| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Hashtable.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/HashSet.html)   [**NEXT CLASS**](http://docs.google.com/java/util/IdentityHashMap.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/Hashtable.html)    [**NO FRAMES**](http://docs.google.com/Hashtable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

## **java.util**

Class Hashtable<K,V>

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

**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:** [Properties](http://docs.google.com/java/util/Properties.html), [UIDefaults](http://docs.google.com/javax/swing/UIDefaults.html)

public class **Hashtable<K,V>**extends [Dictionary](http://docs.google.com/java/util/Dictionary.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)

This class implements a hashtable, which maps keys to values. Any non-null object can be used as a key or as a value.

To successfully store and retrieve objects from a hashtable, the objects used as keys must implement the hashCode method and the equals method.

An instance of Hashtable 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. Note that the hash table is *open*: in the case of a "hash collision", a single bucket stores multiple entries, which must be searched sequentially. The *load factor* is a measure of how full the hash table is allowed to get before its capacity is automatically increased. The initial capacity and load factor parameters are merely hints to the implementation. The exact details as to when and whether the rehash method is invoked are implementation-dependent.

Generally, the default load factor (.75) offers a good tradeoff between time and space costs. Higher values decrease the space overhead but increase the time cost to look up an entry (which is reflected in most Hashtable operations, including get and put).

The initial capacity controls a tradeoff between wasted space and the need for rehash operations, which are time-consuming. No rehash operations will *ever* occur if the initial capacity is greater than the maximum number of entries the Hashtable will contain divided by its load factor. However, setting the initial capacity too high can waste space.

If many entries are to be made into a Hashtable, creating it with a sufficiently large capacity may allow the entries to be inserted more efficiently than letting it perform automatic rehashing as needed to grow the table.

This example creates a hashtable of numbers. It uses the names of the numbers as keys:

Hashtable<String, Integer> numbers  
 = new Hashtable<String, Integer>();  
 numbers.put("one", 1);  
 numbers.put("two", 2);  
 numbers.put("three", 3);

To retrieve a number, use the following code:

Integer n = numbers.get("two");  
 if (n != null) {  
 System.out.println("two = " + n);  
 }

The iterators returned by the iterator method of the collections returned by all of this class's "collection view methods" are *fail-fast*: if the Hashtable 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. The Enumerations returned by Hashtable's keys and elements methods are *not* fail-fast.

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.*

As of the Java 2 platform v1.2, this class was retrofitted to implement the [Map](http://docs.google.com/java/util/Map.html) interface, making it a member of the  [Java Collections Framework](http://docs.google.com/technotes/guides/collections/index.html). Unlike the new collection implementations, Hashtable is synchronized.

**Since:** JDK1.0 **See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [rehash()](http://docs.google.com/java/util/Hashtable.html#rehash()), [Collection](http://docs.google.com/java/util/Collection.html), [Map](http://docs.google.com/java/util/Map.html), [HashMap](http://docs.google.com/java/util/HashMap.html), [TreeMap](http://docs.google.com/java/util/TreeMap.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.util.Hashtable)

| **Constructor Summary** | |
| --- | --- |
| [**Hashtable**](http://docs.google.com/java/util/Hashtable.html#Hashtable())()            Constructs a new, empty hashtable with a default initial capacity (11) and load factor (0.75). |
| [**Hashtable**](http://docs.google.com/java/util/Hashtable.html#Hashtable(int))(int initialCapacity)            Constructs a new, empty hashtable with the specified initial capacity and default load factor (0.75). |
| [**Hashtable**](http://docs.google.com/java/util/Hashtable.html#Hashtable(int,%20float))(int initialCapacity, float loadFactor)            Constructs a new, empty hashtable with the specified initial capacity and the specified load factor. |
| [**Hashtable**](http://docs.google.com/java/util/Hashtable.html#Hashtable(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/Hashtable.html),? extends [V](http://docs.google.com/java/util/Hashtable.html)> t)            Constructs a new hashtable with the same mappings as the given Map. |

| **Method Summary** | |
| --- | --- |
| void | [**clear**](http://docs.google.com/java/util/Hashtable.html#clear())()            Clears this hashtable so that it contains no keys. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/util/Hashtable.html#clone())()            Creates a shallow copy of this hashtable. |
| boolean | [**contains**](http://docs.google.com/java/util/Hashtable.html#contains(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) value)            Tests if some key maps into the specified value in this hashtable. |
| boolean | [**containsKey**](http://docs.google.com/java/util/Hashtable.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 hashtable. |
| boolean | [**containsValue**](http://docs.google.com/java/util/Hashtable.html#containsValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) value)            Returns true if this hashtable maps one or more keys to this value. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[V](http://docs.google.com/java/util/Hashtable.html)> | [**elements**](http://docs.google.com/java/util/Hashtable.html#elements())()            Returns an enumeration of the values in this hashtable. |
| [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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)>> | [**entrySet**](http://docs.google.com/java/util/Hashtable.html#entrySet())()            Returns a [Set](http://docs.google.com/java/util/Set.html) view of the mappings contained in this map. |
| boolean | [**equals**](http://docs.google.com/java/util/Hashtable.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Compares the specified Object with this Map for equality, as per the definition in the Map interface. |
| [V](http://docs.google.com/java/util/Hashtable.html) | [**get**](http://docs.google.com/java/util/Hashtable.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. |
| int | [**hashCode**](http://docs.google.com/java/util/Hashtable.html#hashCode())()            Returns the hash code value for this Map as per the definition in the Map interface. |
| boolean | [**isEmpty**](http://docs.google.com/java/util/Hashtable.html#isEmpty())()            Tests if this hashtable maps no keys to values. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[K](http://docs.google.com/java/util/Hashtable.html)> | [**keys**](http://docs.google.com/java/util/Hashtable.html#keys())()            Returns an enumeration of the keys in this hashtable. |
| [Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/Hashtable.html)> | [**keySet**](http://docs.google.com/java/util/Hashtable.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/Hashtable.html) | [**put**](http://docs.google.com/java/util/Hashtable.html#put(K,%20V))([K](http://docs.google.com/java/util/Hashtable.html) key, [V](http://docs.google.com/java/util/Hashtable.html) value)            Maps the specified key to the specified value in this hashtable. |
| void | [**putAll**](http://docs.google.com/java/util/Hashtable.html#putAll(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/Hashtable.html),? extends [V](http://docs.google.com/java/util/Hashtable.html)> t)            Copies all of the mappings from the specified map to this hashtable. |
| protected  void | [**rehash**](http://docs.google.com/java/util/Hashtable.html#rehash())()            Increases the capacity of and internally reorganizes this hashtable, in order to accommodate and access its entries more efficiently. |
| [V](http://docs.google.com/java/util/Hashtable.html) | [**remove**](http://docs.google.com/java/util/Hashtable.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Removes the key (and its corresponding value) from this hashtable. |
| int | [**size**](http://docs.google.com/java/util/Hashtable.html#size())()            Returns the number of keys in this hashtable. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/util/Hashtable.html#toString())()            Returns a string representation of this Hashtable object in the form of a set of entries, enclosed in braces and separated by the ASCII characters ", " (comma and space). |
| [Collection](http://docs.google.com/java/util/Collection.html)<[V](http://docs.google.com/java/util/Hashtable.html)> | [**values**](http://docs.google.com/java/util/Hashtable.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.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)) |

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

### Hashtable

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

Constructs a new, empty hashtable with the specified initial capacity and the specified load factor.

**Parameters:**initialCapacity - the initial capacity of the hashtable.loadFactor - the load factor of the hashtable. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity is less than zero, or if the load factor is nonpositive.

### Hashtable

public **Hashtable**(int initialCapacity)

Constructs a new, empty hashtable with the specified initial capacity and default load factor (0.75).

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

### Hashtable

public **Hashtable**()

Constructs a new, empty hashtable with a default initial capacity (11) and load factor (0.75).

### Hashtable

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

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

**Parameters:**t - 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.**Since:** 1.2

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

### size

public int **size**()

Returns the number of keys in this hashtable.

**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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)>**Specified by:**[size](http://docs.google.com/java/util/Dictionary.html#size()) in class [Dictionary](http://docs.google.com/java/util/Dictionary.html)<[K](http://docs.google.com/java/util/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Returns:**the number of keys in this hashtable.

### isEmpty

public boolean **isEmpty**()

Tests if this hashtable maps no keys to values.

**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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)>**Specified by:**[isEmpty](http://docs.google.com/java/util/Dictionary.html#isEmpty()) in class [Dictionary](http://docs.google.com/java/util/Dictionary.html)<[K](http://docs.google.com/java/util/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Returns:**true if this hashtable maps no keys to values; false otherwise.

### keys

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

Returns an enumeration of the keys in this hashtable.

**Specified by:**[keys](http://docs.google.com/java/util/Dictionary.html#keys()) in class [Dictionary](http://docs.google.com/java/util/Dictionary.html)<[K](http://docs.google.com/java/util/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Returns:**an enumeration of the keys in this hashtable.**See Also:**[Enumeration](http://docs.google.com/java/util/Enumeration.html), [elements()](http://docs.google.com/java/util/Hashtable.html#elements()), [keySet()](http://docs.google.com/java/util/Hashtable.html#keySet()), [Map](http://docs.google.com/java/util/Map.html)

### elements

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

Returns an enumeration of the values in this hashtable. Use the Enumeration methods on the returned object to fetch the elements sequentially.

**Specified by:**[elements](http://docs.google.com/java/util/Dictionary.html#elements()) in class [Dictionary](http://docs.google.com/java/util/Dictionary.html)<[K](http://docs.google.com/java/util/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Returns:**an enumeration of the values in this hashtable.**See Also:**[Enumeration](http://docs.google.com/java/util/Enumeration.html), [keys()](http://docs.google.com/java/util/Hashtable.html#keys()), [values()](http://docs.google.com/java/util/Hashtable.html#values()), [Map](http://docs.google.com/java/util/Map.html)

### contains

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

Tests if some key maps into the specified value in this hashtable. This operation is more expensive than the [containsKey](http://docs.google.com/java/util/Hashtable.html#containsKey(java.lang.Object)) method.

Note that this method is identical in functionality to [containsValue](http://docs.google.com/java/util/Hashtable.html#containsValue(java.lang.Object)), (which is part of the [Map](http://docs.google.com/java/util/Map.html) interface in the collections framework).

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

### containsValue

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

Returns true if this hashtable maps one or more keys to this value.

Note that this method is identical in functionality to [contains](http://docs.google.com/java/util/Hashtable.html#contains(java.lang.Object)) (which predates the [Map](http://docs.google.com/java/util/Map.html) interface).

**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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Parameters:**value - value whose presence in this hashtable 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 value is null**Since:** 1.2

### containsKey

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

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

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

### get

public [V](http://docs.google.com/java/util/Hashtable.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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)>**Specified by:**[get](http://docs.google.com/java/util/Dictionary.html#get(java.lang.Object)) in class [Dictionary](http://docs.google.com/java/util/Dictionary.html)<[K](http://docs.google.com/java/util/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.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**See Also:**[put(Object, Object)](http://docs.google.com/java/util/Hashtable.html#put(K,%20V))

### rehash

protected void **rehash**()

Increases the capacity of and internally reorganizes this hashtable, in order to accommodate and access its entries more efficiently. This method is called automatically when the number of keys in the hashtable exceeds this hashtable's capacity and load factor.

### put

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

Maps the specified key to the specified value in this hashtable. 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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)>**Specified by:**[put](http://docs.google.com/java/util/Dictionary.html#put(K,%20V)) in class [Dictionary](http://docs.google.com/java/util/Dictionary.html)<[K](http://docs.google.com/java/util/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Parameters:**key - the hashtable keyvalue - the value **Returns:**the previous value of the specified key in this hashtable, or null if it did not have one **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the key or value is null**See Also:**[Object.equals(Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [get(Object)](http://docs.google.com/java/util/Hashtable.html#get(java.lang.Object))

### remove

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

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

**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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)>**Specified by:**[remove](http://docs.google.com/java/util/Dictionary.html#remove(java.lang.Object)) in class [Dictionary](http://docs.google.com/java/util/Dictionary.html)<[K](http://docs.google.com/java/util/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Parameters:**key - the key that needs to be removed **Returns:**the value to which the key had been mapped in this hashtable, or null if the key did not have a mapping **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the key is null

### putAll

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

Copies all of the mappings from the specified map to this hashtable. These mappings will replace any mappings that this hashtable 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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Parameters:**t - mappings to be stored in this map **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified map is null**Since:** 1.2

### clear

public void **clear**()

Clears this hashtable so that it contains no keys.

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

### clone

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

Creates a shallow copy of this hashtable. All the structure of the hashtable itself is copied, but the keys and values are not cloned. This is a relatively expensive operation.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone of the hashtable**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### toString

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

Returns a string representation of this Hashtable object in the form of a set of entries, enclosed in braces and separated by the ASCII characters ", " (comma and space). Each entry is rendered as the key, an equals sign =, and the associated element, where the toString method is used to convert the key and element to strings.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of this hashtable

### keySet

public [Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/Hashtable.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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Returns:**a set view of the keys contained in this map**Since:** 1.2

### 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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Returns:**a set view of the mappings contained in this map**Since:** 1.2

### values

public [Collection](http://docs.google.com/java/util/Collection.html)<[V](http://docs.google.com/java/util/Hashtable.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/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)> **Returns:**a collection view of the values contained in this map**Since:** 1.2

### equals

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

Compares the specified Object with this Map for equality, as per the definition in the Map interface.

**Specified by:**[equals](http://docs.google.com/java/util/Map.html#equals(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)>**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**o - object to be compared for equality with this hashtable **Returns:**true if the specified Object is equal to this Map**Since:** 1.2 **See Also:**[Map.equals(Object)](http://docs.google.com/java/util/Map.html#equals(java.lang.Object))

### hashCode

public int **hashCode**()

Returns the hash code value for this Map as per the definition in the Map interface.

**Specified by:**[hashCode](http://docs.google.com/java/util/Map.html#hashCode()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/Hashtable.html),[V](http://docs.google.com/java/util/Hashtable.html)>**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**Since:** 1.2 **See Also:**[Map.hashCode()](http://docs.google.com/java/util/Map.html#hashCode())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Hashtable.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/HashSet.html)   [**NEXT CLASS**](http://docs.google.com/java/util/IdentityHashMap.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/Hashtable.html)    [**NO FRAMES**](http://docs.google.com/Hashtable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

[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).