| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ConcurrentHashMap.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/concurrent/CompletionService.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/ConcurrentLinkedQueue.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/ConcurrentHashMap.html)    [**NO FRAMES**](http://docs.google.com/ConcurrentHashMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#2jxsxqh) |

## **java.util.concurrent**

Class ConcurrentHashMap<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.concurrent.ConcurrentHashMap<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), [ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<K,V>, [Map](http://docs.google.com/java/util/Map.html)<K,V>

public class **ConcurrentHashMap<K,V>**extends [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<K,V>implements [ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<K,V>, [Serializable](http://docs.google.com/java/io/Serializable.html)

A hash table supporting full concurrency of retrievals and adjustable expected concurrency for updates. This class obeys the same functional specification as [Hashtable](http://docs.google.com/java/util/Hashtable.html), and includes versions of methods corresponding to each method of Hashtable. However, even though all operations are thread-safe, retrieval operations do *not* entail locking, and there is *not* any support for locking the entire table in a way that prevents all access. This class is fully interoperable with Hashtable in programs that rely on its thread safety but not on its synchronization details.

Retrieval operations (including get) generally do not block, so may overlap with update operations (including put and remove). Retrievals reflect the results of the most recently *completed* update operations holding upon their onset. For aggregate operations such as putAll and clear, concurrent retrievals may reflect insertion or removal of only some entries. Similarly, Iterators and Enumerations return elements reflecting the state of the hash table at some point at or since the creation of the iterator/enumeration. They do *not* throw [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html). However, iterators are designed to be used by only one thread at a time.

The allowed concurrency among update operations is guided by the optional concurrencyLevel constructor argument (default 16), which is used as a hint for internal sizing. The table is internally partitioned to try to permit the indicated number of concurrent updates without contention. Because placement in hash tables is essentially random, the actual concurrency will vary. Ideally, you should choose a value to accommodate as many threads as will ever concurrently modify the table. Using a significantly higher value than you need can waste space and time, and a significantly lower value can lead to thread contention. But overestimates and underestimates within an order of magnitude do not usually have much noticeable impact. A value of one is appropriate when it is known that only one thread will modify and all others will only read. Also, resizing this or any other kind of hash table is a relatively slow operation, so, when possible, it is a good idea to provide estimates of expected table sizes in constructors.

This class and its views and iterators implement all of the *optional* methods of the [Map](http://docs.google.com/java/util/Map.html) and [Iterator](http://docs.google.com/java/util/Iterator.html) interfaces.

Like [Hashtable](http://docs.google.com/java/util/Hashtable.html) but unlike [HashMap](http://docs.google.com/java/util/HashMap.html), this class does *not* allow null to be used as a key or value.

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

**Since:** 1.5 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.util.concurrent.ConcurrentHashMap)

| **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)> |

| **Nested classes/interfaces inherited from interface java.util.**[**Map**](http://docs.google.com/java/util/Map.html) |
| --- |
| [Map.Entry](http://docs.google.com/java/util/Map.Entry.html)<[K](http://docs.google.com/java/util/Map.Entry.html),[V](http://docs.google.com/java/util/Map.Entry.html)> |

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

| **Method Summary** | |
| --- | --- |
| void | [**clear**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#clear())()            Removes all of the mappings from this map. |
| boolean | [**contains**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#contains(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) value)            Legacy method testing if some key maps into the specified value in this table. |
| boolean | [**containsKey**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#containsKey(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Tests if the specified object is a key in this table. |
| boolean | [**containsValue**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> | [**elements**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#elements())()            Returns an enumeration of the values in this table. |
| [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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)>> | [**entrySet**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html) | [**get**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html#isEmpty())()            Returns true if this map contains no key-value mappings. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> | [**keys**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#keys())()            Returns an enumeration of the keys in this table. |
| [Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> | [**keySet**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html) | [**put**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#put(K,%20V))([K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) key, [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) value)            Maps the specified key to the specified value in this table. |
| void | [**putAll**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#putAll(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html),? extends [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> m)            Copies all of the mappings from the specified map to this one. |
| [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) | [**putIfAbsent**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#putIfAbsent(K,%20V))([K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) key, [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) value)            If the specified key is not already associated with a value, associate it with the given value. |
| [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) | [**remove**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Removes the key (and its corresponding value) from this map. |
| boolean | [**remove**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#remove(java.lang.Object,%20java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key, [Object](http://docs.google.com/java/lang/Object.html) value)            Removes the entry for a key only if currently mapped to a given value. |
| [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) | [**replace**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#replace(K,%20V))([K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) key, [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) value)            Replaces the entry for a key only if currently mapped to some value. |
| boolean | [**replace**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#replace(K,%20V,%20V))([K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) key, [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) oldValue, [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) newValue)            Replaces the entry for a key only if currently mapped to a given value. |
| int | [**size**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html)> | [**values**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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) |
| --- |
| [clone](http://docs.google.com/java/util/AbstractMap.html#clone()), [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** |
| --- |

### ConcurrentHashMap

public **ConcurrentHashMap**(int initialCapacity,  
 float loadFactor,  
 int concurrencyLevel)

Creates a new, empty map with the specified initial capacity, load factor and concurrency level.

**Parameters:**initialCapacity - the initial capacity. The implementation performs internal sizing to accommodate this many elements.loadFactor - the load factor threshold, used to control resizing. Resizing may be performed when the average number of elements per bin exceeds this threshold.concurrencyLevel - the estimated number of concurrently updating threads. The implementation performs internal sizing to try to accommodate this many threads. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity is negative or the load factor or concurrencyLevel are nonpositive.

### ConcurrentHashMap

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

Creates a new, empty map with the specified initial capacity and load factor and with the default concurrencyLevel (16).

**Parameters:**initialCapacity - The implementation performs internal sizing to accommodate this many elements.loadFactor - the load factor threshold, used to control resizing. Resizing may be performed when the average number of elements per bin exceeds this threshold. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity of elements is negative or the load factor is nonpositive**Since:** 1.6

### ConcurrentHashMap

public **ConcurrentHashMap**(int initialCapacity)

Creates a new, empty map with the specified initial capacity, and with default load factor (0.75) and concurrencyLevel (16).

**Parameters:**initialCapacity - the initial capacity. The implementation performs internal sizing to accommodate this many elements. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity of elements is negative.

### ConcurrentHashMap

public **ConcurrentHashMap**()

Creates a new, empty map with a default initial capacity (16), load factor (0.75) and concurrencyLevel (16).

### ConcurrentHashMap

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

Creates a new map with the same mappings as the given map. The map is created with a capacity of 1.5 times the number of mappings in the given map or 16 (whichever is greater), and a default load factor (0.75) and concurrencyLevel (16).

**Parameters:**m - the map

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

### 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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **Returns:**true if this map contains no key-value mappings

### size

public int **size**()

Returns the number of key-value mappings in this map. If the map contains more than Integer.MAX\_VALUE elements, returns Integer.MAX\_VALUE.

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **Returns:**the number of key-value mappings in this map

### get

public [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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.equals(k), then this method returns v; otherwise it returns null. (There can be at most one such mapping.)

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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 **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key is null

### containsKey

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

Tests if the specified object is a key in this table.

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **Parameters:**key - possible key **Returns:**true if and only if the specified object is a key in this table, as determined by the equals method; false otherwise. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key is null

### 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. Note: This method requires a full internal traversal of the hash table, and so is much slower than method containsKey.

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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 **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified value is null

### contains

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

Legacy method testing if some key maps into the specified value in this table. This method is identical in functionality to [containsValue(java.lang.Object)](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#containsValue(java.lang.Object)), and exists solely to ensure full compatibility with class [Hashtable](http://docs.google.com/java/util/Hashtable.html), which supported this method prior to introduction of the Java Collections framework.

**Parameters:**value - a value to search for **Returns:**true if and only if some key maps to the value argument in this table as determined by the equals method; false otherwise **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified value is null

### put

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

Maps the specified key to the specified value in this table. Neither the key nor the value can be null.

The value can be retrieved by calling the get method with a key that is equal to the original key.

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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 **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key or value is null

### putIfAbsent

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

If the specified key is not already associated with a value, associate it with the given value. This is equivalent to

if (!map.containsKey(key))  
 return map.put(key, value);  
 else  
 return map.get(key);

except that the action is performed atomically.

**Specified by:**[putIfAbsent](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#putIfAbsent(K,%20V)) in interface [ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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 the specified key, or null if there was no mapping for the key **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key or value is null

### putAll

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

Copies all of the mappings from the specified map to this one. These mappings 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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **Parameters:**m - mappings to be stored in this map

### remove

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

Removes the key (and its corresponding value) from this map. This method does nothing if the key is not in the map.

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **Parameters:**key - the key that needs to be removed **Returns:**the previous value associated with key, or null if there was no mapping for key **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key is null

### remove

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

Removes the entry for a key only if currently mapped to a given value. This is equivalent to

if (map.containsKey(key) && map.get(key).equals(value)) {  
 map.remove(key);  
 return true;  
 } else return false;

except that the action is performed atomically.

**Specified by:**[remove](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#remove(java.lang.Object,%20java.lang.Object)) in interface [ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **Parameters:**key - key with which the specified value is associatedvalue - value expected to be associated with the specified key **Returns:**true if the value was removed **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key is null

### replace

public boolean **replace**([K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) key,  
 [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) oldValue,  
 [V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html) newValue)

Replaces the entry for a key only if currently mapped to a given value. This is equivalent to

if (map.containsKey(key) && map.get(key).equals(oldValue)) {  
 map.put(key, newValue);  
 return true;  
 } else return false;

except that the action is performed atomically.

**Specified by:**[replace](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#replace(K,%20V,%20V)) in interface [ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **Parameters:**key - key with which the specified value is associatedoldValue - value expected to be associated with the specified keynewValue - value to be associated with the specified key **Returns:**true if the value was replaced **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if any of the arguments are null

### replace

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

Replaces the entry for a key only if currently mapped to some value. This is equivalent to

if (map.containsKey(key)) {  
 return map.put(key, value);  
 } else return null;

except that the action is performed atomically.

**Specified by:**[replace](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#replace(K,%20V)) in interface [ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **Parameters:**key - key with which the specified value is associatedvalue - value to be associated with the specified key **Returns:**the previous value associated with the specified key, or null if there was no mapping for the key **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key or value is null

### clear

public void **clear**()

Removes all of the mappings from this map.

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)>

### keySet

public [Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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. The set supports element removal, which removes the corresponding mapping from this map, via the Iterator.remove, Set.remove, removeAll, retainAll, and clear operations. It does not support the add or addAll operations.

The view's iterator is a "weakly consistent" iterator that will never throw [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html), and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.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. The collection supports element removal, which removes the corresponding mapping from this map, via the Iterator.remove, Collection.remove, removeAll, retainAll, and clear operations. It does not support the add or addAll operations.

The view's iterator is a "weakly consistent" iterator that will never throw [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html), and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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. 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.

The view's iterator is a "weakly consistent" iterator that will never throw [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html), and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

**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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.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/concurrent/ConcurrentHashMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **Returns:**a set view of the mappings contained in this map

### keys

public [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **keys**()

Returns an enumeration of the keys in this table.

**Returns:**an enumeration of the keys in this table**See Also:**[keySet()](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#keySet())

### elements

public [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[V](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)> **elements**()

Returns an enumeration of the values in this table.

**Returns:**an enumeration of the values in this table**See Also:**[values()](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html#values())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ConcurrentHashMap.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/concurrent/CompletionService.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/ConcurrentLinkedQueue.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/ConcurrentHashMap.html)    [**NO FRAMES**](http://docs.google.com/ConcurrentHashMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#2jxsxqh) |

[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).