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

**Uses of Class**

**java.util.concurrent.ThreadPoolExecutor**

| Packages that use [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) | |
| --- | --- |
| [**java.util.concurrent**](#3znysh7) | Utility classes commonly useful in concurrent programming. |

| Uses of [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) | |
| --- | --- |

| Subclasses of [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) | |
| --- | --- |
| class | [**ScheduledThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ScheduledThreadPoolExecutor.html)            A [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) that can additionally schedule commands to run after a given delay, or to execute periodically. |

| Methods in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) with parameters of type [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) | |
| --- | --- |
| void | **ThreadPoolExecutor.CallerRunsPolicy.**[**rejectedExecution**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.CallerRunsPolicy.html#rejectedExecution(java.lang.Runnable,%20java.util.concurrent.ThreadPoolExecutor))([Runnable](http://docs.google.com/java/lang/Runnable.html) r, [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) e)            Executes task r in the caller's thread, unless the executor has been shut down, in which case the task is discarded. |
| void | **ThreadPoolExecutor.AbortPolicy.**[**rejectedExecution**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.AbortPolicy.html#rejectedExecution(java.lang.Runnable,%20java.util.concurrent.ThreadPoolExecutor))([Runnable](http://docs.google.com/java/lang/Runnable.html) r, [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) e)            Always throws RejectedExecutionException. |
| void | **ThreadPoolExecutor.DiscardPolicy.**[**rejectedExecution**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.DiscardPolicy.html#rejectedExecution(java.lang.Runnable,%20java.util.concurrent.ThreadPoolExecutor))([Runnable](http://docs.google.com/java/lang/Runnable.html) r, [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) e)            Does nothing, which has the effect of discarding task r. |
| void | **ThreadPoolExecutor.DiscardOldestPolicy.**[**rejectedExecution**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.DiscardOldestPolicy.html#rejectedExecution(java.lang.Runnable,%20java.util.concurrent.ThreadPoolExecutor))([Runnable](http://docs.google.com/java/lang/Runnable.html) r, [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) e)            Obtains and ignores the next task that the executor would otherwise execute, if one is immediately available, and then retries execution of task r, unless the executor is shut down, in which case task r is instead discarded. |
| void | **RejectedExecutionHandler.**[**rejectedExecution**](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html#rejectedExecution(java.lang.Runnable,%20java.util.concurrent.ThreadPoolExecutor))([Runnable](http://docs.google.com/java/lang/Runnable.html) r, [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) executor)            Method that may be invoked by a [ThreadPoolExecutor](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) when [execute](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#execute(java.lang.Runnable)) cannot accept a task. |

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html) | **Use** | [**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   NEXT | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent//class-useThreadPoolExecutor.html)    [**NO FRAMES**](http://docs.google.com/ThreadPoolExecutor.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).