| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ThreadInfo.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/lang/management/RuntimeMXBean.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/management/ThreadMXBean.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/management/ThreadInfo.html)    [**NO FRAMES**](http://docs.google.com/ThreadInfo.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#2et92p0) | DETAIL: FIELD | CONSTR | [METHOD](#3dy6vkm) |

## **java.lang.management**

Class ThreadInfo

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

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

Thread information. ThreadInfo contains the information about a thread including:

#### General thread information

* Thread ID.
* Name of the thread.

#### Execution information

* Thread state.
* The object upon which the thread is blocked due to:
  + waiting to enter a synchronization block/method, or
  + waiting to be notified in a [Object.wait](http://docs.google.com/java/lang/Object.html#wait(long)) method, or
  + parking due to a [LockSupport.park](http://docs.google.com/java/util/concurrent/locks/LockSupport.html#park(java.lang.Object)) call.
* The ID of the thread that owns the object that the thread is blocked.
* Stack trace of the thread.
* List of object monitors locked by the thread.
* List of  [ownable synchronizers](http://docs.google.com/LockInfo.html#OwnableSynchronizer) locked by the thread.

#### Synchronization Statistics

* The number of times that the thread has blocked for synchronization or waited for notification.
* The accumulated elapsed time that the thread has blocked for synchronization or waited for notification since [thread contention monitoring](http://docs.google.com/java/lang/management/ThreadMXBean.html#setThreadContentionMonitoringEnabled(boolean)) was enabled. Some Java virtual machine implementation may not support this. The [ThreadMXBean.isThreadContentionMonitoringSupported()](http://docs.google.com/java/lang/management/ThreadMXBean.html#isThreadContentionMonitoringSupported()) method can be used to determine if a Java virtual machine supports this.

This thread information class is designed for use in monitoring of the system, not for synchronization control.

#### MXBean Mapping

ThreadInfo is mapped to a [CompositeData](http://docs.google.com/javax/management/openmbean/CompositeData.html) with attributes as specified in the [from](http://docs.google.com/java/lang/management/ThreadInfo.html#from(javax.management.openmbean.CompositeData)) method.

**Since:** 1.5 **See Also:**[ThreadMXBean.getThreadInfo(long)](http://docs.google.com/java/lang/management/ThreadMXBean.html#getThreadInfo(long)), [ThreadMXBean.dumpAllThreads(boolean, boolean)](http://docs.google.com/java/lang/management/ThreadMXBean.html#dumpAllThreads(boolean,%20boolean))

| **Method Summary** | |
| --- | --- |
| static [ThreadInfo](http://docs.google.com/java/lang/management/ThreadInfo.html) | [**from**](http://docs.google.com/java/lang/management/ThreadInfo.html#from(javax.management.openmbean.CompositeData))([CompositeData](http://docs.google.com/javax/management/openmbean/CompositeData.html) cd)            Returns a ThreadInfo object represented by the given CompositeData. |
| long | [**getBlockedCount**](http://docs.google.com/java/lang/management/ThreadInfo.html#getBlockedCount())()            Returns the total number of times that the thread associated with this ThreadInfo blocked to enter or reenter a monitor. |
| long | [**getBlockedTime**](http://docs.google.com/java/lang/management/ThreadInfo.html#getBlockedTime())()            Returns the approximate accumulated elapsed time (in milliseconds) that the thread associated with this ThreadInfo has blocked to enter or reenter a monitor since thread contention monitoring is enabled. |
| [MonitorInfo](http://docs.google.com/java/lang/management/MonitorInfo.html)[] | [**getLockedMonitors**](http://docs.google.com/java/lang/management/ThreadInfo.html#getLockedMonitors())()            Returns an array of [MonitorInfo](http://docs.google.com/java/lang/management/MonitorInfo.html) objects, each of which represents an object monitor currently locked by the thread associated with this ThreadInfo. |
| [LockInfo](http://docs.google.com/java/lang/management/LockInfo.html)[] | [**getLockedSynchronizers**](http://docs.google.com/java/lang/management/ThreadInfo.html#getLockedSynchronizers())()            Returns an array of [LockInfo](http://docs.google.com/java/lang/management/LockInfo.html) objects, each of which represents an [ownable synchronizer](http://docs.google.com/LockInfo.html#OwnableSynchronizer) currently locked by the thread associated with this ThreadInfo. |
| [LockInfo](http://docs.google.com/java/lang/management/LockInfo.html) | [**getLockInfo**](http://docs.google.com/java/lang/management/ThreadInfo.html#getLockInfo())()            Returns the LockInfo of an object for which the thread associated with this ThreadInfo is blocked waiting. |
| [String](http://docs.google.com/java/lang/String.html) | [**getLockName**](http://docs.google.com/java/lang/management/ThreadInfo.html#getLockName())()            Returns the [string representation](http://docs.google.com/java/lang/management/LockInfo.html#toString()) of an object for which the thread associated with this ThreadInfo is blocked waiting. |
| long | [**getLockOwnerId**](http://docs.google.com/java/lang/management/ThreadInfo.html#getLockOwnerId())()            Returns the ID of the thread which owns the object for which the thread associated with this ThreadInfo is blocked waiting. |
| [String](http://docs.google.com/java/lang/String.html) | [**getLockOwnerName**](http://docs.google.com/java/lang/management/ThreadInfo.html#getLockOwnerName())()            Returns the name of the thread which owns the object for which the thread associated with this ThreadInfo is blocked waiting. |
| [StackTraceElement](http://docs.google.com/java/lang/StackTraceElement.html)[] | [**getStackTrace**](http://docs.google.com/java/lang/management/ThreadInfo.html#getStackTrace())()            Returns the stack trace of the thread associated with this ThreadInfo. |
| long | [**getThreadId**](http://docs.google.com/java/lang/management/ThreadInfo.html#getThreadId())()            Returns the ID of the thread associated with this ThreadInfo. |
| [String](http://docs.google.com/java/lang/String.html) | [**getThreadName**](http://docs.google.com/java/lang/management/ThreadInfo.html#getThreadName())()            Returns the name of the thread associated with this ThreadInfo. |
| [Thread.State](http://docs.google.com/java/lang/Thread.State.html) | [**getThreadState**](http://docs.google.com/java/lang/management/ThreadInfo.html#getThreadState())()            Returns the state of the thread associated with this ThreadInfo. |
| long | [**getWaitedCount**](http://docs.google.com/java/lang/management/ThreadInfo.html#getWaitedCount())()            Returns the total number of times that the thread associated with this ThreadInfo waited for notification. |
| long | [**getWaitedTime**](http://docs.google.com/java/lang/management/ThreadInfo.html#getWaitedTime())()            Returns the approximate accumulated elapsed time (in milliseconds) that the thread associated with this ThreadInfo has waited for notification since thread contention monitoring is enabled. |
| boolean | [**isInNative**](http://docs.google.com/java/lang/management/ThreadInfo.html#isInNative())()            Tests if the thread associated with this ThreadInfo is executing native code via the Java Native Interface (JNI). |
| boolean | [**isSuspended**](http://docs.google.com/java/lang/management/ThreadInfo.html#isSuspended())()            Tests if the thread associated with this ThreadInfo is suspended. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/lang/management/ThreadInfo.html#toString())()            Returns a string representation of this thread info. |

| **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()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [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)) |

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

### getThreadId

public long **getThreadId**()

Returns the ID of the thread associated with this ThreadInfo.

**Returns:**the ID of the associated thread.

### getThreadName

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

Returns the name of the thread associated with this ThreadInfo.

**Returns:**the name of the associated thread.

### getThreadState

public [Thread.State](http://docs.google.com/java/lang/Thread.State.html) **getThreadState**()

Returns the state of the thread associated with this ThreadInfo.

**Returns:**Thread.State of the associated thread.

### getBlockedTime

public long **getBlockedTime**()

Returns the approximate accumulated elapsed time (in milliseconds) that the thread associated with this ThreadInfo has blocked to enter or reenter a monitor since thread contention monitoring is enabled. I.e. the total accumulated time the thread has been in the [BLOCKED](http://docs.google.com/java/lang/Thread.State.html#BLOCKED) state since thread contention monitoring was last enabled. This method returns -1 if thread contention monitoring is disabled.

The Java virtual machine may measure the time with a high resolution timer. This statistic is reset when the thread contention monitoring is reenabled.

**Returns:**the approximate accumulated elapsed time in milliseconds that a thread entered the BLOCKED state; -1 if thread contention monitoring is disabled. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the Java virtual machine does not support this operation.**See Also:**[ThreadMXBean.isThreadContentionMonitoringSupported()](http://docs.google.com/java/lang/management/ThreadMXBean.html#isThreadContentionMonitoringSupported()), [ThreadMXBean.setThreadContentionMonitoringEnabled(boolean)](http://docs.google.com/java/lang/management/ThreadMXBean.html#setThreadContentionMonitoringEnabled(boolean))

### getBlockedCount

public long **getBlockedCount**()

Returns the total number of times that the thread associated with this ThreadInfo blocked to enter or reenter a monitor. I.e. the number of times a thread has been in the [BLOCKED](http://docs.google.com/java/lang/Thread.State.html#BLOCKED) state.

**Returns:**the total number of times that the thread entered the BLOCKED state.

### getWaitedTime

public long **getWaitedTime**()

Returns the approximate accumulated elapsed time (in milliseconds) that the thread associated with this ThreadInfo has waited for notification since thread contention monitoring is enabled. I.e. the total accumulated time the thread has been in the [WAITING](http://docs.google.com/java/lang/Thread.State.html#WAITING) or [TIMED\_WAITING](http://docs.google.com/java/lang/Thread.State.html#TIMED_WAITING) state since thread contention monitoring is enabled. This method returns -1 if thread contention monitoring is disabled.

The Java virtual machine may measure the time with a high resolution timer. This statistic is reset when the thread contention monitoring is reenabled.

**Returns:**the approximate accumulated elapsed time in milliseconds that a thread has been in the WAITING or TIMED\_WAITING state; -1 if thread contention monitoring is disabled. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the Java virtual machine does not support this operation.**See Also:**[ThreadMXBean.isThreadContentionMonitoringSupported()](http://docs.google.com/java/lang/management/ThreadMXBean.html#isThreadContentionMonitoringSupported()), [ThreadMXBean.setThreadContentionMonitoringEnabled(boolean)](http://docs.google.com/java/lang/management/ThreadMXBean.html#setThreadContentionMonitoringEnabled(boolean))

### getWaitedCount

public long **getWaitedCount**()

Returns the total number of times that the thread associated with this ThreadInfo waited for notification. I.e. the number of times that a thread has been in the [WAITING](http://docs.google.com/java/lang/Thread.State.html#WAITING) or [TIMED\_WAITING](http://docs.google.com/java/lang/Thread.State.html#TIMED_WAITING) state.

**Returns:**the total number of times that the thread was in the WAITING or TIMED\_WAITING state.

### getLockInfo

public [LockInfo](http://docs.google.com/java/lang/management/LockInfo.html) **getLockInfo**()

Returns the LockInfo of an object for which the thread associated with this ThreadInfo is blocked waiting. A thread can be blocked waiting for one of the following:

* an object monitor to be acquired for entering or reentering a synchronization block/method.  
  The thread is in the [BLOCKED](http://docs.google.com/java/lang/Thread.State.html#BLOCKED) state waiting to enter the synchronized statement or method.
* an object monitor to be notified by another thread.  
  The thread is in the [WAITING](http://docs.google.com/java/lang/Thread.State.html#WAITING) or [TIMED\_WAITING](http://docs.google.com/java/lang/Thread.State.html#TIMED_WAITING) state due to a call to the [Object.wait](http://docs.google.com/java/lang/Object.html#wait(long)) method.
* a synchronization object responsible for the thread parking.  
  The thread is in the [WAITING](http://docs.google.com/java/lang/Thread.State.html#WAITING) or [TIMED\_WAITING](http://docs.google.com/java/lang/Thread.State.html#TIMED_WAITING) state due to a call to the [LockSupport.park](http://docs.google.com/java/util/concurrent/locks/LockSupport.html#park(java.lang.Object)) method. The synchronization object is the object returned from [LockSupport.getBlocker](http://docs.google.com/java/util/concurrent/locks/LockSupport.html#getBlocker(java.lang.Thread)) method. Typically it is an  [ownable synchronizer](http://docs.google.com/LockInfo.html#OwnableSynchronizer) or a [Condition](http://docs.google.com/java/util/concurrent/locks/Condition.html).

This method returns null if the thread is not in any of the above conditions.

**Returns:**LockInfo of an object for which the thread is blocked waiting if any; null otherwise.**Since:** 1.6

### getLockName

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

Returns the [string representation](http://docs.google.com/java/lang/management/LockInfo.html#toString()) of an object for which the thread associated with this ThreadInfo is blocked waiting. This method is equivalent to calling:

getLockInfo().toString()

This method will return null if this thread is not blocked waiting for any object or if the object is not owned by any thread.

**Returns:**the string representation of the object on which the thread is blocked if any; null otherwise.**See Also:**[getLockInfo()](http://docs.google.com/java/lang/management/ThreadInfo.html#getLockInfo())

### getLockOwnerId

public long **getLockOwnerId**()

Returns the ID of the thread which owns the object for which the thread associated with this ThreadInfo is blocked waiting. This method will return -1 if this thread is not blocked waiting for any object or if the object is not owned by any thread.

**Returns:**the thread ID of the owner thread of the object this thread is blocked on; -1 if this thread is not blocked or if the object lis not owned by any thread.**See Also:**[getLockInfo()](http://docs.google.com/java/lang/management/ThreadInfo.html#getLockInfo())

### getLockOwnerName

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

Returns the name of the thread which owns the object for which the thread associated with this ThreadInfo is blocked waiting. This method will return null if this thread is not blocked waiting for any object or if the object is not owned by any thread.

**Returns:**the name of the thread that owns the object this thread is blocked on; null if this thread is not blocked or if the object is not owned by any thread.**See Also:**[getLockInfo()](http://docs.google.com/java/lang/management/ThreadInfo.html#getLockInfo())

### getStackTrace

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

Returns the stack trace of the thread associated with this ThreadInfo. If no stack trace was requested for this thread info, this method will return a zero-length array. If the returned array is of non-zero length then the first element of the array represents the top of the stack, which is the most recent method invocation in the sequence. The last element of the array represents the bottom of the stack, which is the least recent method invocation in the sequence.

Some Java virtual machines may, under some circumstances, omit one or more stack frames from the stack trace. In the extreme case, a virtual machine that has no stack trace information concerning the thread associated with this ThreadInfo is permitted to return a zero-length array from this method.

**Returns:**an array of StackTraceElement objects of the thread.

### isSuspended

public boolean **isSuspended**()

Tests if the thread associated with this ThreadInfo is suspended. This method returns true if [Thread.suspend()](http://docs.google.com/java/lang/Thread.html#suspend()) has been called.

**Returns:**true if the thread is suspended; false otherwise.

### isInNative

public boolean **isInNative**()

Tests if the thread associated with this ThreadInfo is executing native code via the Java Native Interface (JNI). The JNI native code does not include the virtual machine support code or the compiled native code generated by the virtual machine.

**Returns:**true if the thread is executing native code; false otherwise.

### toString

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

Returns a string representation of this thread info. The format of this string depends on the implementation. The returned string will typically include the [thread name](http://docs.google.com/java/lang/management/ThreadInfo.html#getThreadName()), the [thread ID](http://docs.google.com/java/lang/management/ThreadInfo.html#getThreadId()), its [state](http://docs.google.com/java/lang/management/ThreadInfo.html#getThreadState()), and a [stack trace](http://docs.google.com/java/lang/management/ThreadInfo.html#getStackTrace()) if any.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of this thread info.

### from

public static [ThreadInfo](http://docs.google.com/java/lang/management/ThreadInfo.html) **from**([CompositeData](http://docs.google.com/javax/management/openmbean/CompositeData.html) cd)

Returns a ThreadInfo object represented by the given CompositeData. The given CompositeData must contain the following attributes unless otherwise specified below:

| Attribute Name | Type |
| --- | --- |
| threadId | java.lang.Long |
| threadName | java.lang.String |
| threadState | java.lang.String |
| suspended | java.lang.Boolean |
| inNative | java.lang.Boolean |
| blockedCount | java.lang.Long |
| blockedTime | java.lang.Long |
| waitedCount | java.lang.Long |
| waitedTime | java.lang.Long |
| lockInfo | javax.management.openmbean.CompositeData - the mapped type for [LockInfo](http://docs.google.com/java/lang/management/LockInfo.html) as specified in the  [type mapping rules](http://docs.google.com/javax/management/MXBean.html#mapping-rules) of [MXBeans](http://docs.google.com/javax/management/MXBean.html).  If cd does not contain this attribute, the LockInfo object will be constructed from the value of the lockName attribute. |
| lockName | java.lang.String |
| lockOwnerId | java.lang.Long |
| lockOwnerName | java.lang.String |
| stackTrace | javax.management.openmbean.CompositeData[]  Each element is a CompositeData representing StackTraceElement containing the following attributes:   | Attribute Name | Type | | --- | --- | | className | java.lang.String | | methodName | java.lang.String | | fileName | java.lang.String | | lineNumber | java.lang.Integer | | nativeMethod | java.lang.Boolean | |
| lockedMonitors | javax.management.openmbean.CompositeData[] whose element type is the mapped type for [MonitorInfo](http://docs.google.com/java/lang/management/MonitorInfo.html) as specified in the [Monitor.from](http://docs.google.com/java/lang/management/MonitorInfo.html#from(javax.management.openmbean.CompositeData)) method.  If cd does not contain this attribute, this attribute will be set to an empty array. |
| lockedSynchronizers | javax.management.openmbean.CompositeData[] whose element type is the mapped type for [LockInfo](http://docs.google.com/java/lang/management/LockInfo.html) as specified in the  [type mapping rules](http://docs.google.com/javax/management/MXBean.html#mapping-rules) of [MXBeans](http://docs.google.com/javax/management/MXBean.html).  If cd does not contain this attribute, this attribute will be set to an empty array. |

**Parameters:**cd - CompositeData representing a ThreadInfo **Returns:**a ThreadInfo object represented by cd if cd is not null; null otherwise. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if cd does not represent a ThreadInfo with the attributes described above.

### getLockedMonitors

public [MonitorInfo](http://docs.google.com/java/lang/management/MonitorInfo.html)[] **getLockedMonitors**()

Returns an array of [MonitorInfo](http://docs.google.com/java/lang/management/MonitorInfo.html) objects, each of which represents an object monitor currently locked by the thread associated with this ThreadInfo. If no locked monitor was requested for this thread info or no monitor is locked by the thread, this method will return a zero-length array.

**Returns:**an array of MonitorInfo objects representing the object monitors locked by the thread.**Since:** 1.6

### getLockedSynchronizers

public [LockInfo](http://docs.google.com/java/lang/management/LockInfo.html)[] **getLockedSynchronizers**()

Returns an array of [LockInfo](http://docs.google.com/java/lang/management/LockInfo.html) objects, each of which represents an [ownable synchronizer](http://docs.google.com/LockInfo.html#OwnableSynchronizer) currently locked by the thread associated with this ThreadInfo. If no locked synchronizer was requested for this thread info or no synchronizer is locked by the thread, this method will return a zero-length array.

**Returns:**an array of LockInfo objects representing the ownable synchronizers locked by the thread.**Since:** 1.6

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ThreadInfo.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/lang/management/RuntimeMXBean.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/management/ThreadMXBean.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/management/ThreadInfo.html)    [**NO FRAMES**](http://docs.google.com/ThreadInfo.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#2et92p0) | DETAIL: FIELD | CONSTR | [METHOD](#3dy6vkm) |

[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).