Thread class Property and Methods

Thread class Properties:

In C# Thread class provides multiple properties as

CurrentThread

Returns the instance of currently running thread.

IsAlive

Checks whether the current thread is alive or not. It is used to find the execution status of the thread.

IsBackground

Is used to get or set value whether current thread is in background or not.

ManagedThreadId

Is used to get unique id for the current managed thread.

Name

Is used to get or set the name of the current thread.

Priority

Is used to get or set the priority of the current thread.

ThreadState

Is used to return a value representing the thread state.

Application 1:

Below application is used to demonstrate the concept of Main thread and its thread class Properties.

```
using System;
using System.Threading;

public class Marvellous
{
   public static void Main(string[] args)
   {
      Thread t = Thread.CurrentThread;

      t.Name = "MainThread";
```

```
        Console.WriteLine(t.Name);

        if (t.IsAlive)
        {
            Console.WriteLine("Thread is alive");
        }

        Console.WriteLine("Priority of main thread is {0}", t.Priority);

        Console.WriteLine("Thread state of main thread is {0}", t.ThreadState);

        Console.WriteLine("Thread id of main thread is {0}", t.ManagedThreadId);

        if (t.IsBackground)
        {
            Console.WriteLine("Thread is in background");
        }
    }
}
```

C# Thread Life Cycle

In C#, each thread has a life cycle. The life cycle of a thread is started when instance of System.Threading.Thread class is created. When the task execution of the thread is completed, its life cycle is ended.

There are following states in the life cycle of a Thread in C#.

- Unstarted
- Runnable (Ready to run)
- Running
- Not Runnable
- Dead (Terminated)

Unstarted State

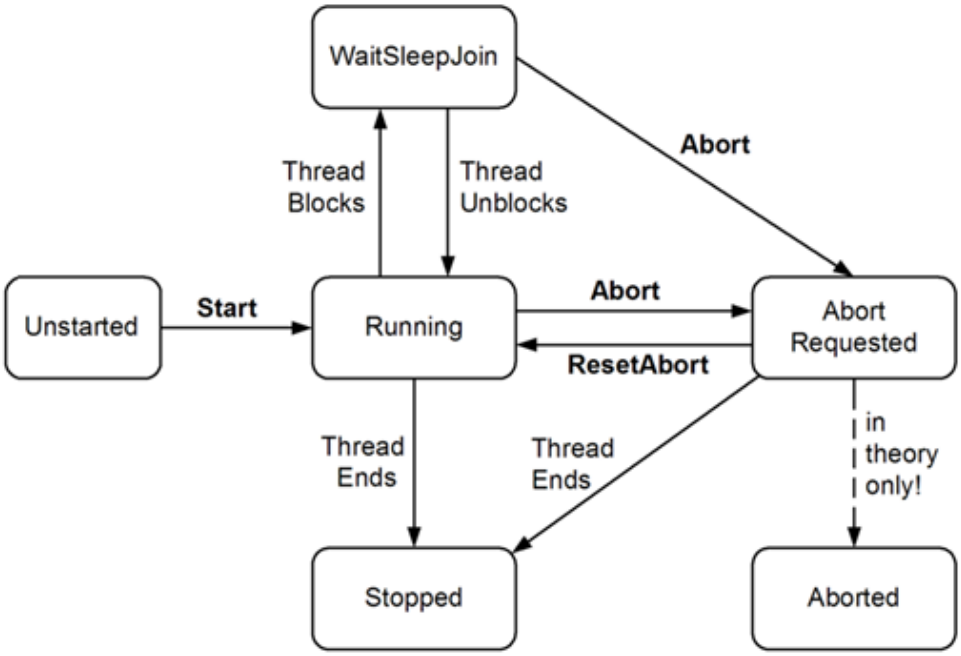
When the instance of Thread class is created, it is in unstarted state by default.

Runnable State

When start() method on the thread is called, it is in runnable or ready to run state.

Running State

Only one thread within a process can be executed at a time. At the time of execution, thread is in running state.



Not Runnable State

The thread is in not runnable state, if sleep() or wait() method is called on the thread, or input/output operation is blocked.

Dead State

After completing the task, thread enters into dead or terminated state.

A list of important methods of Thread class are given below:

Abort()	Is used to terminate the thread. It raises ThreadAbortException.
Interrupt()	Is used to interrupt a thread which is in WaitSleepJoin state.
Join()	Is used to block all the calling threads until this thread terminates.
ResetAbort()	Is used to cancel the Abort request for the current thread.
Resume()	Is used to resume the suspended thread.
Sleep(Int32)	Is used to suspend the current thread for the specified milliseconds.
Start()	Changes the current state of the thread to Runnable.
Suspend()	Suspends the current thread if it is not suspended.
Yield()	Is used to yield the execution of current thread to another thread.