

JAVA.LANG.THREADGROUP.SETDAEMON METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.ThreadGroup.setDaemon** method changes the daemon status of this thread group. A daemon thread group is automatically destroyed when its last thread is stopped or its last thread group is destroyed.

Declaration

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

```
public final void setDaemon(boolean daemon)
```

Parameters

- **daemon** -- if true, marks this thread group as a daemon thread group; otherwise, marks this thread group as normal.

Return Value

This method does not return any value.

Exception

- **SecurityException** -- if the current thread cannot modify this thread group.

Example

The following example shows the usage of java.lang.ThreadGroup.setDaemon 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");
            // daemon status is set to true
            pGroup.setDaemon(true);

            // create a child ThreadGroup for parent ThreadGroup
            ThreadGroup cGroup = new ThreadGroup(pGroup, "Child ThreadGroup");
            // daemon status is set to true
            cGroup.setDaemon(true);

            // 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();
        }
    }
}
```

```

        // returns true if this thread group is a daemon thread group
        System.out.println("Is " + pGroup.getName() + " a daemon
        ThreadGroup? " + pGroup.isDaemon());
        System.out.println("Is " + cGroup.getName() + " a daemon
        ThreadGroup? " + cGroup.isDaemon());

        // 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...
Is Parent ThreadGroup a daemonThreadGroup? true
Is Child ThreadGroup a daemonThreadGroup? true
Thread-0 finished executing.
Thread-1 finished executing.

```

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