

# JAVA.LANG.THREAD.ENUMERATE METHOD

[http://www.tutorialspoint.com/java/lang/thread\\_enumerate.htm](http://www.tutorialspoint.com/java/lang/thread_enumerate.htm)

Copyright © tutorialspoint.com

## Description

The **java.lang.Thread.enumerate** method copies into the specified array every active thread in the current thread's thread group and its subgroups. It calls the enumerate method of the current thread's thread group with the **tarray** argument.

## Declaration

Following is the declaration for **java.lang.Thread.enumerate** method

```
public static int enumerate(Thread[] tarray)
```

## Parameters

- **tarray** -- This is an an array of Thread objects to copy to.

## Return Value

This method returns the number of threads put into the array

## Exception

- **SecurityException** -- if a security manager exists and its checkAccess method doesn't allow the operation.

## Example

The following example shows the usage of java.lang.Thread.enumerate method.

```
package com.tutorialspoint;

import java.lang.*;

public class ThreadDemo {

    public static void main(String[] args) {

        Thread t = Thread.currentThread();
        t.setName("Admin Thread");

        // prints the current thread
        System.out.println("Thread = " + t);

        int count = Thread.activeCount();
        System.out.println("currently active threads = " + count);

        Thread th[] = new Thread[count];
        // returns the number of threads put into the array
        Thread.enumerate(th);

        // prints active threads
        for (int i = 0; i < count; i++) {
            System.out.println(i + ": " + th[i]);
        }
    }
}
```

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

```
Thread = Thread[Admin Thread,5,main]
```

```
currently active threads = 1  
0: Thread[Admin Thread 5,main]
```

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