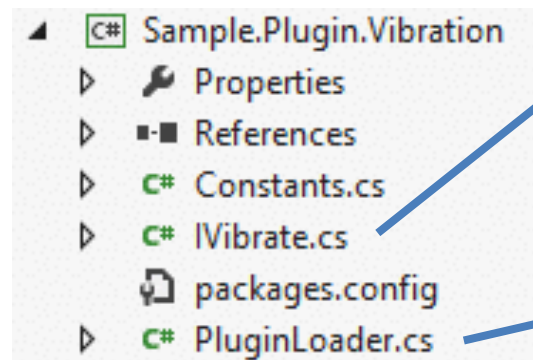


What is a plugin?

- Plugins are just a layer on top of IoC
- Each Plugin has:
 - A Portable Core
 - Includes the interface definition
 - Has PluginLoader.Instance entry point
 - (optional) Platform extensions
 - Extensions identified by naming conventions
 - Each has a Plugin entry point

A Vibrate plugin

Core - PCL:



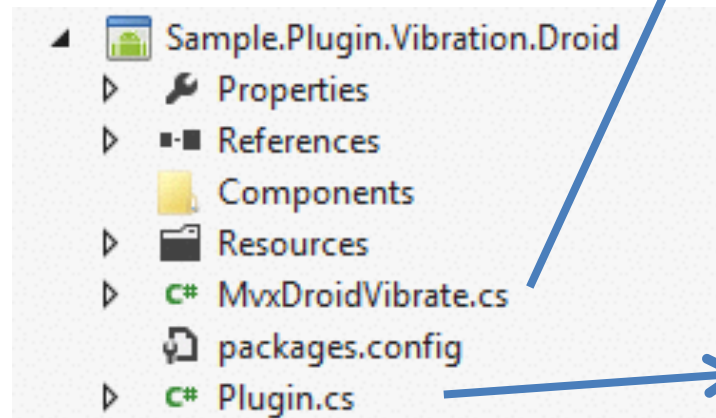
```
public interface IVibrate
{
    void Shake();
}
```

```
public class PluginLoader
    : IMvxPluginLoader
{
    public static readonly PluginLoader Instance = new PluginLoader();

    public void EnsureLoaded()
    {
        var manager = Mvx.Resolve<IMvxPluginManager>();
        manager.EnsurePlatformAdaptionLoaded<PluginLoader>();
    }
}
```

A Vibrate plugin

e.g. Droid Adaption:



```
public class MvxDroidVibrate : IVibrate
{
    public void Shake()
    {
        var globals = Mvx.Resolve<Cirrious.CrossCore.Droid.IMvxAndroidGlobals>();
        var vibrator = globals.ApplicationContext
            .GetSystemService(Context.VibratorService)
            as Vibrator;
        vibrator.Vibrate((long)Constants.Short.TotalMilliseconds);
    }
}

public class Plugin
    : IMvxPlugin
{
    public void Load()
    {
        Mvx.RegisterSingleton<IVibrate>(new MvxDroidVibrate());
    }
}
```

General Plugins

- Plugins are just a wrapper around IoC
- NuGet makes them very easy
- NuGet also allows inter-Plugin dependencies
- You can have many interfaces in 1 Plugin