new Mutex(false), WaitOne

```
using System;
using System.Threading;

class Database
{
static Mutex mutex = new Mutex(false);

public static void SaveData(string text)
{
mutex.WaitOne();
Console.WriteLine(“Database.SaveData – Started”);
Console.WriteLine(“Database.SaveData – Working”);
for (int i = 0; i < 100; i++) { Console.Write(text); }
Console.WriteLine("  Database.SaveData  -  Ended");
mutex.Close(); } } class ThreadMutexApp{ public static void
WorkerThreadMethod1() { Console.WriteLine("Worker thread #1 -
Started"); Database.SaveData("x"); Console.WriteLine("Worker
thread #1 - Returned from Output"); } public static void
WorkerThreadMethod2() { Console.WriteLine("Worker thread #2 -
Started"); Database.SaveData("o"); Console.WriteLine("Worker
thread #2 - Returned from Output"); } public static void
Main()    {    ThreadStart    worker1    =    new
ThreadStart(WorkerThreadMethod1); ThreadStart worker2 = new
ThreadStart(WorkerThreadMethod2); Console.WriteLine("Main -
Creating worker threads"); Thread t1 = new Thread(worker1);
Thread t2 = new Thread(worker2); t1.Start(); t2.Start();
Console.ReadLine(); } } [/csharp]
```