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

## **java.util.concurrent**

Interface ConcurrentMap<K,V>

**Type Parameters:**K - the type of keys maintained by this mapV - the type of mapped values **All Superinterfaces:** [Map](http://docs.google.com/java/util/Map.html)<K,V> **All Known Subinterfaces:** [ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<K,V> **All Known Implementing Classes:** [ConcurrentHashMap](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html), [ConcurrentSkipListMap](http://docs.google.com/java/util/concurrent/ConcurrentSkipListMap.html)

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

A [Map](http://docs.google.com/java/util/Map.html) providing additional atomic putIfAbsent, remove, and replace methods.

Memory consistency effects: As with other concurrent collections, actions in a thread prior to placing an object into a ConcurrentMap as a key or value [*happen-before*](http://docs.google.com/package-summary.html#MemoryVisibility) actions subsequent to the access or removal of that object from the ConcurrentMap in another thread.

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

**Since:** 1.5

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

| **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)> |

| **Method Summary** | |
| --- | --- |
| [V](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) | [**putIfAbsent**](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#putIfAbsent(K,%20V))([K](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) key, [V](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) value)            If the specified key is not already associated with a value, associate it with the given value. |
| boolean | [**remove**](http://docs.google.com/java/util/concurrent/ConcurrentMap.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/ConcurrentMap.html) | [**replace**](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#replace(K,%20V))([K](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) key, [V](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) value)            Replaces the entry for a key only if currently mapped to some value. |
| boolean | [**replace**](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#replace(K,%20V,%20V))([K](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) key, [V](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) oldValue, [V](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) newValue)            Replaces the entry for a key only if currently mapped to a given value. |

| **Methods inherited from interface java.util.**[**Map**](http://docs.google.com/java/util/Map.html) |
| --- |
| [clear](http://docs.google.com/java/util/Map.html#clear()), [containsKey](http://docs.google.com/java/util/Map.html#containsKey(java.lang.Object)), [containsValue](http://docs.google.com/java/util/Map.html#containsValue(java.lang.Object)), [entrySet](http://docs.google.com/java/util/Map.html#entrySet()), [equals](http://docs.google.com/java/util/Map.html#equals(java.lang.Object)), [get](http://docs.google.com/java/util/Map.html#get(java.lang.Object)), [hashCode](http://docs.google.com/java/util/Map.html#hashCode()), [isEmpty](http://docs.google.com/java/util/Map.html#isEmpty()), [keySet](http://docs.google.com/java/util/Map.html#keySet()), [put](http://docs.google.com/java/util/Map.html#put(K,%20V)), [putAll](http://docs.google.com/java/util/Map.html#putAll(java.util.Map)), [remove](http://docs.google.com/java/util/Map.html#remove(java.lang.Object)), [size](http://docs.google.com/java/util/Map.html#size()), [values](http://docs.google.com/java/util/Map.html#values()) |

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

### putIfAbsent

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

**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. (A null return can also indicate that the map previously associated null with the key, if the implementation supports null values.) **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the put operation is not supported by this map [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of the specified key or value prevents it from being stored in this map [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key or value is null, and this map does not permit null keys or values [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if some property of the specified key or value prevents it from being stored in this map

### remove

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.

**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:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the remove operation is not supported by this map [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the key or value is of an inappropriate type for this map (optional) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key or value is null, and this map does not permit null keys or values (optional)

### replace

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

**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:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the put operation is not supported by this map [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of a specified key or value prevents it from being stored in this map [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if a specified key or value is null, and this map does not permit null keys or values [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if some property of a specified key or value prevents it from being stored in this map

### replace

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

**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. (A null return can also indicate that the map previously associated null with the key, if the implementation supports null values.) **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the put operation is not supported by this map [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of the specified key or value prevents it from being stored in this map [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key or value is null, and this map does not permit null keys or values [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if some property of the specified key or value prevents it from being stored 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/ConcurrentMap.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/ConcurrentLinkedQueue.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/ConcurrentMap.html)    [**NO FRAMES**](http://docs.google.com/ConcurrentMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#1t3h5sf) |

[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).