Execute a Method Asynchronously Using a Background Worker Thread

Start

```
//File:Window.xaml.cs
using System;
using System.Windows;
using System.ComponentModel;

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

private long from= 1;
private long to = 200;
private long biggestPrime;

public Window1(): base()
{
InitializeComponent();
worker.DoWork += new DoWorkEventHandler(worker_DoWork);
worker.RunWorkerCompleted += new
RunWorkerCompletedEventHandler(worker_RunWorkerCompleted);
}

private void Start_Click(object sender, RoutedEventArgs e)
```

```
{
worker.RunWorkerAsync();
btnStart.IsEnabled = false;
txtBiggestPrime.Text = string.Empty;
}

private void worker_RunWorkerCompleted(object sender,
RunWorkerCompletedEventArgs e)
{
btnStart.IsEnabled = true;
txtBiggestPrime.Text = biggestPrime.ToString();
}

private void worker_DoWork(object sender, DoWorkEventArgs e)
{
for(long current = from; current <= to; current++) {
biggestPrime = current; } } } } [/csharp]
```