

如果.NET下的一个多线程程序不能正确的处理异常将很有可能会发生由于未处理异常导致进程意外终止的情况，尤其是在使用System.Threading.Timer的时候，由于TimerCallBack是在一个单独的线程中执行的，因此在TimerCallBack方法中发生异常而没有Catch的话将会导致未处理异常是进程意外终止。

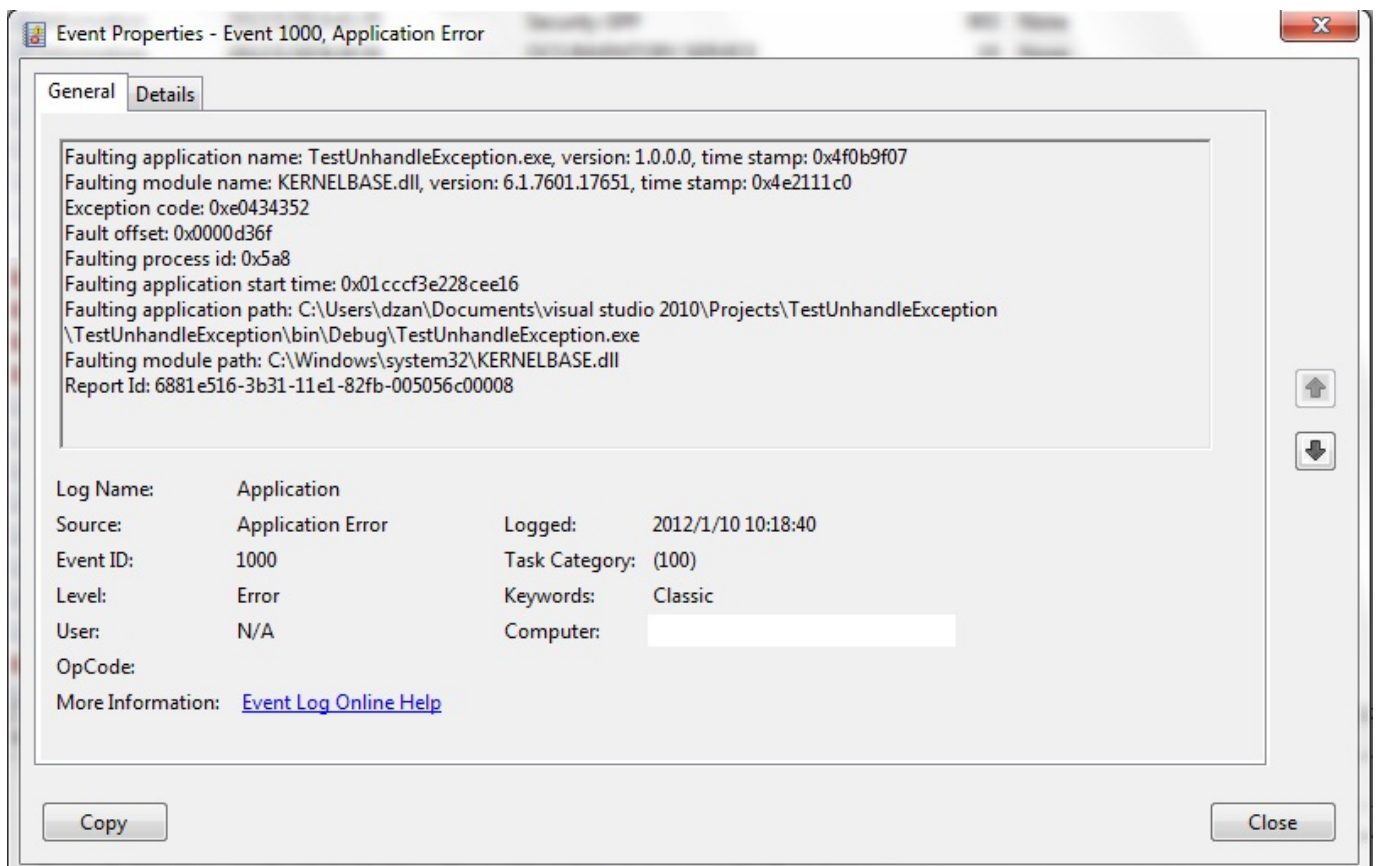
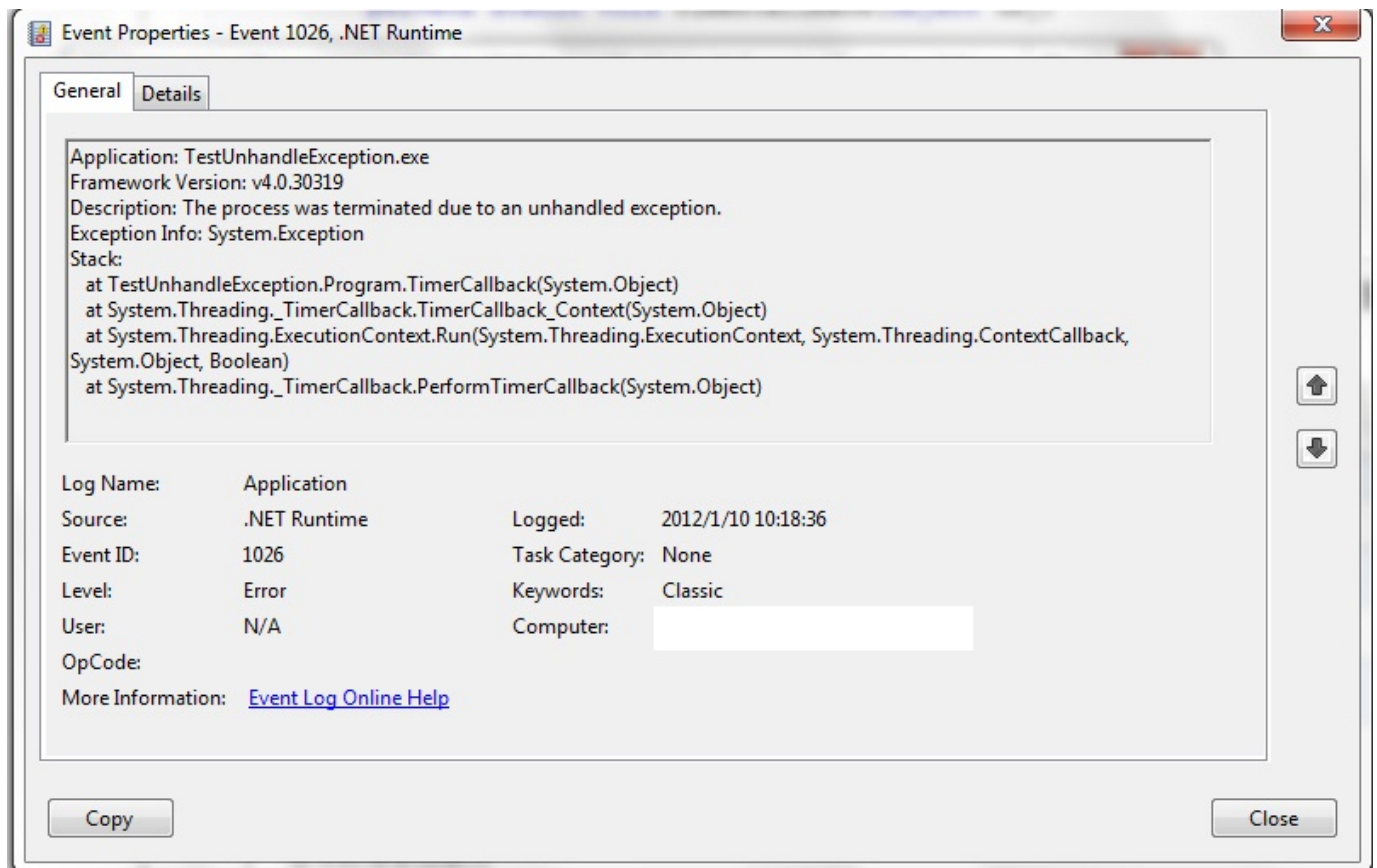
如下的代码所示：

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

namespace TestUnhandleException
{
    class Program
    {
        static void Main(string[] args)
        {
            Timer timer = new Timer(TimerCallback, null, 10000, 10000);
            Console.ReadLine();
        }

        private static void TimerCallback(object obj)
        {
            throw new Exception("Throw a unHandledException for test");
        }
    }
}
```

上面的程序启动后10秒钟便会意外终止，如果查看Windows EventLog的话将会看到下面的两条Log：



正确的做法如下：

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

```
using System.Threading;

namespace TestUnhandleException
{
    class Program
    {
        static void Main(string[] args)
        {
            AppDomain.CurrentDomain.UnhandledException += CurrentDomain_UnhandledException
            Timer timer = new Timer(TimerCallback, null, 10000, 10000);
            Console.ReadLine();
        }

        private static void CurrentDomain_UnhandledException(object sender, UnhandledExcep
        {
            //Todo:Log
        }

        private static void TimerCallback(object obj)
        {
            try
            {
                throw new Exception("Throw a unHandledException for test");
            }
            catch(Exception)
            {
                //Todo:Log
            }
        }
    }
}
```