

| | |
|----------|--|
| عنوان: | معرفی پروژه NotifyPropertyWeaver |
| نویسنده: | وحید نصیری |
| تاریخ: | ۹:۱۲ ۱۳۹۲/۰۱/۲۲ |
| آدرس: | www.dotnettips.info |
| گروه‌ها: | C#, Design patterns, AOP, Architecture |

پس از معرفی مباحث IL Code Weaving و همچنین ارائه راه حلی در مورد «استفاده از AOP Interceptors برای حذف کدهای تکراری INotifyPropertyChanged در WPF» راه حل مشابهی به نام NotifyPropertyWeaver ارائه شده است که همان کار AOP Interceptors را انجام می‌دهد؛ اما بدون نیاز به تشکیل پروکسی و سربار اضافی. کار نهایی را توسط ویرایش اسمبلی و افزودن کدهای IL لازم انجام می‌دهد؛ البته بدون استفاده از PostSharp. این پروژه از کتابخانه سورس باز پایه‌ای به نام [Fody](#) استفاده می‌کند که جهت IL Code weaving طراحی شده است. اگر به [Wiki](#) آن مراجعه نمائید، لیست افزونه‌های قابل توجهی را در مورد آن خواهید یافت که PropertyChanged تنها یکی از آنها است.

پیشنیازها

الف) [صفحه پروژه در GitHub](#)

ب) [دریافت از طریق نوگت](#)

روش استفاده

پس از نصب بسته نوگت پروژه PropertyChanged.Fody

```
PM> Install-Package PropertyChanged.Fody
```

کلاسی را که باید پس از کامپایل، پیاده سازی‌های خودکار OnPropertyChanged را شامل شود، با ویژگی ImplementPropertyChanged مزین کنید.

```
using PropertyChanged;

namespace AOP02
{
    [ImplementPropertyChanged]
    public class Person
    {
        public string Id { set; get; }
        public string Name { set; get; }
    }
}
```

و سپس پروژه را کامپایل نمائید. خروجی کنسول Build در VS.NET :

```
----- Build started: Project: AOP02, Configuration: Debug x86 -----
Fody (version 1.13.6.1) Executing
Finished Fody 287ms.
AOP02 -> D:\Prog\AOP02\bin\Debug\AOP02.exe
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
```

اکنون اگر فایل اسمبلی نهایی پروژه را در برنامه ILSpy باز کنیم، چنین پیاده سازی را می‌توان شاهد بود:

```
using System;
using System.ComponentModel;
using System.Runtime.CompilerServices;
namespace AOP02
{
    public class Person : INotifyPropertyChanged
    {
        public event PropertyChangedEventHandler PropertyChanged;
        public string Id
```

```
{
    [System.Runtime.CompilerServices.CompilerGenerated]
    get
    {
        return this.<Id>k__BackingField;
    }
    [System.Runtime.CompilerServices.CompilerGenerated]
    set
    {
        if (string.Equals(this.<Id>k__BackingField, value, System.StringComparison.Ordinal))
        {
            return;
        }
        this.<Id>k__BackingField = value;
        this.OnPropertyChanged("Id");
    }
}
public string Name
{
    [System.Runtime.CompilerServices.CompilerGenerated]
    get
    {
        return this.<Name>k__BackingField;
    }
    [System.Runtime.CompilerServices.CompilerGenerated]
    set
    {
        if (string.Equals(this.<Name>k__BackingField, value, System.StringComparison.Ordinal))
        {
            return;
        }
        this.<Name>k__BackingField = value;
        this.OnPropertyChanged("Name");
    }
}
public virtual void OnPropertyChanged(string propertyName)
{
    PropertyChangedEventHandler propertyChanged = this.PropertyChanged;
    if (propertyChanged != null)
    {
        propertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}
}
```

نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۱۰:۱۶ ۱۳۹۲/۰۱/۲۲

تشکر فراوان.

برای پیاده سازی زیر از NotifyPropertyWeaver می توان استفاده کرد ؟

```
public Project ActiveProject
{
    get { return _activeProject; }
    set
    {
        _activeProject = value;
        RaisePropertyChanged(() => ActiveProject);
        RaisePropertyChanged(() => HasProject);
    }
}

public bool HasProject
{
    get { return ActiveProject != null; }
}
```

چون تغییر HasProject به ActiveProject وابسته است قصد داریم هر زمان ActiveProject عوض می شود RaisePropertyChanged برای HasProject هم فراخوانی شود.
پ ن : آقای نصیری تلفظ صحیحش نوگت هست یا نیوگت؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۴۹ ۱۳۹۲/۰۱/۲۲

- بله. یک سری ویژگی دیگر هم دارد به نام های [AlsoNotifyFor](#) و DoNotNotify.
- بله! [نیوگت](#) هست.

نویسنده: ایمان محمدی
تاریخ: ۱۳:۲۴ ۱۳۹۲/۰۳/۱۵

حیفم اومد نظر ندم دستتون درد نکنه فوق العاده عالی بود.
دو تا نکته جالبی که من دیدم:
اگه بدنه set رو تعریف کرده باشید ولی OnPropertyChanged نداشته باشه باز هم به بدنه ست OnPropertyChanged اضافه میکنه.
اگه پروپرتی های دیگه ای تو متد get خودشون از این پروپرتی استفاده کرده باشند برای هر کدوم از اونها هم OnPropertyChanged اضافه می کنه.

نویسنده: سوین
تاریخ: ۱۱:۴۳ ۱۳۹۲/۰۹/۰۸

سلام

من هیچ Package ای رو نمی تونم با Nuget نصب کنم و این خطا رو می ده
Install-Package : The underlying connection was closed: An unexpected error occurred on a send

اما کانکشنم close نیست ممنون میشم راهنمایی کنید .

نویسنده: وحید نصیری

تاریخ: ۱۳۹۲/۰۹/۰۸ ۱۲:۲۰

- فید NuGet در VS.NET به Https تنظیم شده است. اگر دسترسی به Http s برای شما به کندی صورت می‌گیرد فقط کافی است مسیر فید آن را در منوی Tools، گزینه‌ی Options، ذیل قسمت Package manager یافته و به <http://nuget.org/api/v2> تغییر دهید؛ یعنی به Http خالی، بجای Http s؛ تا سرعت دریافت بسته‌های NuGet مورد نظر افزایش یابد.

- این بسته از طریق آدرس ذیل نیز قابل دریافت است:

<https://az320820.vo.msecnd.net/packages/propertychanged.fody.1.42.0.nupkg>

همین آدرس را در IE وارد کنید. اگر کار نکرد احتمالاً تنظیمات IE شما به هم ریخته است؛ چون تنظیمات آن به صورت مستقیم روی تنظیمات اتصالی برنامه‌های دات نت تاثیر دارند.

نویسنده: داستان

تاریخ: ۱۳۹۳/۰۲/۲۱ ۱۰:۲۷

ممنون؛ خب چطوری رویداد PropertyChanged رو هندل کنیم؟ (همچنین رویدادی به کلاس اضافه نمی‌شود)

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۲/۲۱ ۱۱:۰۰

- نیازی به اینکار نیست (چون کدهای آن به صورت خودکار به اسمبلی اضافه می‌شوند).

- اگر هدف این است که در اینجا در مورد یک خاصیت دیگر نیز اطلاع رسانی صورت گیرد، [از ویژگی‌های دیگر آن](#) به نام‌های AlsoNotifyFor و DoNotNotify استفاده کنید.

نویسنده: جوادی

تاریخ: ۱۳۹۳/۰۴/۱۴ ۱۳:۵۷

سلام؛

چطور یک Subscriber برای رویداد PropertyChanged بنویسیم، در صورتیکه این رویداد وجود ندارد؟ در صورت امکان، یک مثال از نحوه اطلاع رسانی هنگام تغییر مقدار Property بزنید. ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۴/۱۴ ۱۴:۰۵

از [NotificationInterception](#) آن استفاده کنید یا از روش [On_PropertyName_Changed](#)