

JAVA.LANG.THREADGROUP.DESTROY METHOD

http://www.tutorialspoint.com/java/lang/threadgroup_destroy.htm

Copyright © tutorialspoint.com

Description

The **java.lang.ThreadGroup.destroy** method destroys this thread group and all of its subgroups. This thread group must be empty, indicating that all threads that had been in this thread group have since stopped.

Declaration

Following is the declaration for **java.lang.ThreadGroup.destroy** method

```
public final void destroy()
```

Parameters

- NA

Return Value

This method does not return any value.

Exception

- **IllegalThreadStateException** -- if the thread group is not empty or if the thread group has already been destroyed.
- **SecurityException** -- if the current thread cannot modify this thread group.

Example

The following example shows the usage of java.lang.ThreadGroup.destroy method.

```
package com.tutorialspoint;

import java.lang.*;

public class ThreadGroupDemo implements Runnable
{
    public static void main(String[] args) {
        ThreadGroupDemo tg = new ThreadGroupDemo();
        tg.func();
    }

    public void func() {
        try {
            // create a parent ThreadGroup
            ThreadGroup pGroup = new ThreadGroup("Parent ThreadGroup");

            // create a child ThreadGroup for parent ThreadGroup
            ThreadGroup cGroup = new ThreadGroup(pGroup, "Child ThreadGroup");

            // create a thread
            Thread t1 = new Thread(pGroup, this);
            System.out.println("Starting " + t1.getName() + "...");
            t1.start();

            // create another thread
            Thread t2 = new Thread(cGroup, this);
            System.out.println("Starting " + t2.getName() + "...");
            t2.start();

            // block until the other threads finish
```

```

        t1.join();
        t2.join();

        // child group destroyed
        cGroup.destroy();
        System.out.println(cGroup.getName() + " destroyed");

        // parent group destroyed
        pGroup.destroy();
        System.out.println(pGroup.getName() + " destroyed");

    }
    catch (InterruptedException ex) {
        System.out.println(ex.toString());
    }
}

// implements run()
public void run() {

    for(int i=0;i<1000;i++) {
        i++;
    }
    System.out.println(Thread.currentThread().getName() +
        " finished executing.");
}
}

```

Let us compile and run the above program, this will produce the following result:

```

Starting Thread-0...
Starting Thread-1...
Thread-0 finished executing.
Thread-1 finished executing.
Child ThreadGroup destroyed
Parent ThreadGroup destroyed

```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js