

The Source Code without extraneous bits

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-04/msg01855.html>

- *From:* "David Beoulve" <egyptianmaadi@xxxxxxxx>
 - *Date:* 12 Apr 2006 05:57:39 -0700
-

Sure thing, Jon. I love your explanation page -- but it's much too wordy. I skimmed it and found that my assumptions before clicking your link were right -- you wanted a compilable program that didn't have extraneous logic.

Here it goes: (this compiles)

"Form1.cs"

```
using System;
using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Threading;

namespace Navigation_Creator_2
{
    public delegate void doUpdateUIthreadCounter();
    public partial class Form1 : Form
    {
        private Thread[] oThreads = new Thread[5000];
        private int iThreadIndex = 0;
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void buttonRunProgram_Click(object sender, EventArgs e)
        {
            // Create a new thread to separate it from the UI

```

The Source Code without extraneous bits

```
Thread tMain = new Thread(new
ThreadStart(doFoldersThread));
tMain.Name = "Main Thread";
tMain.IsBackground = true;
tMain.Start();

// add it to our watch list of threads.
addThreadToThreadWatch(tMain, tMain.Name);

Thread tWatcher = new Thread(new
ThreadStart(awakenTheWatcher));
tWatcher.Name = "The Watcher";
tWatcher.IsBackground = true;
tWatcher.Start();

/* disable the RUN button – we don't want threads trying
* to access the same files */
buttonRunProgram.Enabled = false;
}
private void addThreadToThreadWatch(Thread t, string
strThreadName)
{
lock (oThreads)
{
if (t.IsAlive)
{
// add this thread to our sta
```