| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | Class | Use | **Tree** | [**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**](http://docs.google.com/java/util/package-tree.html)   [**NEXT**](http://docs.google.com/java/util/concurrent/atomic/package-tree.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/package-tree.html)    [**NO FRAMES**](http://docs.google.com/package-tree.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

Hierarchy For Package java.util.concurrent

**Package Hierarchies:**[All Packages](http://docs.google.com/overview-tree.html)

## Class Hierarchy

* java.lang.[**Object**](http://docs.google.com/java/lang/Object.html)
  + java.util.[**AbstractCollection**](http://docs.google.com/java/util/AbstractCollection.html)<E> (implements java.util.[Collection](http://docs.google.com/java/util/Collection.html)<E>)
    - java.util.[**AbstractQueue**](http://docs.google.com/java/util/AbstractQueue.html)<E> (implements java.util.[Queue](http://docs.google.com/java/util/Queue.html)<E>)
      * java.util.concurrent.[**ArrayBlockingQueue**](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html)<E> (implements java.util.concurrent.[BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<E>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
      * java.util.concurrent.[**ConcurrentLinkedQueue**](http://docs.google.com/java/util/concurrent/ConcurrentLinkedQueue.html)<E> (implements java.util.[Queue](http://docs.google.com/java/util/Queue.html)<E>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
      * java.util.concurrent.[**DelayQueue**](http://docs.google.com/java/util/concurrent/DelayQueue.html)<E> (implements java.util.concurrent.[BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<E>)
      * java.util.concurrent.[**LinkedBlockingDeque**](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html)<E> (implements java.util.concurrent.[BlockingDeque](http://docs.google.com/java/util/concurrent/BlockingDeque.html)<E>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
      * java.util.concurrent.[**LinkedBlockingQueue**](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html)<E> (implements java.util.concurrent.[BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<E>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
      * java.util.concurrent.[**PriorityBlockingQueue**](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html)<E> (implements java.util.concurrent.[BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<E>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
      * java.util.concurrent.[**SynchronousQueue**](http://docs.google.com/java/util/concurrent/SynchronousQueue.html)<E> (implements java.util.concurrent.[BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<E>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
    - java.util.[**AbstractSet**](http://docs.google.com/java/util/AbstractSet.html)<E> (implements java.util.[Set](http://docs.google.com/java/util/Set.html)<E>)
      * java.util.concurrent.[**ConcurrentSkipListSet**](http://docs.google.com/java/util/concurrent/ConcurrentSkipListSet.html)<E> (implements java.lang.[Cloneable](http://docs.google.com/java/lang/Cloneable.html), java.util.[NavigableSet](http://docs.google.com/java/util/NavigableSet.html)<E>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
      * java.util.concurrent.[**CopyOnWriteArraySet**](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html)<E> (implements java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
  + java.util.concurrent.[**AbstractExecutorService**](http://docs.google.com/java/util/concurrent/AbstractExecutorService.html) (implements java.util.concurrent.[ExecutorService](http://docs.google.com/java/util/concurrent/ExecutorService.html))
    - java.util.concurrent.[**ThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html)
      * java.util.concurrent.[**ScheduledThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ScheduledThreadPoolExecutor.html) (implements java.util.concurrent.[ScheduledExecutorService](http://docs.google.com/java/util/concurrent/ScheduledExecutorService.html))
  + java.util.[**AbstractMap**](http://docs.google.com/java/util/AbstractMap.html)<K,V> (implements java.util.[Map](http://docs.google.com/java/util/Map.html)<K,V>)
    - java.util.concurrent.[**ConcurrentHashMap**](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html)<K,V> (implements java.util.concurrent.[ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<K,V>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
    - java.util.concurrent.[**ConcurrentSkipListMap**](http://docs.google.com/java/util/concurrent/ConcurrentSkipListMap.html)<K,V> (implements java.lang.[Cloneable](http://docs.google.com/java/lang/Cloneable.html), java.util.concurrent.[ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<K,V>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
  + java.util.concurrent.[**CopyOnWriteArrayList**](http://docs.google.com/java/util/concurrent/CopyOnWriteArrayList.html)<E> (implements java.lang.[Cloneable](http://docs.google.com/java/lang/Cloneable.html), java.util.[List](http://docs.google.com/java/util/List.html)<E>, java.util.[RandomAccess](http://docs.google.com/java/util/RandomAccess.html), java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
  + java.util.concurrent.[**CountDownLatch**](http://docs.google.com/java/util/concurrent/CountDownLatch.html)
  + java.util.concurrent.[**CyclicBarrier**](http://docs.google.com/java/util/concurrent/CyclicBarrier.html)
  + java.util.concurrent.[**Exchanger**](http://docs.google.com/java/util/concurrent/Exchanger.html)<V>
  + java.util.concurrent.[**ExecutorCompletionService**](http://docs.google.com/java/util/concurrent/ExecutorCompletionService.html)<V> (implements java.util.concurrent.[CompletionService](http://docs.google.com/java/util/concurrent/CompletionService.html)<V>)
  + java.util.concurrent.[**Executors**](http://docs.google.com/java/util/concurrent/Executors.html)
  + java.util.concurrent.[**FutureTask**](http://docs.google.com/java/util/concurrent/FutureTask.html)<V> (implements java.util.concurrent.[RunnableFuture](http://docs.google.com/java/util/concurrent/RunnableFuture.html)<V>)
  + java.util.concurrent.[**Semaphore**](http://docs.google.com/java/util/concurrent/Semaphore.html) (implements java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
  + java.util.concurrent.[**ThreadPoolExecutor.AbortPolicy**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.AbortPolicy.html) (implements java.util.concurrent.[RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html))
  + java.util.concurrent.[**ThreadPoolExecutor.CallerRunsPolicy**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.CallerRunsPolicy.html) (implements java.util.concurrent.[RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html))
  + java.util.concurrent.[**ThreadPoolExecutor.DiscardOldestPolicy**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.DiscardOldestPolicy.html) (implements java.util.concurrent.[RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html))
  + java.util.concurrent.[**ThreadPoolExecutor.DiscardPolicy**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.DiscardPolicy.html) (implements java.util.concurrent.[RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html))
  + java.lang.[**Throwable**](http://docs.google.com/java/lang/Throwable.html) (implements java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
    - java.lang.[**Exception**](http://docs.google.com/java/lang/Exception.html)
      * java.util.concurrent.[**BrokenBarrierException**](http://docs.google.com/java/util/concurrent/BrokenBarrierException.html)
      * java.util.concurrent.[**ExecutionException**](http://docs.google.com/java/util/concurrent/ExecutionException.html)
      * java.lang.[**RuntimeException**](http://docs.google.com/java/lang/RuntimeException.html)
        + java.lang.[**IllegalStateException**](http://docs.google.com/java/lang/IllegalStateException.html)

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

* + - * + java.util.concurrent.[**RejectedExecutionException**](http://docs.google.com/java/util/concurrent/RejectedExecutionException.html)
      * java.util.concurrent.[**TimeoutException**](http://docs.google.com/java/util/concurrent/TimeoutException.html)

## Interface Hierarchy

* java.util.concurrent.[**Callable**](http://docs.google.com/java/util/concurrent/Callable.html)<V>
* java.lang.[**Comparable**](http://docs.google.com/java/lang/Comparable.html)<T>
  + java.util.concurrent.[**Delayed**](http://docs.google.com/java/util/concurrent/Delayed.html)
    - java.util.concurrent.[**ScheduledFuture**](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<V> (also extends java.util.concurrent.[Future](http://docs.google.com/java/util/concurrent/Future.html)<V>)
      * java.util.concurrent.[**RunnableScheduledFuture**](http://docs.google.com/java/util/concurrent/RunnableScheduledFuture.html)<V> (also extends java.util.concurrent.[RunnableFuture](http://docs.google.com/java/util/concurrent/RunnableFuture.html)<V>)
* java.util.concurrent.[**CompletionService**](http://docs.google.com/java/util/concurrent/CompletionService.html)<V>
* java.util.concurrent.[**Executor**](http://docs.google.com/java/util/concurrent/Executor.html)
  + java.util.concurrent.[**ExecutorService**](http://docs.google.com/java/util/concurrent/ExecutorService.html)
    - java.util.concurrent.[**ScheduledExecutorService**](http://docs.google.com/java/util/concurrent/ScheduledExecutorService.html)
* java.util.concurrent.[**Future**](http://docs.google.com/java/util/concurrent/Future.html)<V>
  + java.util.concurrent.[**RunnableFuture**](http://docs.google.com/java/util/concurrent/RunnableFuture.html)<V> (also extends java.lang.[Runnable](http://docs.google.com/java/lang/Runnable.html))
    - java.util.concurrent.[**RunnableScheduledFuture**](http://docs.google.com/java/util/concurrent/RunnableScheduledFuture.html)<V> (also extends java.util.concurrent.[ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<V>)
  + java.util.concurrent.[**ScheduledFuture**](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<V> (also extends java.util.concurrent.[Delayed](http://docs.google.com/java/util/concurrent/Delayed.html))
    - java.util.concurrent.[**RunnableScheduledFuture**](http://docs.google.com/java/util/concurrent/RunnableScheduledFuture.html)<V> (also extends java.util.concurrent.[RunnableFuture](http://docs.google.com/java/util/concurrent/RunnableFuture.html)<V>)
* java.lang.[**Iterable**](http://docs.google.com/java/lang/Iterable.html)<T>
  + java.util.[**Collection**](http://docs.google.com/java/util/Collection.html)<E>
    - java.util.[**Queue**](http://docs.google.com/java/util/Queue.html)<E>
      * java.util.concurrent.[**BlockingQueue**](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<E>
        + java.util.concurrent.[**BlockingDeque**](http://docs.google.com/java/util/concurrent/BlockingDeque.html)<E> (also extends java.util.[Deque](http://docs.google.com/java/util/Deque.html)<E>)
      * java.util.[**Deque**](http://docs.google.com/java/util/Deque.html)<E>
        + java.util.concurrent.[**BlockingDeque**](http://docs.google.com/java/util/concurrent/BlockingDeque.html)<E> (also extends java.util.concurrent.[BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<E>)
* java.util.[**Map**](http://docs.google.com/java/util/Map.html)<K,V>
  + java.util.concurrent.[**ConcurrentMap**](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<K,V>
    - java.util.concurrent.[**ConcurrentNavigableMap**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<K,V> (also extends java.util.[NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<K,V>)
  + java.util.[**SortedMap**](http://docs.google.com/java/util/SortedMap.html)<K,V>
    - java.util.[**NavigableMap**](http://docs.google.com/java/util/NavigableMap.html)<K,V>
      * java.util.concurrent.[**ConcurrentNavigableMap**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<K,V> (also extends java.util.concurrent.[ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<K,V>)
* java.util.concurrent.[**RejectedExecutionHandler**](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html)
* java.lang.[**Runnable**](http://docs.google.com/java/lang/Runnable.html)
  + java.util.concurrent.[**RunnableFuture**](http://docs.google.com/java/util/concurrent/RunnableFuture.html)<V> (also extends java.util.concurrent.[Future](http://docs.google.com/java/util/concurrent/Future.html)<V>)
    - java.util.concurrent.[**RunnableScheduledFuture**](http://docs.google.com/java/util/concurrent/RunnableScheduledFuture.html)<V> (also extends java.util.concurrent.[ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<V>)
* java.util.concurrent.[**ThreadFactory**](http://docs.google.com/java/util/concurrent/ThreadFactory.html)

## Enum Hierarchy

* java.lang.[**Object**](http://docs.google.com/java/lang/Object.html)
  + java.lang.[**Enum**](http://docs.google.com/java/lang/Enum.html)<E> (implements java.lang.[Comparable](http://docs.google.com/java/lang/Comparable.html)<T>, java.io.[Serializable](http://docs.google.com/java/io/Serializable.html))
    - java.util.concurrent.[**TimeUnit**](http://docs.google.com/java/util/concurrent/TimeUnit.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | Class | Use | **Tree** | [**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**](http://docs.google.com/java/util/package-tree.html)   [**NEXT**](http://docs.google.com/java/util/concurrent/atomic/package-tree.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/package-tree.html)    [**NO FRAMES**](http://docs.google.com/package-tree.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).