| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Throwable.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/ThreadLocal.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/TypeNotPresentException.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/Throwable.html)    [**NO FRAMES**](http://docs.google.com/Throwable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

## **java.lang**

Class Throwable

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html) **Direct Known Subclasses:** [Error](http://docs.google.com/java/lang/Error.html), [Exception](http://docs.google.com/java/lang/Exception.html)

public class **Throwable**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html)

The Throwable class is the superclass of all errors and exceptions in the Java language. Only objects that are instances of this class (or one of its subclasses) are thrown by the Java Virtual Machine or can be thrown by the Java throw statement. Similarly, only this class or one of its subclasses can be the argument type in a catch clause.

Instances of two subclasses, [Error](http://docs.google.com/java/lang/Error.html) and [Exception](http://docs.google.com/java/lang/Exception.html), are conventionally used to indicate that exceptional situations have occurred. Typically, these instances are freshly created in the context of the exceptional situation so as to include relevant information (such as stack trace data).

A throwable contains a snapshot of the execution stack of its thread at the time it was created. It can also contain a message string that gives more information about the error. Finally, it can contain a *cause*: another throwable that caused this throwable to get thrown. The cause facility is new in release 1.4. It is also known as the *chained exception* facility, as the cause can, itself, have a cause, and so on, leading to a "chain" of exceptions, each caused by another.

One reason that a throwable may have a cause is that the class that throws it is built atop a lower layered abstraction, and an operation on the upper layer fails due to a failure in the lower layer. It would be bad design to let the throwable thrown by the lower layer propagate outward, as it is generally unrelated to the abstraction provided by the upper layer. Further, doing so would tie the API of the upper layer to the details of its implementation, assuming the lower layer's exception was a checked exception. Throwing a "wrapped exception" (i.e., an exception containing a cause) allows the upper layer to communicate the details of the failure to its caller without incurring either of these shortcomings. It preserves the flexibility to change the implementation of the upper layer without changing its API (in particular, the set of exceptions thrown by its methods).

A second reason that a throwable may have a cause is that the method that throws it must conform to a general-purpose interface that does not permit the method to throw the cause directly. For example, suppose a persistent collection conforms to the [Collection](http://docs.google.com/java/util/Collection.html) interface, and that its persistence is implemented atop java.io. Suppose the internals of the add method can throw an [IOException](http://docs.google.com/java/io/IOException.html). The implementation can communicate the details of the IOException to its caller while conforming to the Collection interface by wrapping the IOException in an appropriate unchecked exception. (The specification for the persistent collection should indicate that it is capable of throwing such exceptions.)

A cause can be associated with a throwable in two ways: via a constructor that takes the cause as an argument, or via the [initCause(Throwable)](http://docs.google.com/java/lang/Throwable.html#initCause(java.lang.Throwable)) method. New throwable classes that wish to allow causes to be associated with them should provide constructors that take a cause and delegate (perhaps indirectly) to one of the Throwable constructors that takes a cause. For example:

try {  
 lowLevelOp();  
 } catch (LowLevelException le) {  
 throw new HighLevelException(le); // Chaining-aware constructor  
 }

Because the initCause method is public, it allows a cause to be associated with any throwable, even a "legacy throwable" whose implementation predates the addition of the exception chaining mechanism to Throwable. For example:

try {  
 lowLevelOp();  
 } catch (LowLevelException le) {  
 throw (HighLevelException)  
 new HighLevelException().initCause(le); // Legacy constructor  
 }

Prior to release 1.4, there were many throwables that had their own non-standard exception chaining mechanisms ( [ExceptionInInitializerError](http://docs.google.com/java/lang/ExceptionInInitializerError.html), [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html), [UndeclaredThrowableException](http://docs.google.com/java/lang/reflect/UndeclaredThrowableException.html), [InvocationTargetException](http://docs.google.com/java/lang/reflect/InvocationTargetException.html), [WriteAbortedException](http://docs.google.com/java/io/WriteAbortedException.html), [PrivilegedActionException](http://docs.google.com/java/security/PrivilegedActionException.html), [PrinterIOException](http://docs.google.com/java/awt/print/PrinterIOException.html), [RemoteException](http://docs.google.com/java/rmi/RemoteException.html) and [NamingException](http://docs.google.com/javax/naming/NamingException.html)). All of these throwables have been retrofitted to use the standard exception chaining mechanism, while continuing to implement their "legacy" chaining mechanisms for compatibility.

Further, as of release 1.4, many general purpose Throwable classes (for example [Exception](http://docs.google.com/java/lang/Exception.html), [RuntimeException](http://docs.google.com/java/lang/RuntimeException.html), [Error](http://docs.google.com/java/lang/Error.html)) have been retrofitted with constructors that take a cause. This was not strictly necessary, due to the existence of the initCause method, but it is more convenient and expressive to delegate to a constructor that takes a cause.

By convention, class Throwable and its subclasses have two constructors, one that takes no arguments and one that takes a String argument that can be used to produce a detail message. Further, those subclasses that might likely have a cause associated with them should have two more constructors, one that takes a Throwable (the cause), and one that takes a String (the detail message) and a Throwable (the cause).

Also introduced in release 1.4 is the [getStackTrace()](http://docs.google.com/java/lang/Throwable.html#getStackTrace()) method, which allows programmatic access to the stack trace information that was previously available only in text form, via the various forms of the [printStackTrace()](http://docs.google.com/java/lang/Throwable.html#printStackTrace()) method. This information has been added to the *serialized representation* of this class so getStackTrace and printStackTrace will operate properly on a throwable that was obtained by deserialization.

**Since:** JDK1.0 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.lang.Throwable)

| **Constructor Summary** | |
| --- | --- |
| [**Throwable**](http://docs.google.com/java/lang/Throwable.html#Throwable())()            Constructs a new throwable with null as its detail message. |
| [**Throwable**](http://docs.google.com/java/lang/Throwable.html#Throwable(java.lang.String))([String](http://docs.google.com/java/lang/String.html) message)            Constructs a new throwable with the specified detail message. |
| [**Throwable**](http://docs.google.com/java/lang/Throwable.html#Throwable(java.lang.String,%20java.lang.Throwable))([String](http://docs.google.com/java/lang/String.html) message, [Throwable](http://docs.google.com/java/lang/Throwable.html) cause)            Constructs a new throwable with the specified detail message and cause. |
| [**Throwable**](http://docs.google.com/java/lang/Throwable.html#Throwable(java.lang.Throwable))([Throwable](http://docs.google.com/java/lang/Throwable.html) cause)            Constructs a new throwable with the specified cause and a detail message of (cause==null ? null : cause.toString()) (which typically contains the class and detail message of cause). |

| **Method Summary** | |
| --- | --- |
| [Throwable](http://docs.google.com/java/lang/Throwable.html) | [**fillInStackTrace**](http://docs.google.com/java/lang/Throwable.html#fillInStackTrace())()            Fills in the execution stack trace. |
| [Throwable](http://docs.google.com/java/lang/Throwable.html) | [**getCause**](http://docs.google.com/java/lang/Throwable.html#getCause())()            Returns the cause of this throwable or null if the cause is nonexistent or unknown. |
| [String](http://docs.google.com/java/lang/String.html) | [**getLocalizedMessage**](http://docs.google.com/java/lang/Throwable.html#getLocalizedMessage())()            Creates a localized description of this throwable. |
| [String](http://docs.google.com/java/lang/String.html) | [**getMessage**](http://docs.google.com/java/lang/Throwable.html#getMessage())()            Returns the detail message string of this throwable. |
| [StackTraceElement](http://docs.google.com/java/lang/StackTraceElement.html)[] | [**getStackTrace**](http://docs.google.com/java/lang/Throwable.html#getStackTrace())()            Provides programmatic access to the stack trace information printed by [printStackTrace()](http://docs.google.com/java/lang/Throwable.html#printStackTrace()). |
| [Throwable](http://docs.google.com/java/lang/Throwable.html) | [**initCause**](http://docs.google.com/java/lang/Throwable.html#initCause(java.lang.Throwable))([Throwable](http://docs.google.com/java/lang/Throwable.html) cause)            Initializes the *cause* of this throwable to the specified value. |
| void | [**printStackTrace**](http://docs.google.com/java/lang/Throwable.html#printStackTrace())()            Prints this throwable and its backtrace to the standard error stream. |
| void | [**printStackTrace**](http://docs.google.com/java/lang/Throwable.html#printStackTrace(java.io.PrintStream))([PrintStream](http://docs.google.com/java/io/PrintStream.html) s)            Prints this throwable and its backtrace to the specified print stream. |
| void | [**printStackTrace**](http://docs.google.com/java/lang/Throwable.html#printStackTrace(java.io.PrintWriter))([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) s)            Prints this throwable and its backtrace to the specified print writer. |
| void | [**setStackTrace**](http://docs.google.com/java/lang/Throwable.html#setStackTrace(java.lang.StackTraceElement%5B%5D))([StackTraceElement](http://docs.google.com/java/lang/StackTraceElement.html)[] stackTrace)            Sets the stack trace elements that will be returned by [getStackTrace()](http://docs.google.com/java/lang/Throwable.html#getStackTrace()) and printed by [printStackTrace()](http://docs.google.com/java/lang/Throwable.html#printStackTrace()) and related methods. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/lang/Throwable.html#toString())()            Returns a short description of this throwable. |

| **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)) |

| **Constructor Detail** |
| --- |

### Throwable

public **Throwable**()

Constructs a new throwable with null as its detail message. The cause is not initialized, and may subsequently be initialized by a call to [initCause(java.lang.Throwable)](http://docs.google.com/java/lang/Throwable.html#initCause(java.lang.Throwable)).

The [fillInStackTrace()](http://docs.google.com/java/lang/Throwable.html#fillInStackTrace()) method is called to initialize the stack trace data in the newly created throwable.

### Throwable

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

Constructs a new throwable with the specified detail message. The cause is not initialized, and may subsequently be initialized by a call to [initCause(java.lang.Throwable)](http://docs.google.com/java/lang/Throwable.html#initCause(java.lang.Throwable)).

The [fillInStackTrace()](http://docs.google.com/java/lang/Throwable.html#fillInStackTrace()) method is called to initialize the stack trace data in the newly created throwable.

**Parameters:**message - the detail message. The detail message is saved for later retrieval by the [getMessage()](http://docs.google.com/java/lang/Throwable.html#getMessage()) method.

### Throwable

public **Throwable**([String](http://docs.google.com/java/lang/String.html) message,  
 [Throwable](http://docs.google.com/java/lang/Throwable.html) cause)

Constructs a new throwable with the specified detail message and cause.

Note that the detail message associated with cause is *not* automatically incorporated in this throwable's detail message.

The [fillInStackTrace()](http://docs.google.com/java/lang/Throwable.html#fillInStackTrace()) method is called to initialize the stack trace data in the newly created throwable.

**Parameters:**message - the detail message (which is saved for later retrieval by the [getMessage()](http://docs.google.com/java/lang/Throwable.html#getMessage()) method).cause - the cause (which is saved for later retrieval by the [getCause()](http://docs.google.com/java/lang/Throwable.html#getCause()) method). (A null value is permitted, and indicates that the cause is nonexistent or unknown.)**Since:** 1.4

### Throwable

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

Constructs a new throwable with the specified cause and a detail message of (cause==null ? null : cause.toString()) (which typically contains the class and detail message of cause). This constructor is useful for throwables that are little more than wrappers for other throwables (for example, [PrivilegedActionException](http://docs.google.com/java/security/PrivilegedActionException.html)).

The [fillInStackTrace()](http://docs.google.com/java/lang/Throwable.html#fillInStackTrace()) method is called to initialize the stack trace data in the newly created throwable.

**Parameters:**cause - the cause (which is saved for later retrieval by the [getCause()](http://docs.google.com/java/lang/Throwable.html#getCause()) method). (A null value is permitted, and indicates that the cause is nonexistent or unknown.)**Since:** 1.4

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

### getMessage

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

Returns the detail message string of this throwable.

**Returns:**the detail message string of this Throwable instance (which may be null).

### getLocalizedMessage

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

Creates a localized description of this throwable. Subclasses may override this method in order to produce a locale-specific message. For subclasses that do not override this method, the default implementation returns the same result as getMessage().

**Returns:**The localized description of this throwable.**Since:** JDK1.1

### getCause

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

Returns the cause of this throwable or null if the cause is nonexistent or unknown. (The cause is the throwable that caused this throwable to get thrown.)

This implementation returns the cause that was supplied via one of the constructors requiring a Throwable, or that was set after creation with the [initCause(Throwable)](http://docs.google.com/java/lang/Throwable.html#initCause(java.lang.Throwable)) method. While it is typically unnecessary to override this method, a subclass can override it to return a cause set by some other means. This is appropriate for a "legacy chained throwable" that predates the addition of chained exceptions to Throwable. Note that it is *not* necessary to override any of the PrintStackTrace methods, all of which invoke the getCause method to determine the cause of a throwable.

**Returns:**the cause of this throwable or null if the cause is nonexistent or unknown.**Since:** 1.4

### initCause

public [Throwable](http://docs.google.com/java/lang/Throwable.html) **initCause**([Throwable](http://docs.google.com/java/lang/Throwable.html) cause)

Initializes the *cause* of this throwable to the specified value. (The cause is the throwable that caused this throwable to get thrown.)

This method can be called at most once. It is generally called from within the constructor, or immediately after creating the throwable. If this throwable was created with [Throwable(Throwable)](http://docs.google.com/java/lang/Throwable.html#Throwable(java.lang.Throwable)) or [Throwable(String,Throwable)](http://docs.google.com/java/lang/Throwable.html#Throwable(java.lang.String,%20java.lang.Throwable)), this method cannot be called even once.

**Parameters:**cause - the cause (which is saved for later retrieval by the [getCause()](http://docs.google.com/java/lang/Throwable.html#getCause()) method). (A null value is permitted, and indicates that the cause is nonexistent or unknown.) **Returns:**a reference to this Throwable instance. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if cause is this throwable. (A throwable cannot be its own cause.) [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this throwable was created with [Throwable(Throwable)](http://docs.google.com/java/lang/Throwable.html#Throwable(java.lang.Throwable)) or [Throwable(String,Throwable)](http://docs.google.com/java/lang/Throwable.html#Throwable(java.lang.String,%20java.lang.Throwable)), or this method has already been called on this throwable.**Since:** 1.4

### toString

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

Returns a short description of this throwable. The result is the concatenation of:

* the [name](http://docs.google.com/java/lang/Class.html#getName()) of the class of this object
* ": " (a colon and a space)
* the result of invoking this object's [getLocalizedMessage()](http://docs.google.com/java/lang/Throwable.html#getLocalizedMessage()) method

If getLocalizedMessage returns null, then just the class name is returned.

**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 throwable.

### printStackTrace

public void **printStackTrace**()

Prints this throwable and its backtrace to the standard error stream. This method prints a stack trace for this Throwable object on the error output stream that is the value of the field System.err. The first line of output contains the result of the [toString()](http://docs.google.com/java/lang/Throwable.html#toString()) method for this object. Remaining lines represent data previously recorded by the method [fillInStackTrace()](http://docs.google.com/java/lang/Throwable.html#fillInStackTrace()). The format of this information depends on the implementation, but the following example may be regarded as typical:

java.lang.NullPointerException  
 at MyClass.mash(MyClass.java:9)  
 at MyClass.crunch(MyClass.java:6)  
 at MyClass.main(MyClass.java:3)

This example was produced by running the program:

class MyClass {  
 public static void main(String[] args) {  
 crunch(null);  
 }  
 static void crunch(int[] a) {  
 mash(a);  
 }  
 static void mash(int[] b) {  
 System.out.println(b[0]);  
 }  
 }

The backtrace for a throwable with an initialized, non-null cause should generally include the backtrace for the cause. The format of this information depends on the implementation, but the following example may be regarded as typical:

HighLevelException: MidLevelException: LowLevelException  
 at Junk.a(Junk.java:13)  
 at Junk.main(Junk.java:4)  
 Caused by: MidLevelException: LowLevelException  
 at Junk.c(Junk.java:23)  
 at Junk.b(Junk.java:17)  
 at Junk.a(Junk.java:11)  
 ... 1 more  
 Caused by: LowLevelException  
 at Junk.e(Junk.java:30)  
 at Junk.d(Junk.java:27)  
 at Junk.c(Junk.java:21)  
 ... 3 more

Note the presence of lines containing the characters "...". These lines indicate that the remainder of the stack trace for this exception matches the indicated number of frames from the bottom of the stack trace of the exception that was caused by this exception (the "enclosing" exception). This shorthand can greatly reduce the length of the output in the common case where a wrapped exception is thrown from same method as the "causative exception" is caught. The above example was produced by running the program:

public class Junk {  
 public static void main(String args[]) {  
 try {  
 a();  
 } catch(HighLevelException e) {  
 e.printStackTrace();  
 }  
 }  
 static void a() throws HighLevelException {  
 try {  
 b();  
 } catch(MidLevelException e) {  
 throw new HighLevelException(e);  
 }  
 }  
 static void b() throws MidLevelException {  
 c();  
 }  
 static void c() throws MidLevelException {  
 try {  
 d();  
 } catch(LowLevelException e) {  
 throw new MidLevelException(e);  
 }  
 }  
 static void d() throws LowLevelException {  
 e();  
 }  
 static void e() throws LowLevelException {  
 throw new LowLevelException();  
 }  
 }  
  
 class HighLevelException extends Exception {  
 HighLevelException(Throwable cause) { super(cause); }  
 }  
  
 class MidLevelException extends Exception {  
 MidLevelException(Throwable cause) { super(cause); }  
 }  
  
 class LowLevelException extends Exception {  
 }

### printStackTrace

public void **printStackTrace**([PrintStream](http://docs.google.com/java/io/PrintStream.html) s)

Prints this throwable and its backtrace to the specified print stream.

**Parameters:**s - PrintStream to use for output

### printStackTrace

public void **printStackTrace**([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) s)

Prints this throwable and its backtrace to the specified print writer.

**Parameters:**s - PrintWriter to use for output**Since:** JDK1.1

### fillInStackTrace

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

Fills in the execution stack trace. This method records within this Throwable object information about the current state of the stack frames for the current thread.

**Returns:**a reference to this Throwable instance.**See Also:**[printStackTrace()](http://docs.google.com/java/lang/Throwable.html#printStackTrace())

### getStackTrace

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

Provides programmatic access to the stack trace information printed by [printStackTrace()](http://docs.google.com/java/lang/Throwable.html#printStackTrace()). Returns an array of stack trace elements, each representing one stack frame. The zeroth element of the array (assuming the array's length is non-zero) represents the top of the stack, which is the last method invocation in the sequence. Typically, this is the point at which this throwable was created and thrown. The last element of the array (assuming the array's length is non-zero) represents the bottom of the stack, which is the first method invocation in the sequence.

Some 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 this throwable is permitted to return a zero-length array from this method. Generally speaking, the array returned by this method will contain one element for every frame that would be printed by printStackTrace.

**Returns:**an array of stack trace elements representing the stack trace pertaining to this throwable.**Since:** 1.4

### setStackTrace

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

Sets the stack trace elements that will be returned by [getStackTrace()](http://docs.google.com/java/lang/Throwable.html#getStackTrace()) and printed by [printStackTrace()](http://docs.google.com/java/lang/Throwable.html#printStackTrace()) and related methods. This method, which is designed for use by RPC frameworks and other advanced systems, allows the client to override the default stack trace that is either generated by [fillInStackTrace()](http://docs.google.com/java/lang/Throwable.html#fillInStackTrace()) when a throwable is constructed or deserialized when a throwable is read from a serialization stream.

**Parameters:**stackTrace - the stack trace elements to be associated with this Throwable. The specified array is copied by this call; changes in the specified array after the method invocation returns will have no affect on this Throwable's stack trace. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if stackTrace is null, or if any of the elements of stackTrace are null**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Throwable.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/ThreadLocal.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/TypeNotPresentException.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/Throwable.html)    [**NO FRAMES**](http://docs.google.com/Throwable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

[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).