Daemon Thread in Java

Daemon thread in java is a service provider thread that provides services to the user thread. Its life depend on the mercy of user threads i.e. when all the user threads dies, JVM terminates this thread automatically.

There are many java daemon threads running automatically e.g. gc, finalizer etc.

You can see all the detail by typing the jconsole in the command prompt. The jconsole tool provides information about the loaded classes, memory usage, running threads etc.

Points to remember for Daemon Thread in Java

- It provides services to user threads for background supporting tasks. It has no role in life than to serve user threads.
- Its life depends on user threads.
- It is a low priority thread.

Why JVM terminates the daemon thread if there is no user thread?

The sole purpose of the daemon thread is that it provides services to user thread for background supporting task. If there is no user thread, why should JVM keep running this thread. That is why JVM terminates the daemon thread if there is no user thread.

Methods for Java Daemon thread by Thread class

The java.lang.Thread class provides two methods for java daemon thread.

No.	Method	Description
1)	public void setDaemon(boolean status)	is used to mark the current thread as daemon thread or user thread.
2)	public boolean isDaemon()	is used to check that current is daemon.

Simple example of Daemon thread in java

File: MyThread.java

```
public class TestDaemonThread1 extends Thread {
        public void run() {
                if(Thread.currentThread().isDaemon()){
                //checking for daemon thread
                System.out.println("daemon thread work");
                }  else {
                        System.out.println("user thread work");
                        }
        }

 public static void main(String[] args) {

        TestDaemonThread1 t1=new TestDaemonThread1();//creating thread
        TestDaemonThread1 t2=new TestDaemonThread1();
        TestDaemonThread1 t3=new TestDaemonThread1();

        t1.setDaemon(true);//now t1 is daemon thread

        t1.start();//starting threads
        t2.start();
        t3.start();
 }
}
```

Test it Now

Output

```
daemon thread work
user thread work
user thread work
```

Note: If you want to make a user thread as Daemon, it must not be started otherwise it will throw IllegalThreadStateException.

File: MyThread.java

```
class TestDaemonThread2 extends Thread {
        public void run() {
                System.out.println("Name: "+Thread.currentThread().getName());
                System.out.println("Daemon: "+Thread.currentThread().isDaemon());
        }
        public static void main(String[] args) {
                TestDaemonThread2 t1=new TestDaemonThread2();
                TestDaemonThread2 t2=new TestDaemonThread2();

                t1.start();
                t1.setDaemon(true);//will throw exception here
                t2.start();
        }
}
```

Test it Now

```
Output:exception in thread main: java.lang.IllegalThreadStateException
```