

Show a Continuous Animation During an Asynchronous Process



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

```
using System.Windows;
```

```

using System.Threading;
using System.ComponentModel;
using System.Windows.Media.Animation;

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

public Window1()
{
InitializeComponent();

pulseStoryboard = (Storyboard)
this.Resources["PulseStoryboard"];
pulseStoryboard.RepeatBehavior = RepeatBehavior.Forever;
worker = new BackgroundWorker();

worker.DoWork += new DoWorkEventHandler(worker_DoWork);
worker.RunWorkerCompleted += new
RunWorkerCompletedEventHandler(worker_RunWorkerCompleted);
}

private void button_Click(object sender, RoutedEventArgs e)
{
pulseStoryboard.Begin(this, true);
worker.RunWorkerAsync();
button.IsEnabled = false;
}

private void worker_RunWorkerCompleted(object sender,
RunWorkerCompletedEventArgs e)
{
button.IsEnabled = true;
pulseStoryboard.Stop(this);
}
}

```

```
private void worker_DoWork(object sender, DoWorkEventArgs e)
{
for(int i = 1; i <= 50; i++) { Thread.Sleep(100); } } } }
[/csharp]
```