access a hash map concurrently, and at least one of the threads modifies the map structurally, it *must* be synchronized externally. (A structural modification is any operation that adds or deletes one or more mappings; merely changing the value associated with a key that an instance already contains is not a structural modification.) This is typically accomplished by synchronizing on some object that naturally encapsulates the map. If no such object exists, the map should be "wrapped" using the [Collections.synchronizedMap](http://docs.google.com/java/util/Collections.html#synchronizedMap(java.util.Map)) method. This is best done at creation time, to prevent accidental unsynchronized access to the map:

Map m = Collections.synchronizedMap(new HashMap(...));

The iterators returned by all of this class's "collection view methods" are *fail-fast*: if the map is structurally modified at any time after the iterator is created, in any way except through the iterator's own remove method, the iterator will throw a [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html). Thus, in the face of concurrent modification, the iterator fails quickly and cleanly, rather than risking arbitrary, non-deterministic behavior at an undetermined time in the future.

Note that the fail-fast behavior of an iterator cannot be guaranteed as it is, generally speaking, impossible to make any hard guarantees in the presence of unsynchronized concurrent modification. Fail-fast iterators throw ConcurrentModificationException on a best-effort basis. Therefore, it would be wrong to write a program that depended on this exception for its correctness: *the fail-fast behavior of iterators should be used only to detect bugs.*

This class is a member of the  [Java Collections Framework](http://docs.google.com/technotes/guides/collections/index.html).

**Since:** 1.2 **See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Collection](http://docs.google.com/java/util/Collection.html), [Map](http://docs.google.com/java/util/Map.html), [TreeMap](http://docs.google.com/java/util/TreeMap.html), [Hashtable](http://docs.google.com/java/util/Hashtable.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.util.HashMap)

| **Nested Class Summary** | |
| --- | --- |

| **Nested classes/interfaces inherited from class java.util.**[**AbstractMap**](http://docs.google.com/java/util/AbstractMap.html) |
| --- |
| [AbstractMap.SimpleEntry](http://docs.google.com/java/util/AbstractMap.SimpleEntry.html)<[K](http://docs.google.com/java/util/AbstractMap.SimpleEntry.html),[V](http://docs.google.com/java/util/AbstractMap.SimpleEntry.html)>, [AbstractMap.SimpleImmutableEntry](http://docs.google.com/java/util/AbstractMap.SimpleImmutableEntry.html)<[K](http://docs.google.com/java/util/AbstractMap.SimpleImmutableEntry.html),[V](http://docs.google.com/java/util/AbstractMap.SimpleImmutableEntry.html)> |

| **Constructor Summary** | |
| --- | --- |
| [**HashMap**](http://docs.google.com/java/util/HashMap.html#HashMap())()            Constructs an empty HashMap with the default initial capacity (16) and the default load factor (0.75). |
| [**HashMap**](http://docs.google.com/java/util/HashMap.html#HashMap(int))(int initialCapacity)            Constructs an empty HashMap with the specified initial capacity and the default load factor (0.75). |
| [**HashMap**](http://docs.google.com/java/util/HashMap.html#HashMap(int,%20float))(int initialCapacity, float loadFactor)            Constructs an empty HashMap with the specified initial capacity and load factor. |
| [**HashMap**](http://docs.google.com/java/util/HashMap.html#HashMap(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/HashMap.html),? extends [V](http://docs.google.com/java/util/HashMap.html)> m)            Constructs a new HashMap with the same mappings as the specified Map. |

| **Method Summary** | |
| --- | --- |
| void | [**clear**](http://docs.google.com/java/util/HashMap.html#clear())()            Removes all of the mappings from this map. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/util/HashMap.html#clone())()            Returns a shallow copy of this HashMap instance: the keys and values themselves are not cloned. |
| boolean | [**containsKey**](http://docs.google.com/java/util/HashMap.html#containsKey(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Returns true if this map contains a mapping for the specified key. |
| boolean | [**containsValue**](http://docs.google.com/java/util/HashMap.html#containsValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) value)            Returns true if this map maps one or more keys to the specified value. |
| [Set](http://docs.google.com/java/util/Set.html)<[Map.Entry](http://docs.google.com/java/util/Map.Entry.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>> | [**entrySet**](http://docs.google.com/java/util/HashMap.html#entrySet())()            Returns a [Set](http://docs.google.com/java/util/Set.html) view of the mappings contained in this map. |
| [V](http://docs.google.com/java/util/HashMap.html) | [**get**](http://docs.google.com/java/util/HashMap.html#get(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key. |
| boolean | [**isEmpty**](http://docs.google.com/java/util/HashMap.html#isEmpty())()            Returns true if this map contains no key-value mappings. |
| [Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/HashMap.html)> | [**keySet**](http://docs.google.com/java/util/HashMap.html#keySet())()            Returns a [Set](http://docs.google.com/java/util/Set.html) view of the keys contained in this map. |
| [V](http://docs.google.com/java/util/HashMap.html) | [**put**](http://docs.google.com/java/util/HashMap.html#put(K,%20V))([K](http://docs.google.com/java/util/HashMap.html) key, [V](http://docs.google.com/java/util/HashMap.html) value)            Associates the specified value with the specified key in this map. |
| void | [**putAll**](http://docs.google.com/java/util/HashMap.html#putAll(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/HashMap.html),? extends [V](http://docs.google.com/java/util/HashMap.html)> m)            Copies all of the mappings from the specified map to this map. |
| [V](http://docs.google.com/java/util/HashMap.html) | [**remove**](http://docs.google.com/java/util/HashMap.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Removes the mapping for the specified key from this map if present. |
| int | [**size**](http://docs.google.com/java/util/HashMap.html#size())()            Returns the number of key-value mappings in this map. |
| [Collection](http://docs.google.com/java/util/Collection.html)<[V](http://docs.google.com/java/util/HashMap.html)> | [**values**](http://docs.google.com/java/util/HashMap.html#values())()            Returns a [Collection](http://docs.google.com/java/util/Collection.html) view of the values contained in this map. |

| **Methods inherited from class java.util.**[**AbstractMap**](http://docs.google.com/java/util/AbstractMap.html) |
| --- |
| [equals](http://docs.google.com/java/util/AbstractMap.html#equals(java.lang.Object)), [hashCode](http://docs.google.com/java/util/AbstractMap.html#hashCode()), [toString](http://docs.google.com/java/util/AbstractMap.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Methods inherited from interface java.util.**[**Map**](http://docs.google.com/java/util/Map.html) |
| --- |
| [equals](http://docs.google.com/java/util/Map.html#equals(java.lang.Object)), [hashCode](http://docs.google.com/java/util/Map.html#hashCode()) |

| **Constructor Detail** |
| --- |

### HashMap

public **HashMap**(int initialCapacity,  
 float loadFactor)

Constructs an empty HashMap with the specified initial capacity and load factor.

**Parameters:**initialCapacity - the initial capacityloadFactor - the load factor **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity is negative or the load factor is nonpositive

### HashMap

public **HashMap**(int initialCapacity)

Constructs an empty HashMap with the specified initial capacity and the default load factor (0.75).

**Parameters:**initialCapacity - the initial capacity. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity is negative.

### HashMap

public **HashMap**()

Constructs an empty HashMap with the default initial capacity (16) and the default load factor (0.75).

### HashMap

public **HashMap**([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/HashMap.html),? extends [V](http://docs.google.com/java/util/HashMap.html)> m)

Constructs a new HashMap with the same mappings as the specified Map. The HashMap is created with default load factor (0.75) and an initial capacity sufficient to hold the mappings in the specified Map.

**Parameters:**m - the map whose mappings are to be placed in this map **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified map is null

| **Method Detail** |
| --- |

### size

public int **size**()

Returns the number of key-value mappings in this map.

**Specified by:**[size](http://docs.google.com/java/util/Map.html#size()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[size](http://docs.google.com/java/util/AbstractMap.html#size()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Returns:**the number of key-value mappings in this map

### isEmpty

public boolean **isEmpty**()

Returns true if this map contains no key-value mappings.

**Specified by:**[isEmpty](http://docs.google.com/java/util/Map.html#isEmpty()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[isEmpty](http://docs.google.com/java/util/AbstractMap.html#isEmpty()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Returns:**true if this map contains no key-value mappings

### get

public [V](http://docs.google.com/java/util/HashMap.html) **get**([Object](http://docs.google.com/java/lang/Object.html) key)

Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.

More formally, if this map contains a mapping from a key k to a value v such that (key==null ? k==null : key.equals(k)), then this method returns v; otherwise it returns null. (There can be at most one such mapping.)

A return value of null does not *necessarily* indicate that the map contains no mapping for the key; it's also possible that the map explicitly maps the key to null. The [containsKey](http://docs.google.com/java/util/HashMap.html#containsKey(java.lang.Object)) operation may be used to distinguish these two cases.

**Specified by:**[get](http://docs.google.com/java/util/Map.html#get(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[get](http://docs.google.com/java/util/AbstractMap.html#get(java.lang.Object)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Parameters:**key - the key whose associated value is to be returned **Returns:**the value to which the specified key is mapped, or null if this map contains no mapping for the key**See Also:**[put(Object, Object)](http://docs.google.com/java/util/HashMap.html#put(K,%20V))

### containsKey

public boolean **containsKey**([Object](http://docs.google.com/java/lang/Object.html) key)

Returns true if this map contains a mapping for the specified key.

**Specified by:**[containsKey](http://docs.google.com/java/util/Map.html#containsKey(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[containsKey](http://docs.google.com/java/util/AbstractMap.html#containsKey(java.lang.Object)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Parameters:**key - The key whose presence in this map is to be tested **Returns:**true if this map contains a mapping for the specified key.

### put

public [V](http://docs.google.com/java/util/HashMap.html) **put**([K](http://docs.google.com/java/util/HashMap.html) key,  
 [V](http://docs.google.com/java/util/HashMap.html) value)

Associates the specified value with the specified key in this map. If the map previously contained a mapping for the key, the old value is replaced.

**Specified by:**[put](http://docs.google.com/java/util/Map.html#put(K,%20V)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[put](http://docs.google.com/java/util/AbstractMap.html#put(K,%20V)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Parameters:**key - key with which the specified value is to be associatedvalue - value to be associated with the specified key **Returns:**the previous value associated with key, or null if there was no mapping for key. (A null return can also indicate that the map previously associated null with key.)

### putAll

public void **putAll**([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/HashMap.html),? extends [V](http://docs.google.com/java/util/HashMap.html)> m)

Copies all of the mappings from the specified map to this map. These mappings will replace any mappings that this map had for any of the keys currently in the specified map.

**Specified by:**[putAll](http://docs.google.com/java/util/Map.html#putAll(java.util.Map)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[putAll](http://docs.google.com/java/util/AbstractMap.html#putAll(java.util.Map)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Parameters:**m - mappings to be stored in this map **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified map is null

### remove

public [V](http://docs.google.com/java/util/HashMap.html) **remove**([Object](http://docs.google.com/java/lang/Object.html) key)

Removes the mapping for the specified key from this map if present.

**Specified by:**[remove](http://docs.google.com/java/util/Map.html#remove(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[remove](http://docs.google.com/java/util/AbstractMap.html#remove(java.lang.Object)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Parameters:**key - key whose mapping is to be removed from the map **Returns:**the previous value associated with key, or null if there was no mapping for key. (A null return can also indicate that the map previously associated null with key.)

### clear

public void **clear**()

Removes all of the mappings from this map. The map will be empty after this call returns.

**Specified by:**[clear](http://docs.google.com/java/util/Map.html#clear()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[clear](http://docs.google.com/java/util/AbstractMap.html#clear()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>

### containsValue

public boolean **containsValue**([Object](http://docs.google.com/java/lang/Object.html) value)

Returns true if this map maps one or more keys to the specified value.

**Specified by:**[containsValue](http://docs.google.com/java/util/Map.html#containsValue(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[containsValue](http://docs.google.com/java/util/AbstractMap.html#containsValue(java.lang.Object)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Parameters:**value - value whose presence in this map is to be tested **Returns:**true if this map maps one or more keys to the specified value

### clone

public [Object](http://docs.google.com/java/lang/Object.html) **clone**()

Returns a shallow copy of this HashMap instance: the keys and values themselves are not cloned.

**Overrides:**[clone](http://docs.google.com/java/util/AbstractMap.html#clone()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Returns:**a shallow copy of this map**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### keySet

public [Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/HashMap.html)> **keySet**()

Returns a [Set](http://docs.google.com/java/util/Set.html) view of the keys contained in this map. The set is backed by the map, so changes to the map are reflected in the set, and vice-versa. If the map is modified while an iteration over the set is in progress (except through the iterator's own remove operation), the results of the iteration are undefined. The set supports element removal, which removes the corresponding mapping from the map, via the Iterator.remove, Set.remove, removeAll, retainAll, and clear operations. It does not support the add or addAll operations.

**Specified by:**[keySet](http://docs.google.com/java/util/Map.html#keySet()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[keySet](http://docs.google.com/java/util/AbstractMap.html#keySet()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Returns:**a set view of the keys contained in this map

### values

public [Collection](http://docs.google.com/java/util/Collection.html)<[V](http://docs.google.com/java/util/HashMap.html)> **values**()

Returns a [Collection](http://docs.google.com/java/util/Collection.html) view of the values contained in this map. The collection is backed by the map, so changes to the map are reflected in the collection, and vice-versa. If the map is modified while an iteration over the collection is in progress (except through the iterator's own remove operation), the results of the iteration are undefined. The collection supports element removal, which removes the corresponding mapping from the map, via the Iterator.remove, Collection.remove, removeAll, retainAll and clear operations. It does not support the add or addAll operations.

**Specified by:**[values](http://docs.google.com/java/util/Map.html#values()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Overrides:**[values](http://docs.google.com/java/util/AbstractMap.html#values()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Returns:**a collection view of the values contained in this map

### entrySet

public [Set](http://docs.google.com/java/util/Set.html)<[Map.Entry](http://docs.google.com/java/util/Map.Entry.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>> **entrySet**()

Returns a [Set](http://docs.google.com/java/util/Set.html) view of the mappings contained in this map. The set is backed by the map, so changes to the map are reflected in the set, and vice-versa. If the map is modified while an iteration over the set is in progress (except through the iterator's own remove operation, or through the setValue operation on a map entry returned by the iterator) the results of the iteration are undefined. The set supports element removal, which removes the corresponding mapping from the map, via the Iterator.remove, Set.remove, removeAll, retainAll and clear operations. It does not support the add or addAll operations.

**Specified by:**[entrySet](http://docs.google.com/java/util/Map.html#entrySet()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)>**Specified by:**[entrySet](http://docs.google.com/java/util/AbstractMap.html#entrySet()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/HashMap.html),[V](http://docs.google.com/java/util/HashMap.html)> **Returns:**a set view of the mappings contained in this map

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/HashMap.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/GregorianCalendar.html)   [**NEXT CLASS**](http://docs.google.com/java/util/HashSet.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/HashMap.html)    [**NO FRAMES**](http://docs.google.com/HashMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#1ksv4uv) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).