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

**Uses of Interface**

**java.util.concurrent.ThreadFactory**

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

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

| Methods in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) that return [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) | |
| --- | --- |
| static [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) | **Executors.**[**defaultThreadFactory**](http://docs.google.com/java/util/concurrent/Executors.html#defaultThreadFactory())()            Returns a default thread factory used to create new threads. |
| [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) | **ThreadPoolExecutor.**[**getThreadFactory**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#getThreadFactory())()            Returns the thread factory used to create new threads. |
| static [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) | **Executors.**[**privilegedThreadFactory**](http://docs.google.com/java/util/concurrent/Executors.html#privilegedThreadFactory())()            Returns a thread factory used to create new threads that have the same permissions as the current thread. |

| Methods in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) with parameters of type [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) | |
| --- | --- |
| static [ExecutorService](http://docs.google.com/java/util/concurrent/ExecutorService.html) | **Executors.**[**newCachedThreadPool**](http://docs.google.com/java/util/concurrent/Executors.html#newCachedThreadPool(java.util.concurrent.ThreadFactory))([ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory)            Creates a thread pool that creates new threads as needed, but will reuse previously constructed threads when they are available, and uses the provided ThreadFactory to create new threads when needed. |
| static [ExecutorService](http://docs.google.com/java/util/concurrent/ExecutorService.html) | **Executors.**[**newFixedThreadPool**](http://docs.google.com/java/util/concurrent/Executors.html#newFixedThreadPool(int,%20java.util.concurrent.ThreadFactory))(int nThreads, [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory)            Creates a thread pool that reuses a fixed number of threads operating off a shared unbounded queue, using the provided ThreadFactory to create new threads when needed. |
| static [ScheduledExecutorService](http://docs.google.com/java/util/concurrent/ScheduledExecutorService.html) | **Executors.**[**newScheduledThreadPool**](http://docs.google.com/java/util/concurrent/Executors.html#newScheduledThreadPool(int,%20java.util.concurrent.ThreadFactory))(int corePoolSize, [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory)            Creates a thread pool that can schedule commands to run after a given delay, or to execute periodically. |
| static [ExecutorService](http://docs.google.com/java/util/concurrent/ExecutorService.html) | **Executors.**[**newSingleThreadExecutor**](http://docs.google.com/java/util/concurrent/Executors.html#newSingleThreadExecutor(java.util.concurrent.ThreadFactory))([ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory)            Creates an Executor that uses a single worker thread operating off an unbounded queue, and uses the provided ThreadFactory to create a new thread when needed. |
| static [ScheduledExecutorService](http://docs.google.com/java/util/concurrent/ScheduledExecutorService.html) | **Executors.**[**newSingleThreadScheduledExecutor**](http://docs.google.com/java/util/concurrent/Executors.html#newSingleThreadScheduledExecutor(java.util.concurrent.ThreadFactory))([ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory)            Creates a single-threaded executor that can schedule commands to run after a given delay, or to execute periodically. |
| void | **ThreadPoolExecutor.**[**setThreadFactory**](http://docs.google.com/java/util/concurrent/ThreadPoolExecutor.html#setThreadFactory(java.util.concurrent.ThreadFactory))([ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory)            Sets the thread factory used to create new threads. |

| Constructors in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) with parameters of type [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) | |
| --- | --- |
| [**ScheduledThreadPoolExecutor**](http://docs.google.com/java/util/concurrent/ScheduledThreadPoolExecutor.html#ScheduledThreadPoolExecutor(int,%20java.util.concurrent.ThreadFactory))(int corePoolSize, [ThreadFactory](http://docs.google.com/java/util/concurrent/ThreadFactory.html) threadFactory)            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.ThreadFactory))(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)            Creates a new ThreadPoolExecutor with the given initial parameters and default rejected execution handler. |
| [**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/ThreadFactory.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-useThreadFactory.html)    [**NO FRAMES**](http://docs.google.com/ThreadFactory.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).