| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.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-useRejectedExecutionHandler.html)    [**NO FRAMES**](http://docs.google.com/RejectedExecutionHandler.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

**Uses of Interface**

**java.util.concurrent.RejectedExecutionHandler**

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

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

| Classes in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) that implement [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) | |
| --- | --- |
| static class | [**ThreadPoolExecutor.AbortPolicy**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.AbortPolicy.html)            A handler for rejected tasks that throws a RejectedExecutionException. |
| static class | [**ThreadPoolExecutor.CallerRunsPolicy**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.CallerRunsPolicy.html)            A handler for rejected tasks that runs the rejected task directly in the calling thread of the execute method, unless the executor has been shut down, in which case the task is discarded. |
| static class | [**ThreadPoolExecutor.DiscardOldestPolicy**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.DiscardOldestPolicy.html)            A handler for rejected tasks that discards the oldest unhandled request and then retries execute, unless the executor is shut down, in which case the task is discarded. |
| static class | [**ThreadPoolExecutor.DiscardPolicy**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.DiscardPolicy.html)            A handler for rejected tasks that silently discards the rejected task. |

| Methods in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) that return [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) | |
| --- | --- |
| [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) | **ThreadPoolExecutor.**[**getRejectedExecutionHandler**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#getRejectedExecutionHandler())()            Returns the current handler for unexecutable tasks. |

| Methods in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) with parameters of type [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) | |
| --- | --- |
| void | **ThreadPoolExecutor.**[**setRejectedExecutionHandler**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#setRejectedExecutionHandler(java.util.concurrent.RejectedExecutionHandler))([RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) handler)            Sets a new handler for unexecutable tasks. |

| Constructors in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) with parameters of type [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) | |
| --- | --- |
| [**ScheduledThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ScheduledThreadPoolExecutor.html#ScheduledThreadPoolExecutor(int,%20java.util.concurrent.RejectedExecutionHandler))(int corePoolSize, [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) handler)            Creates a new ScheduledThreadPoolExecutor with the given initial parameters. |
| [**ScheduledThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ScheduledThreadPoolExecutor.html#ScheduledThreadPoolExecutor(int,%20java.util.concurrent.ThreadFactory,%20java.util.concurrent.RejectedExecutionHandler))(int corePoolSize, [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory, [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) handler)            Creates a new ScheduledThreadPoolExecutor with the given initial parameters. |
| [**ThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#ThreadPoolExecutor(int,%20int,%20long,%20java.util.concurrent.TimeUnit,%20java.util.concurrent.BlockingQueue,%20java.util.concurrent.RejectedExecutionHandler))(int corePoolSize, int maximumPoolSize, long keepAliveTime, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit, [BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<[Runnable](http://docs.google.com/java/lang/Runnable.html)> workQueue, [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) handler)            Creates a new ThreadPoolExecutor with the given initial parameters and default thread factory. |
| [**ThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#ThreadPoolExecutor(int,%20int,%20long,%20java.util.concurrent.TimeUnit,%20java.util.concurrent.BlockingQueue,%20java.util.concurrent.ThreadFactory,%20java.util.concurrent.RejectedExecutionHandler))(int corePoolSize, int maximumPoolSize, long keepAliveTime, [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html) unit, [BlockingQueue](http://docs.google.com/java/util/concurrent/BlockingQueue.html)<[Runnable](http://docs.google.com/java/lang/Runnable.html)> workQueue, [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory, [RejectedExecutionHandler](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.html) handler)            Creates a new ThreadPoolExecutor with the given initial parameters. |

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/util/concurrent/RejectedExecutionHandler.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-useRejectedExecutionHandler.html)    [**NO FRAMES**](http://docs.google.com/RejectedExecutionHandler.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).