

JAVA.LANG.THREADGROUP.ACTIVECOUNT METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.ThreadGroup.activeCount** method returns an estimate of the number of active threads in this thread group.

Declaration

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

```
public int activeCount()
```

Parameters

- NA

Return Value

This method returns an estimate of the number of active threads in this thread group and in any other thread group that has this thread group as an ancestor.

Exception

- NA

Example

The following example shows the usage of java.lang.ThreadGroup.activeCount 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();

            // display the number of active threads
            System.out.println("Active threads in \"" + pGroup.getName()
                + "\" = " + pGroup.activeCount());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        // block until the other threads finish
        t1.join();
        t2.join();
    }
    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...
Active threads in "Parent ThreadGroup" = 2
Thread-0 finished executing
Thread-1 finished executing

```

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