Loxodon Framework Fody

(中文版)

Developed by Clark

Requires Unity 2018.4 or higher.

This is a plugin for static weaving code that integrates Fody into a Unity project.

PropertyChanged.Fody：Injects code which raises the PropertyChanged event, into property setters of classes which implement INotifyPropertyChanged.

ToString.Fody：Generates ToString method from public properties for class decorated with a [ToString] Attribute.

BindingProxy.Fody Statically weave proxy classes for fields, properties, and methods for ViewModel, use statically injected proxy classes to access object properties and methods during data binding, and optimize call performance.

Installation

Install via OpenUPM (recommended)

OpenUPM can automatically manage dependencies, it is recommended to use it to install the framework.

Requires nodejs's npm and openupm-cli, if not installed please install them first.

```
# Install openupm-cli,please ignore if it is already installed.
npm install -g openupm-cli

#Go to the root directory of your project
cd F:/workspace/New Unity Project

#Install loxodon-framework-fody-propertychanged
openupm add com.vovgou.loxodon-framework-fody-propertychanged

#Install loxodon-framework-fody-tostring
openupm add com.vovgou.loxodon-framework-fody-tostring

#Install loxodon-framework-fody-bindingproxy
openupm add com.vovgou.loxodon-framework-fody-bindingproxy
```

Install via Packages/manifest.json

Modify the Packages/manifest.json file in your project, add the third-party repository "package.openupm.com"'s configuration and add "com.vovgou.loxodon-framework-fody" in the "dependencies" node.

Installing the framework in this way does not require nodejs and openm-cli.

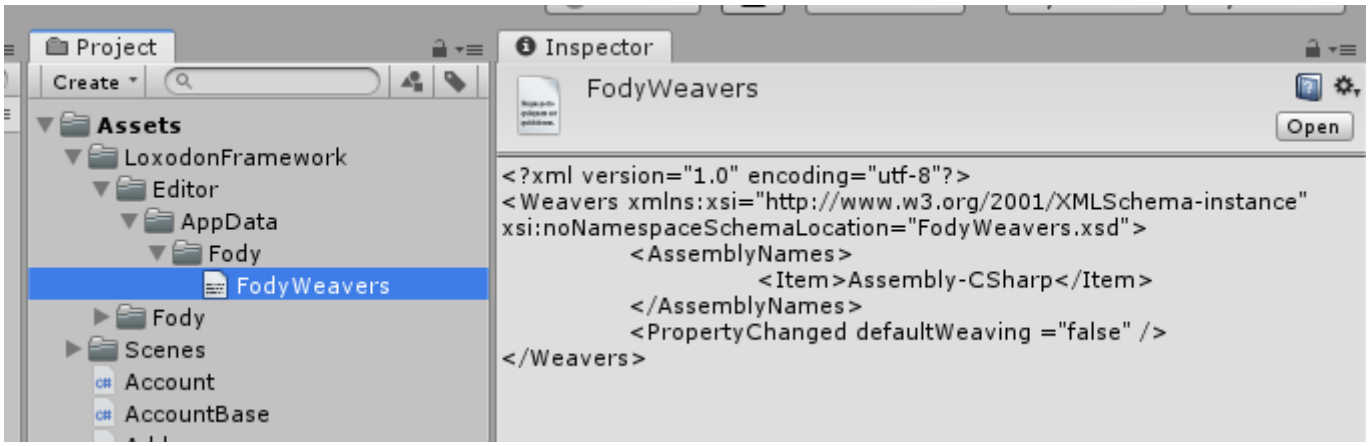
```
{
  "dependencies": {
    ...
    "com.vovgou.loxodon-framework-fody": "2.6.0",
    "com.vovgou.loxodon-framework-fody-propertychanged": "2.6.0",
    "com.vovgou.loxodon-framework-fody-tostring": "2.6.0",
        "com.vovgou.loxodon-framework-fody-bindingproxy": "2.6.0"
  },
  "scopedRegistries": [
    {
      "name": "package.openupm.com",
      "url": "https://package.openupm.com",
      "scopes": [
        "com.vovgou",
        "com.openupm"
      ]
    }
  ]
}
```

Quick Start

PropertyChanged.Fody

After the plugin is imported into the project, the FodyWeavers.xml file will be automatically generated in the Assets\LoxodonFramework\Editor\AppData\Fody directory. Add the names of the assemblies that need to weave the notification code into this configuration file. The PropertyChanged node in the XML file is about the configuration of the PropertyChanged.Fody plug-in. For details, please refer to the document of PropertyChanged.Fody.

PropertyChanged.Fody will weave notification events into all classes that inherit INotifyPropertyChanged or add the AddINotifyPropertyChangedInterface annotation by default. If a class does not want to be weaved into the code, you can use the [DoNotNotify] annotation to exclude it.If you only want to weave notification codes for classes annotated with [AddINotifyPropertyChangedInterface], then you can set the "defaultWeaving" property to false in the FodyWeavers.xml file.



FodyWeavers.xml

```
<Weavers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="Fc
        <AssemblyNames>
                <Item>Assembly-CSharp</Item>
        </AssemblyNames>
        <PropertyChanged defaultWeaving ="true" />
</Weavers>
```

Create a User class in the project and add annotation "AddINotifyPropertyChangedInterface" to the class, as follows:

```
[AddINotifyPropertyChangedInterface]
public class User
{
    public string FirstName { get; set; }

    public string LastName { get; set; }

    public string FullName => $"{FirstName} {LastName}";
}
```

After the project is compiled, use the ILSpy tool to view the assembly.the code of the User class is as follows, "PropertyChanged.Fody" weaves the INotifyPropertyChanged interface and related code for User.cs

```
public class User : INotifyPropertyChanged
{
        public string FirstName
        {
                [CompilerGenerated]
                get
                {
                        return FirstName;
                }
                [CompilerGenerated]
                set
                {
                        if (!string.Equals(FirstName, value, StringComparison.Ordinal))
                        {
                                FirstName = value;
                                <>OnPropertyChanged(<>PropertyChangedEventArgs.FullName);
                                <>OnPropertyChanged(<>PropertyChangedEventArgs.FirstName);
                        }
                }
        }

        public string LastName
        {
                [CompilerGenerated]
                get
                {
                        return LastName;
                }
                [CompilerGenerated]
                set
                {
                        if (!string.Equals(LastName, value, StringComparison.Ordinal))
                        {
                                LastName = value;
                                <>OnPropertyChanged(<>PropertyChangedEventArgs.FullName);
                                <>OnPropertyChanged(<>PropertyChangedEventArgs.LastName);
                        }
                }
        }

        public string FullName => FirstName + " " + LastName;

        [field: NonSerialized]
        public event PropertyChangedEventHandler PropertyChanged;

        [GeneratedCode("PropertyChanged.Fody", "3.4.1.0")]
        [DebuggerNonUserCode]
        protected void <>OnPropertyChanged(PropertyChangedEventArgs eventArgs)
        {
                this.PropertyChanged?.Invoke(this, eventArgs);
```

```
        }
}
```

ToString.Fody

FodyWeavers.xml

```
<Weavers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="Fc
        <AssemblyNames>
                <Item>Assembly-CSharp</Item>
        </AssemblyNames>
        <ToString />
</Weavers>
```

The code of the User class is as follows, and the [ToString] annotation is added to the User class.

```
[AddINotifyPropertyChangedInterface]
[ToString]
public class User
{
    public string FirstName { get; set; }

    public string LastName { get; set; }

    public string FullName => $"{FirstName} {LastName}";
}
```

The code after weaving the ToString method is as follows:

```
public string FirstName
{
        [CompilerGenerated]
        get
        {
                return FirstName;
        }
        [CompilerGenerated]
        set
        {
                if (!string.Equals(FirstName, value, StringComparison.Ordinal))
                {
                        FirstName = value;
                        <>OnPropertyChanged(<>PropertyChangedEventArgs.FullName);
                        <>OnPropertyChanged(<>PropertyChangedEventArgs.FirstName);
                }
        }
}

public string LastName
{
        [CompilerGenerated]
        get
        {
                return LastName;
        }
        [CompilerGenerated]
        set
        {
                if (!string.Equals(LastName, value, StringComparison.Ordinal))
                {
                        LastName = value;
                        <>OnPropertyChanged(<>PropertyChangedEventArgs.FullName);
                        <>OnPropertyChanged(<>PropertyChangedEventArgs.LastName);
                }
        }
}

public string FullName => FirstName + " " + LastName;

[field: NonSerialized]
public event PropertyChangedEventHandler PropertyChanged;

[GeneratedCode("PropertyChanged.Fody", "4.0.2.0")]
[DebuggerNonUserCode]
protected void <>OnPropertyChanged(PropertyChangedEventArgs eventArgs)
{
        this.PropertyChanged?.Invoke(this, eventArgs);
}
```

```
[GeneratedCode("Fody.ToString", "1.11.1.0")]
[DebuggerNonUserCode]
public override string ToString()
{
        return string.Format(CultureInfo.InvariantCulture, "{{T: \"User\", FirstName: \"{0}\", L
        {
                FirstName ?? "null",
                LastName ?? "null",
                FullName ?? "null"
        });
}
```

BindingProxy.Fody

BindingProxy.Fody is a performance optimization component, which can statically weave binding proxy classes for the Property, Field and Method of a class, and directly access the Field, Property and Method of the class without reflection. This plugin will only scan all classes that inherit the INotifyPropertyChanged interface. By default, it will not generate binding proxy objects (PropertyNodeProxy, FieldNodeProxy) for the Property and Field of the class, But you can change the default values by modifying the "defaultWeaveProperty" and "defaultWeaveField" in the FodyWeavers.xml file to true.If you need to weave a proxy class for the method of a class, you must explicitly add the [GenerateMethodProxy] annotation to the method.

FodyWeavers.xml

```
<Weavers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="Fo
        <AssemblyNames>
                <Item>Assembly-CSharp</Item>
        </AssemblyNames>
        <BindingProxy defaultWeaveProperty ="false"  defaultWeaveField="false" Debug="true"/>
</Weavers>
```

Attribute

- GenerateFieldProxyAttribute
  If marked on the class, FieldNodeProxy will be generated for all public fields of this class. Marked on Field, it only means that this field needs to generate FieldNodeProxy.
- GeneratePropertyProxyAttribute
  If marked on the class, PropertyNodeProxy will be generated for all public properties of this class. Marked on Property, it only means that this property needs to generate PropertyNodeProxy.
- GenerateMethodProxyAttribute
  It can only be marked on the method, which means that this method needs to generate MethodNodeProxy.

- IgnoreAttribute
  It can be marked on classes, properties, fields, and methods, it means ignore generating proxy classes.

how to use

After the binding service BindingServiceBundle starts, register the WovenNodeProxyFactory to INodeProxyFactoryRegister.

```
ApplicationContext context = Context.GetApplicationContext();
IServiceContainer container = context.GetContainer();

/* Initialize the data binding service */
BindingServiceBundle bundle = new BindingServiceBundle(context.GetContainer());
bundle.Start();

INodeProxyFactoryRegister nodeFactoryRegister = container.Resolve<INodeProxyFactoryRegister>();
nodeFactoryRegister.Register(new WovenNodeProxyFactory(),100);//Set the priority to 100
```

sample code：

```
[GenerateFieldProxy]
[GeneratePropertyProxy]
[AddINotifyPropertyChangedInterface]
public class AccountViewModel : INotifyPropertyChanged
{

    public event PropertyChangedEventHandler PropertyChanged;

    public string Mobile;

    public string FirstName { get; set; }

    public string LastName { get; protected set; }

    public string FullName => $"{FirstName} {LastName}";

    [Ignore]
    public int Age { get; set; }

    [GenerateMethodProxy]
    public void OnValueChanged()
    {
    }

    [GenerateMethodProxy]
    public void OnValueChanged(int value)
    {
    }
}
```

BindingProxy.Fody will automatically weave proxy classes for public fields, properties, and methods marked with [GenerateMethodProxy] after Unity compiles, such as the FirstName property. This plugin will automatically generate a nest class named FirstNamePropertyNodeProxy in the AccountViewModel class. At the same time, the IWovenNodeProxyFinder interface will be woven into the AccountViewModel class.

```
public class AccountViewModel : INotifyPropertyChanged, IWovenNodeProxyFinder
{
        [GeneratedCode("BindingProxy.Fody", "1.0.0.0")]
        [DebuggerNonUserCode]
        [Preserve]
        private class MobileFieldNodeProxy : WovenFieldNodeProxy<AccountViewModel, string>
        {
                public MobileFieldNodeProxy(AccountViewModel source)
                        : base(source)
                {
                }

                public override string GetValue()
                {
                        return source.Mobile;
                }

                public override void SetValue(string value)
                {
                        source.Mobile = value;
                }
        }

        [GeneratedCode("BindingProxy.Fody", "1.0.0.0")]
        [DebuggerNonUserCode]
        [Preserve]
        private class FirstNamePropertyNodeProxy : WovenPropertyNodeProxy<AccountViewModel, stri
        {
                public FirstNamePropertyNodeProxy(AccountViewModel source)
                        : base(source)
                {
                }

                public override string GetValue()
                {
                        return source.FirstName;
                }

                public override void SetValue(string value)
                {
                        source.FirstName = value;
                }
        }

        [GeneratedCode("BindingProxy.Fody", "1.0.0.0")]
        [DebuggerNonUserCode]
        [Preserve]
        private class LastNamePropertyNodeProxy : WovenPropertyNodeProxy<AccountViewModel, strir
        {
                public LastNamePropertyNodeProxy(AccountViewModel source)
```

```
                : base(source)
        {
        }

        public override string GetValue()
        {
                return source.LastName;
        }

        public override void SetValue(string value)
        {
                throw new MemberAccessException("AccountViewModel.LastName is read-only
        }
}

[GeneratedCode("BindingProxy.Fody", "1.0.0.0")]
[DebuggerNonUserCode]
[Preserve]
private class FullNamePropertyNodeProxy : WovenPropertyNodeProxy<AccountViewModel, strir
{
        public FullNamePropertyNodeProxy(AccountViewModel source)
                : base(source)
        {
        }

        public override string GetValue()
        {
                return source.FullName;
        }

        public override void SetValue(string value)
        {
                throw new MemberAccessException("AccountViewModel.FullName is read-only
        }
}

[GeneratedCode("BindingProxy.Fody", "1.0.0.0")]
[DebuggerNonUserCode]
[Preserve]
private class OnValueChangedMethodNodeProxy : WovenMethodNodeProxy<AccountViewModel>, II
{
        public OnValueChangedMethodNodeProxy(AccountViewModel source)
                : base(source)
        {
        }

        public object Invoke()
        {
                source.OnValueChanged();
                return null;
        }
```

```
        public object Invoke(int value)
        {
                source.OnValueChanged(value);
                return null;
        }

        public override object Invoke(params object[] args)
        {
                switch ((args != null) ? args.Length : 0)
                {
                case 0:
                        return Invoke();
                case 1:
                        return Invoke((int)args[0]);
                default:
                        return null;
                }
        }
}

public string Mobile;

[GeneratedCode("BindingProxy.Fody", "1.0.0.0")]
private WovenNodeProxyFinder _finder;

public string FirstName
{
        [CompilerGenerated]
        get
        {
                return FirstName;
        }
        [CompilerGenerated]
        set
        {
                if (!string.Equals(FirstName, value, StringComparison.Ordinal))
                {
                        FirstName = value;
                        <>OnPropertyChanged(<>PropertyChangedEventArgs.FullName);
                        <>OnPropertyChanged(<>PropertyChangedEventArgs.FirstName);
                }
        }
}

public string LastName
{
        [CompilerGenerated]
        get
        {
                return LastName;
```

```
        }
        [CompilerGenerated]
        protected set
        {
                if (!string.Equals(LastName, value, StringComparison.Ordinal))
                {
                        LastName = value;
                        <>OnPropertyChanged(<>PropertyChangedEventArgs.FullName);
                        <>OnPropertyChanged(<>PropertyChangedEventArgs.LastName);
                }
        }
}

public string FullName => FirstName + " " + LastName;

public int Age
{
        [CompilerGenerated]
        get
        {
                return Age;
        }
        [CompilerGenerated]
        set
        {
                if (Age != value)
                {
                        Age = value;
                        <>OnPropertyChanged(<>PropertyChangedEventArgs.Age);
                }
        }
}

public event PropertyChangedEventHandler PropertyChanged;

public void OnValueChanged()
{
}

public void OnValueChanged(int value)
{
}

[GeneratedCode("PropertyChanged.Fody", "3.4.1.0")]
[DebuggerNonUserCode]
protected void <>OnPropertyChanged(PropertyChangedEventArgs eventArgs)
{
        this.PropertyChanged?.Invoke(this, eventArgs);
}

[GeneratedCode("BindingProxy.Fody", "1.0.0.0")]
```

```
        [DebuggerNonUserCode]
        ISourceProxy IWovenNodeProxyFinder.GetSourceProxy(string name)
        {
                if (_finder == null)
                {
                        _finder = new WovenNodeProxyFinder(this);
                }
                return _finder.GetSourceProxy(name);
        }
}
```

Contact Us

Email: yangpc.china@gmail.com

Website: https://vovgou.github.io/loxodon-framework/

QQ Group: 622321589