

Support the Cancellation of a Background Worker Thread



Start

```
//File:Window.xaml.cs

using System;
using System.ComponentModel;
using System.Windows;

namespace WpfApplication1
{
    public partial class Window1 : Window
    {
        private BackgroundWorker worker = new BackgroundWorker();

        private long from = 2;
        private long to = 2000;
        private long biggestPrime;

        public Window1(): base()
        {
            InitializeComponent();
            worker.WorkerSupportsCancellation = true;
            worker.DoWork += new DoWorkEventHandler(worker_DoWork);
            worker.RunWorkerCompleted += new RunWorkerCompletedEventHandler(worker_RunWorkerCompleted);
        }
        private void StartStop_Click(object sender, RoutedEventArgs e)
        {
```

```

if(!worker.IsBusy)
{
worker.RunWorkerAsync();
btnStartStop.Content = "Cancel";
txtBiggestPrime.Text = string.Empty;
}
else
{
worker.CancelAsync();
}
}

private void worker_RunWorkerCompleted(object sender,
RunWorkerCompletedEventArgs e)
{
if(e.Cancelled)
{
MessageBox.Show("Operation was canceled");
}

btnStartStop.Content = "Start";
txtBiggestPrime.Text = biggestPrime.ToString();
}

private void worker_DoWork(object sender, DoWorkEventArgs e)
{
for(long current = from; current <= to; current++) {
if(worker.CancellationPending) { e.Cancel = true; return; }
biggestPrime = current; } } } } [/csharp]

```