| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ThreadLocal.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/ThreadGroup.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/Throwable.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/ThreadLocal.html)    [**NO FRAMES**](http://docs.google.com/ThreadLocal.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.lang**

Class ThreadLocal<T>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.lang.ThreadLocal<T>**

**Direct Known Subclasses:** [InheritableThreadLocal](http://docs.google.com/java/lang/InheritableThreadLocal.html)

public class **ThreadLocal<T>**extends [Object](http://docs.google.com/java/lang/Object.html)

This class provides thread-local variables. These variables differ from their normal counterparts in that each thread that accesses one (via its get or set method) has its own, independently initialized copy of the variable. ThreadLocal instances are typically private static fields in classes that wish to associate state with a thread (e.g., a user ID or Transaction ID).

For example, the class below generates unique identifiers local to each thread. A thread's id is assigned the first time it invokes UniqueThreadIdGenerator.getCurrentThreadId() and remains unchanged on subsequent calls.

import java.util.concurrent.atomic.AtomicInteger;  
  
 public class UniqueThreadIdGenerator {  
  
 private static final AtomicInteger uniqueId = new AtomicInteger(0);  
  
 private static final ThreadLocal < Integer > uniqueNum =   
 new ThreadLocal < Integer > () {  
 @Override protected Integer initialValue() {  
 return uniqueId.getAndIncrement();  
 }  
 };  
   
 public static int getCurrentThreadId() {  
 return uniqueId.get();  
 }  
 } // UniqueThreadIdGenerator

Each thread holds an implicit reference to its copy of a thread-local variable as long as the thread is alive and the ThreadLocal instance is accessible; after a thread goes away, all of its copies of thread-local instances are subject to garbage collection (unless other references to these copies exist).

**Since:** 1.2

| **Constructor Summary** | |
| --- | --- |
| [**ThreadLocal**](http://docs.google.com/java/lang/ThreadLocal.html#ThreadLocal())()            Creates a thread local variable. |

| **Method Summary** | |
| --- | --- |
| [T](http://docs.google.com/java/lang/ThreadLocal.html) | [**get**](http://docs.google.com/java/lang/ThreadLocal.html#get())()            Returns the value in the current thread's copy of this thread-local variable. |
| protected  [T](http://docs.google.com/java/lang/ThreadLocal.html) | [**initialValue**](http://docs.google.com/java/lang/ThreadLocal.html#initialValue())()            Returns the current thread's "initial value" for this thread-local variable. |
| void | [**remove**](http://docs.google.com/java/lang/ThreadLocal.html#remove())()            Removes the current thread's value for this thread-local variable. |
| void | [**set**](http://docs.google.com/java/lang/ThreadLocal.html#set(T))([T](http://docs.google.com/java/lang/ThreadLocal.html) value)            Sets the current thread's copy of this thread-local variable to the specified value. |

| **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()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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** |
| --- |

### ThreadLocal

public **ThreadLocal**()

Creates a thread local variable.

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

### initialValue

protected [T](http://docs.google.com/java/lang/ThreadLocal.html) **initialValue**()

Returns the current thread's "initial value" for this thread-local variable. This method will be invoked the first time a thread accesses the variable with the [get()](http://docs.google.com/java/lang/ThreadLocal.html#get()) method, unless the thread previously invoked the [set(T)](http://docs.google.com/java/lang/ThreadLocal.html#set(T)) method, in which case the initialValue method will not be invoked for the thread. Normally, this method is invoked at most once per thread, but it may be invoked again in case of subsequent invocations of [remove()](http://docs.google.com/java/lang/ThreadLocal.html#remove()) followed by [get()](http://docs.google.com/java/lang/ThreadLocal.html#get()).

This implementation simply returns null; if the programmer desires thread-local variables to have an initial value other than null, ThreadLocal must be subclassed, and this method overridden. Typically, an anonymous inner class will be used.

**Returns:**the initial value for this thread-local

### get

public [T](http://docs.google.com/java/lang/ThreadLocal.html) **get**()

Returns the value in the current thread's copy of this thread-local variable. If the variable has no value for the current thread, it is first initialized to the value returned by an invocation of the [initialValue()](http://docs.google.com/java/lang/ThreadLocal.html#initialValue()) method.

**Returns:**the current thread's value of this thread-local

### set

public void **set**([T](http://docs.google.com/java/lang/ThreadLocal.html) value)

Sets the current thread's copy of this thread-local variable to the specified value. Most subclasses will have no need to override this method, relying solely on the [initialValue()](http://docs.google.com/java/lang/ThreadLocal.html#initialValue()) method to set the values of thread-locals.

**Parameters:**value - the value to be stored in the current thread's copy of this thread-local.

### remove

public void **remove**()

Removes the current thread's value for this thread-local variable. If this thread-local variable is subsequently [read](http://docs.google.com/java/lang/ThreadLocal.html#get()) by the current thread, its value will be reinitialized by invoking its [initialValue()](http://docs.google.com/java/lang/ThreadLocal.html#initialValue()) method, unless its value is [set](http://docs.google.com/java/lang/ThreadLocal.html#set(T)) by the current thread in the interim. This may result in multiple invocations of the initialValue method in the current thread.

**Since:** 1.5

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ThreadLocal.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/ThreadGroup.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/Throwable.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/ThreadLocal.html)    [**NO FRAMES**](http://docs.google.com/ThreadLocal.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).