java.lang.* Thread



```
Thread
Thread (Runnable target)
Thread (String name)
Thread (ThreadGroup group, Runnable target)
Thread (ThreadGroup group, String name)
Thread (Runnable target, String name)
Thread (ThreadGroup group, Runnable target, String name)
Thread (ThreadGroup group, Runnable target, String name,
       long stackSize)
Static Methods
         int activeCount ()
     Thread currentThread ()
        void dumpStack ()
          int enumerate (Thread tarray[])
    boolean holdsLock (Object obj)
    boolean interrupted ()
        void sleep (long millis) &
        void sleep (long millis, int nanos) &
        void yield ()
ClassLoader get / setContextClassLoader ()
      String get / setName ()
          int get / setPriority ()
ThreadGroup getThreadGroup ()
    boolean isAlive ()
    boolean is / setDaemon ()
    boolean isInterrupted ()
      String toString ()
Other Publ
        void checkAccess ()
        void destroy ()
        void interrupt ()
        void join () 🖔
        void join (long millis) 🕴
        void join (long millis, int nanos) 🖔
        void run ()
        void start ()
int MIN_PRIORITY, NORM_PRIORITY, MAX_PRIORITY
```

```
ThreadGroup
ThreadGroup (String name)
ThreadGroup (ThreadGroup parent, String name)
         int get / setMaxPriority ()
      String getName ()
ThreadGroup getParent ()
    boolean is / setDaemon ()
    boolean isDestroyed ()
      String toString ()
          int activeCount ()
         int activeGroupCount ()
        void checkAccess ()
        void destroy ()
          int enumerate (Thread list[])
          int enumerate (ThreadGroup list[])
         int enumerate (Thread list[], boolean recurse)
         int enumerate (ThreadGroup list[], boolean recurse)
        void interrupt ()
        void list ()
    boolean parentOf (ThreadGroup g)
        void uncaughtException (Thread t, Throwable e)
```

```
java.util. TimerTask

# TimerTask ()

boolean cancel ()
 void run ()
 long scheduledExecutionTime ()
```

```
java.util. Timer

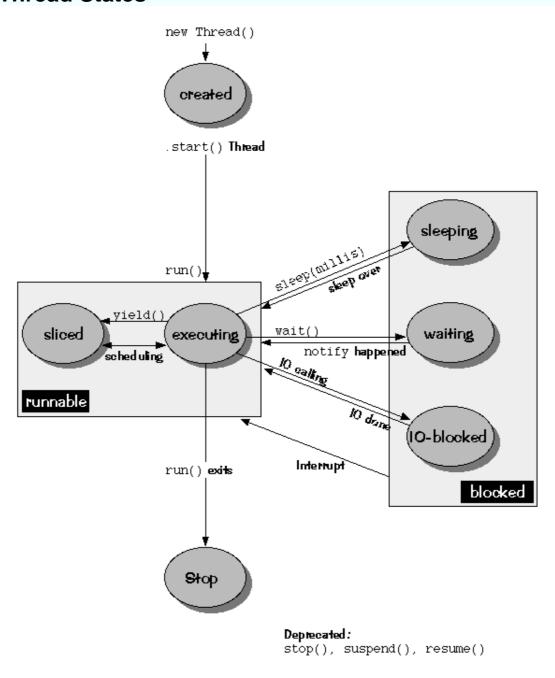
Timer ()
Timer (boolean isDaemon)

void cancel ()
void schedule (TimerTask task, long delay)
void schedule (TimerTask task, Date time)
void schedule (TimerTask task, long delay, long period)
void schedule (TimerTask task, Date firstTime, long period)
void scheduleAtFixedRate (TimerTask task, long delay, long period)
void scheduleAtFixedRate (TimerTask task, Date firstTime, long period)
void scheduleAtFixedRate (TimerTask task, Date firstTime, long period)
```



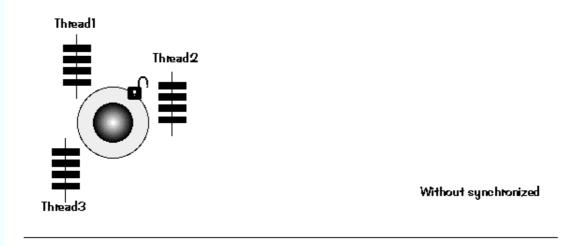


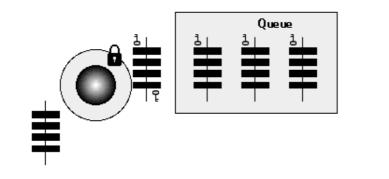
Thread States



www.falkhausen.de Version 0.8 Copyright 2002 by Markus Falkhausen. All rights reserved.

synchronized and wait





 ${\color{red}\mathbb{L}}$ synchronized

