

# Current Thread Properties

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;

class Program {
    static int interval;
    static void Main(string[] args) {
        interval = 100;

        ThreadPool.QueueUserWorkItem(new WaitCallback(StartMethod));
        Thread.Sleep(100);
        ThreadPool.QueueUserWorkItem(new WaitCallback(StartMethod));
        Console.ReadLine();
    }

    static void StartMethod(Object stateInfo) {
        DisplayNumbers("Thread " + stateInfo + " started at " +
            DateTime.Now.Millisecond.ToString());
        Console.WriteLine("Thread Finished");
    }

    static void DisplayNumbers(string GivenThreadName) {
        Console.WriteLine("Starting thread: " + GivenThreadName);

        for (int i = 1; i <= 8 * interval; i++) { if (i % interval ==
            0) { Console.WriteLine("Count has reached " + i);
            Console.WriteLine("CurrentCulture: " + Thread.CurrentThread.CurrentCulture.ToString());
            Console.WriteLine("IsThreadPoolThread: " + Thread.CurrentThread.IsThreadPoolThread.ToString());
            Console.WriteLine("ManagedThreadId: " + Thread.CurrentThread.ManagedThreadId.ToString());
            Console.WriteLine("Priority: " + Thread.CurrentThread.Priority.ToString());
```

```
Thread.CurrentThread.Priority.ToString());  
Console.WriteLine("ThreadState:          "          +  
Thread.CurrentThread.ThreadState.ToString());  
Thread.Sleep(1000); } } } } [/csharp]
```