| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CopyOnWriteArraySet.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/CopyOnWriteArrayList.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/CountDownLatch.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/CopyOnWriteArraySet.html)    [**NO FRAMES**](http://docs.google.com/CopyOnWriteArraySet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#3rdcrjn) |

## **java.util.concurrent**

Class CopyOnWriteArraySet<E>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.util.AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<E>  
 [java.util.AbstractSet](http://docs.google.com/java/util/AbstractSet.html)<E>  
 **java.util.concurrent.CopyOnWriteArraySet<E>**

**Type Parameters:**E - the type of elements held in this collection **All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Iterable](http://docs.google.com/java/lang/Iterable.html)<E>, [Collection](http://docs.google.com/java/util/Collection.html)<E>, [Set](http://docs.google.com/java/util/Set.html)<E>

public class **CopyOnWriteArraySet<E>**extends [AbstractSet](http://docs.google.com/java/util/AbstractSet.html)<E>implements [Serializable](http://docs.google.com/java/io/Serializable.html)

A [Set](http://docs.google.com/java/util/Set.html) that uses an internal [CopyOnWriteArrayList](http://docs.google.com/java/util/concurrent/CopyOnWriteArrayList.html) for all of its operations. Thus, it shares the same basic properties:

* It is best suited for applications in which set sizes generally stay small, read-only operations vastly outnumber mutative operations, and you need to prevent interference among threads during traversal.
* It is thread-safe.
* Mutative operations (add, set, remove, etc.) are expensive since they usually entail copying the entire underlying array.
* Iterators do not support the mutative remove operation.
* Traversal via iterators is fast and cannot encounter interference from other threads. Iterators rely on unchanging snapshots of the array at the time the iterators were constructed.

**Sample Usage.** The following code sketch uses a copy-on-write set to maintain a set of Handler objects that perform some action upon state updates.

class Handler { void handle(); ... }  
  
 class X {  
 private final CopyOnWriteArraySet<Handler> handlers  
 = new CopyOnWriteArraySet<Handler>();  
 public void addHandler(Handler h) { handlers.add(h); }  
  
 private long internalState;  
 private synchronized void changeState() { internalState = ...; }  
  
 public void update() {  
 changeState();  
 for (Handler handler : handlers)  
 handler.handle();  
 }  
 }

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

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

| **Constructor Summary** | |
| --- | --- |
| [**CopyOnWriteArraySet**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#CopyOnWriteArraySet())()            Creates an empty set. |
| [**CopyOnWriteArraySet**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#CopyOnWriteArraySet(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> c)            Creates a set containing all of the elements of the specified collection. |

| **Method Summary** | |
| --- | --- |
| boolean | [**add**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#add(E))([E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html) e)            Adds the specified element to this set if it is not already present. |
| boolean | [**addAll**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#addAll(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> c)            Adds all of the elements in the specified collection to this set if they're not already present. |
| void | [**clear**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#clear())()            Removes all of the elements from this set. |
| boolean | [**contains**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#contains(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Returns true if this set contains the specified element. |
| boolean | [**containsAll**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#containsAll(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<?> c)            Returns true if this set contains all of the elements of the specified collection. |
| boolean | [**equals**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Compares the specified object with this set for equality. |
| boolean | [**isEmpty**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#isEmpty())()            Returns true if this set contains no elements. |
| [Iterator](http://docs.google.com/java/util/Iterator.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> | [**iterator**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#iterator())()            Returns an iterator over the elements contained in this set in the order in which these elements were added. |
| boolean | [**remove**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Removes the specified element from this set if it is present. |
| boolean | [**removeAll**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#removeAll(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<?> c)            Removes from this set all of its elements that are contained in the specified collection. |
| boolean | [**retainAll**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#retainAll(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<?> c)            Retains only the elements in this set that are contained in the specified collection. |
| int | [**size**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#size())()            Returns the number of elements in this set. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**toArray**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#toArray())()            Returns an array containing all of the elements in this set. |
| | <T> T[] | | --- | | [**toArray**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#toArray(T%5B%5D))(T[] a)            Returns an array containing all of the elements in this set; the runtime type of the returned array is that of the specified array. |

| **Methods inherited from class java.util.**[**AbstractSet**](http://docs.google.com/java/util/AbstractSet.html) |
| --- |
| [hashCode](http://docs.google.com/java/util/AbstractSet.html#hashCode()) |

| **Methods inherited from class java.util.**[**AbstractCollection**](http://docs.google.com/java/util/AbstractCollection.html) |
| --- |
| [toString](http://docs.google.com/java/util/AbstractCollection.html#toString()) |

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

### CopyOnWriteArraySet

public **CopyOnWriteArraySet**()

Creates an empty set.

### CopyOnWriteArraySet

public **CopyOnWriteArraySet**([Collection](http://docs.google.com/java/util/Collection.html)<? extends [E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> c)

Creates a set containing all of the elements of the specified collection.

**Parameters:**c - the collection of elements to initially contain **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified collection is null

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

### size

public int **size**()

Returns the number of elements in this set.

**Specified by:**[size](http://docs.google.com/java/util/Collection.html#size()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[size](http://docs.google.com/java/util/Set.html#size()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[size](http://docs.google.com/java/util/AbstractCollection.html#size()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Returns:**the number of elements in this set

### isEmpty

public boolean **isEmpty**()

Returns true if this set contains no elements.

**Specified by:**[isEmpty](http://docs.google.com/java/util/Collection.html#isEmpty()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[isEmpty](http://docs.google.com/java/util/Set.html#isEmpty()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[isEmpty](http://docs.google.com/java/util/AbstractCollection.html#isEmpty()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Returns:**true if this set contains no elements

### contains

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

Returns true if this set contains the specified element. More formally, returns true if and only if this set contains an element e such that (o==null ? e==null : o.equals(e)).

**Specified by:**[contains](http://docs.google.com/java/util/Collection.html#contains(java.lang.Object)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[contains](http://docs.google.com/java/util/Set.html#contains(java.lang.Object)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[contains](http://docs.google.com/java/util/AbstractCollection.html#contains(java.lang.Object)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Parameters:**o - element whose presence in this set is to be tested **Returns:**true if this set contains the specified element

### toArray

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

Returns an array containing all of the elements in this set. If this set makes any guarantees as to what order its elements are returned by its iterator, this method must return the elements in the same order.

The returned array will be "safe" in that no references to it are maintained by this set. (In other words, this method must allocate a new array even if this set is backed by an array). The caller is thus free to modify the returned array.

This method acts as bridge between array-based and collection-based APIs.

**Specified by:**[toArray](http://docs.google.com/java/util/Collection.html#toArray()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[toArray](http://docs.google.com/java/util/Set.html#toArray()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Returns:**an array containing all the elements in this set

### toArray

public <T> T[] **toArray**(T[] a)

Returns an array containing all of the elements in this set; the runtime type of the returned array is that of the specified array. If the set fits in the specified array, it is returned therein. Otherwise, a new array is allocated with the runtime type of the specified array and the size of this set.

If this set fits in the specified array with room to spare (i.e., the array has more elements than this set), the element in the array immediately following the end of the set is set to null. (This is useful in determining the length of this set *only* if the caller knows that this set does not contain any null elements.)

If this set makes any guarantees as to what order its elements are returned by its iterator, this method must return the elements in the same order.

Like the [toArray()](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#toArray()) method, this method acts as bridge between array-based and collection-based APIs. Further, this method allows precise control over the runtime type of the output array, and may, under certain circumstances, be used to save allocation costs.

Suppose x is a set known to contain only strings. The following code can be used to dump the set into a newly allocated array of String:

String[] y = x.toArray(new String[0]);

Note that toArray(new Object[0]) is identical in function to toArray().

**Specified by:**[toArray](http://docs.google.com/java/util/Collection.html#toArray(T%5B%5D)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[toArray](http://docs.google.com/java/util/Set.html#toArray(T%5B%5D)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray(T%5B%5D)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Parameters:**a - the array into which the elements of this set are to be stored, if it is big enough; otherwise, a new array of the same runtime type is allocated for this purpose. **Returns:**an array containing all the elements in this set **Throws:** [ArrayStoreException](http://docs.google.com/java/lang/ArrayStoreException.html) - if the runtime type of the specified array is not a supertype of the runtime type of every element in this set [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified array is null

### clear

public void **clear**()

Removes all of the elements from this set. The set will be empty after this call returns.

**Specified by:**[clear](http://docs.google.com/java/util/Collection.html#clear()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[clear](http://docs.google.com/java/util/Set.html#clear()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[clear](http://docs.google.com/java/util/AbstractCollection.html#clear()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>

### remove

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

Removes the specified element from this set if it is present. More formally, removes an element e such that (o==null ? e==null : o.equals(e)), if this set contains such an element. Returns true if this set contained the element (or equivalently, if this set changed as a result of the call). (This set will not contain the element once the call returns.)

**Specified by:**[remove](http://docs.google.com/java/util/Collection.html#remove(java.lang.Object)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[remove](http://docs.google.com/java/util/Set.html#remove(java.lang.Object)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[remove](http://docs.google.com/java/util/AbstractCollection.html#remove(java.lang.Object)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Parameters:**o - object to be removed from this set, if present **Returns:**true if this set contained the specified element

### add

public boolean **add**([E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html) e)

Adds the specified element to this set if it is not already present. More formally, adds the specified element e to this set if the set contains no element e2 such that (e==null ? e2==null : e.equals(e2)). If this set already contains the element, the call leaves the set unchanged and returns false.

**Specified by:**[add](http://docs.google.com/java/util/Collection.html#add(E)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[add](http://docs.google.com/java/util/Set.html#add(E)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[add](http://docs.google.com/java/util/AbstractCollection.html#add(E)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Parameters:**e - element to be added to this set **Returns:**true if this set did not already contain the specified element

### containsAll

public boolean **containsAll**([Collection](http://docs.google.com/java/util/Collection.html)<?> c)

Returns true if this set contains all of the elements of the specified collection. If the specified collection is also a set, this method returns true if it is a *subset* of this set.

**Specified by:**[containsAll](http://docs.google.com/java/util/Collection.html#containsAll(java.util.Collection)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[containsAll](http://docs.google.com/java/util/Set.html#containsAll(java.util.Collection)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[containsAll](http://docs.google.com/java/util/AbstractCollection.html#containsAll(java.util.Collection)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Parameters:**c - collection to be checked for containment in this set **Returns:**true if this set contains all of the elements of the specified collection **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified collection is null**See Also:**[contains(Object)](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#contains(java.lang.Object))

### addAll

public boolean **addAll**([Collection](http://docs.google.com/java/util/Collection.html)<? extends [E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> c)

Adds all of the elements in the specified collection to this set if they're not already present. If the specified collection is also a set, the addAll operation effectively modifies this set so that its value is the *union* of the two sets. The behavior of this operation is undefined if the specified collection is modified while the operation is in progress.

**Specified by:**[addAll](http://docs.google.com/java/util/Collection.html#addAll(java.util.Collection)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[addAll](http://docs.google.com/java/util/Set.html#addAll(java.util.Collection)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[addAll](http://docs.google.com/java/util/AbstractCollection.html#addAll(java.util.Collection)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Parameters:**c - collection containing elements to be added to this set **Returns:**true if this set changed as a result of the call **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified collection is null**See Also:**[add(Object)](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#add(E))

### removeAll

public boolean **removeAll**([Collection](http://docs.google.com/java/util/Collection.html)<?> c)

Removes from this set all of its elements that are contained in the specified collection. If the specified collection is also a set, this operation effectively modifies this set so that its value is the *asymmetric set difference* of the two sets.

**Specified by:**[removeAll](http://docs.google.com/java/util/Collection.html#removeAll(java.util.Collection)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[removeAll](http://docs.google.com/java/util/Set.html#removeAll(java.util.Collection)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[removeAll](http://docs.google.com/java/util/AbstractSet.html#removeAll(java.util.Collection)) in class [AbstractSet](http://docs.google.com/java/util/AbstractSet.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Parameters:**c - collection containing elements to be removed from this set **Returns:**true if this set changed as a result of the call **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of an element of this set is incompatible with the specified collection (optional) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if this set contains a null element and the specified collection does not permit null elements (optional), or if the specified collection is null**See Also:**[remove(Object)](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#remove(java.lang.Object))

### retainAll

public boolean **retainAll**([Collection](http://docs.google.com/java/util/Collection.html)<?> c)

Retains only the elements in this set that are contained in the specified collection. In other words, removes from this set all of its elements that are not contained in the specified collection. If the specified collection is also a set, this operation effectively modifies this set so that its value is the *intersection* of the two sets.

**Specified by:**[retainAll](http://docs.google.com/java/util/Collection.html#retainAll(java.util.Collection)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[retainAll](http://docs.google.com/java/util/Set.html#retainAll(java.util.Collection)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[retainAll](http://docs.google.com/java/util/AbstractCollection.html#retainAll(java.util.Collection)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Parameters:**c - collection containing elements to be retained in this set **Returns:**true if this set changed as a result of the call **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of an element of this set is incompatible with the specified collection (optional) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if this set contains a null element and the specified collection does not permit null elements (optional), or if the specified collection is null**See Also:**[remove(Object)](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html#remove(java.lang.Object))

### iterator

public [Iterator](http://docs.google.com/java/util/Iterator.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **iterator**()

Returns an iterator over the elements contained in this set in the order in which these elements were added.

The returned iterator provides a snapshot of the state of the set when the iterator was constructed. No synchronization is needed while traversing the iterator. The iterator does *NOT* support the remove method.

**Specified by:**[iterator](http://docs.google.com/java/lang/Iterable.html#iterator()) in interface [Iterable](http://docs.google.com/java/lang/Iterable.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[iterator](http://docs.google.com/java/util/Collection.html#iterator()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[iterator](http://docs.google.com/java/util/Set.html#iterator()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[iterator](http://docs.google.com/java/util/AbstractCollection.html#iterator()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Returns:**an iterator over the elements in this set

### equals

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

Compares the specified object with this set for equality. Returns true if the specified object is the same object as this object, or if it is also a [Set](http://docs.google.com/java/util/Set.html) and the elements returned by an [iterator](http://docs.google.com/java/util/List.html#iterator()) over the specified set are the same as the elements returned by an iterator over this set. More formally, the two iterators are considered to return the same elements if they return the same number of elements and for every element e1 returned by the iterator over the specified set, there is an element e2 returned by the iterator over this set such that (e1==null ? e2==null : e1.equals(e2)).

**Specified by:**[equals](http://docs.google.com/java/util/Collection.html#equals(java.lang.Object)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Specified by:**[equals](http://docs.google.com/java/util/Set.html#equals(java.lang.Object)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)>**Overrides:**[equals](http://docs.google.com/java/util/AbstractSet.html#equals(java.lang.Object)) in class [AbstractSet](http://docs.google.com/java/util/AbstractSet.html)<[E](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)> **Parameters:**o - object to be compared for equality with this set **Returns:**true if the specified object is equal to this set**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CopyOnWriteArraySet.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/CopyOnWriteArrayList.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/CountDownLatch.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/CopyOnWriteArraySet.html)    [**NO FRAMES**](http://docs.google.com/CopyOnWriteArraySet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [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).