

Create a Background Worker Thread in XAML



Start

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

using System;
using System.ComponentModel;
using System.Threading;
using System.Windows;
using System.Windows.Input;

namespace WpfApplication1
{
    public partial class Window1 : Window
    {
        private readonly BackgroundWorker worker;

        public Window1()
        {
            InitializeComponent();
            worker = this.FindResource("backgroundWorker") as
            BackgroundWorker;
        }

        private void button_Click(object sender, RoutedEventArgs e)
        {
```

```
if(!worker.IsBusy)
{
this.Cursor = Cursors.Wait;
worker.RunWorkerAsync();
button.Content = "Cancel";
}else{
worker.CancelAsync();
}
}
```

```
private void BackgroundWorker_DoWork(object sender,
System.ComponentModel.DoWorkEventArgs e)
{
for(int i = 1; i <= 100; i++) { if(worker.CancellationPending)
break; Thread.Sleep(100); worker.ReportProgress(i); } }
private void BackgroundWorker_RunWorkerCompleted(object
sender, System.ComponentModel.RunWorkerCompletedEventArgs e) {
this.Cursor = Cursors.Arrow;
Console.WriteLine(e.Error.Message); button.Content = "Start";
} private void BackgroundWorker_ProgressChanged(object sender,
System.ComponentModel.ProgressChangedEventArgs e) {
progressBar.Value = e.ProgressPercentage; } } } [/csharp]
```