```csharp
using System;
using System.Drawing;
using MonoTouch.Foundation;
using MonoTouch.UIKit;
using System.Threading;

namespace ControlDemo_LMT
{
    public partial class ActivityIndicatorView : UIViewController
    {

        public ActivityIndicatorView () : base ("ActivityIndicatorView", null)
        {

        }

        UIActivityIndicatorView _actView;

        public override void DidReceiveMemoryWarning ()
        {
            // Releases the view if it doesn't have a superview.
            base.DidReceiveMemoryWarning ();

            // Release any cached data, images, etc that aren't in use.
        }

        public override void ViewDidLoad ()
        {
            base.ViewDidLoad ();
            View.BackgroundColor = UIColor.Green;
            showActivityBtn.TouchUpInside += HandleShowActivityTouchUpInside;
        }

        void HandleShowActivityTouchUpInside (object sender, EventArgs e) // auto-
created method to handle the TouchUpInside event on the Btn
        {
            _actView = new UIActivityIndicatorView ();
            _actView.Frame = new RectangleF (0, 0, 50, 50);
            _actView.Center = View.Center;
            _actView.ActivityIndicatorViewStyle = UIActivityIndicatorViewStyle.WhiteLarg
e;
            _actView.BackgroundColor = UIColor.Blue;
            View.AddSubview (_actView);
            _actView.StartAnimating (); // start the animation

            Thread t = new Thread (DoSomething); // start the loading task on another thread
otherwise you would not see the activity indicator
            t.Start (); // start that thread
```

```
void DoSomething()
{
    Thread.Sleep(5000); // sleep the thread for 5 seconds
    //after thread awakes, use an autorealeasepool to release the thread and on the main thread, stop the activity indicator animation.
    using (var pool = new NSAutoreleasePool()){
        this.InvokeOnMainThread(delegate {
            _actView.StopAnimating();
        });
    }
}
```