

عنوان: کامپایل خودکار یک پروژه برای دو فریم ورک

نویسنده: وحید نصیری

تاریخ: ۱۸:۵ ۱۳۹۳/۰۶/۲۵

آدرس: www.dotnettips.info

گروه‌ها: MSBuild, .NET 4.5, Compile

فرض کنید می‌خواهید زمانیکه دکمه‌ی build در VS.NET فشرده شد، دو نسخه‌ی دات نت 4 و دات نت 4.5، از پروژه‌ی شما در پوشه‌های مجزایی کامپایل شده و قرار گیرند. در ادامه نحوه‌ی انجام این‌کار را بررسی خواهیم کرد.

پروژه نمونه

تنظیمات ذیل را بر روی یک پروژه از نوع class library دات نت 4 در VS 2013 اعمال خواهیم کرد.

ویرایش فایل پروژه برنامه

برای اینکه تنظیمات کامپایل خودکار مخصوص دات نت 4.5 را نیز به این پروژه دات نت 4 اضافه کنیم، نیاز است فایل csproj آن‌را مستقیماً ویرایش نمائیم. این تغییرات شامل مراحل ذیل هستند:

الف) تعریف متغیر Framework

```
<PropertyGroup>
  <!-- ...-->
  <Framework Condition=" '$(Framework)' == '' ">NET40</Framework>
</PropertyGroup>
```

به ابتدای فایل csproj در قسمت PropertyGroup آن یک متغیر جدید را به نام Framework اضافه کنید. از این متغیر در شرط‌های کامپایل استفاده خواهد شد.

ب) ویرایش مسیر خروجی تنظیمات کامپایل فعلی

```
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
  <!-- ...-->
  <OutputPath>bin\$(Configuration)\$(Framework)\</OutputPath>
</PropertyGroup>
```

در حال حاضر حداقل تنظیمات کامپایل حالت debug، در فایل پروژه موجود است. مقدار OutputPath آن‌را به نحو فوق تغییر دهید تا خروجی نهایی را در پوشه‌ای مانند bin\Debug\NET40 ایجاد کند. بدیهی است اگر حالت release هم وجود دارد، نیاز است مقدار OutputPath آن‌را نیز به همین ترتیب ویرایش کرد.

ج) افزودن تنظیمات کامپایل دات نت 4.5 به پروژه جاری

```
<PropertyGroup Condition=" '$(Framework)' == 'NET45' And '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
  <TargetFrameworkVersion>v4.5</TargetFrameworkVersion>
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DebugSymbols>true</DebugSymbols>
  <DebugType>full</DebugType>
  <Optimize>>false</Optimize>
  <OutputPath>bin\$(Configuration)\$(Framework)\</OutputPath>
  <DefineConstants>DEBUG;TRACE;NET45</DefineConstants>
  <ErrorReport>prompt</ErrorReport>
  <WarningLevel>4</WarningLevel>
</PropertyGroup>

<PropertyGroup Condition=" '$(Framework)' == 'NET45' And '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
  <TargetFrameworkVersion>v4.5</TargetFrameworkVersion>
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DebugType>pdbonly</DebugType>
  <Optimize>true</Optimize>
  <OutputPath>bin\$(Configuration)\$(Framework)\</OutputPath>
  <DefineConstants>TRACE;NET45</DefineConstants>
```

```
<ErrorReport>prompt</ErrorReport>
<WarningLevel>4</WarningLevel>
</PropertyGroup>
```

در اینجا تنظیمات حالت debug و release مخصوص دات نت 4.5 را مشاهده می‌کنید. برای نگارش‌های دیگر، تنها کافی است مقدار TargetFrameworkVersion را ویرایش کنید. همچنین اگر به DefineConstants آن دقت کنید، مقدار NET45 نیز به آن اضافه شده‌است. این مورد سبب می‌شود که بتوانید در پروژه‌ی جاری، شرطی‌هایی را ایجاد کنید که کدهای آن فقط در حین کامپایل برای دات نت 4.5 به خروجی اسمبلی نهایی اضافه شوند:

```
#if NET45
public class ExtensionAttribute : Attribute { }
#endif
```

د) افزودن تنظیمات پس از build

در انتهای فایل csproj قسمت AfterBuild به صورت کامنت شده موجود است. آن را به نحو ذیل تغییر دهید:

```
<Target Name="AfterBuild">
  <Message Text="Enter After Build TargetFrameworkVersion:${TargetFrameworkVersion}
Framework:${Framework}" Importance="high" />
  <MSBuild Condition="'${Framework}' != 'NET45'" Projects="$(MSBuildProjectFile)"
Properties="Framework=NET45" RunEachTargetSeparately="true" />
  <Message Text="Exiting After Build TargetFrameworkVersion:${TargetFrameworkVersion}
Framework:${Framework}" Importance="high" />
</Target>
```

این تنظیم سبب می‌شود تا کامپایل مخصوص دات نت 4.5 نیز به صورت خودکار فعال گردد و خروجی آن در مسیر bin\Debug\NET45 به صورت جداگانه‌ای قرار گیرد.

```
Test.cs
DualTargetFrameworks
DualTargetFrameworks.Test
5
6 namespace DualTargetFrameworks
7 {
8
9 #if NET45
10 public class ExtensionAttribute : Attribute { }
11 #endif
12
13 public class Test
14 {
15 }
16 }
```

```
Output
Show output from: Build
1>----- Build started: Project: DualTargetFrameworks, Configuration: Debug Any CPU -----
1> DualTargetFrameworks -> D:\Prog\1393\DualTargetFrameworks\DualTargetFrameworks\bin\Debug\NET40\DualTargetFrameworks.dll
1> Enter After Build TargetFrameworkVersion:v4.0 Framework:NET40
1> DualTargetFrameworks -> D:\Prog\1393\DualTargetFrameworks\DualTargetFrameworks\bin\Debug\NET45\DualTargetFrameworks.dll
1> Enter After Build TargetFrameworkVersion:v4.5 Framework:NET45
1> Exiting After Build TargetFrameworkVersion:v4.5 Framework:NET45
1> Exiting After Build TargetFrameworkVersion:v4.0 Framework:NET40
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

برای آزمایش بیشتر، فایل csproj نهایی را از اینجا می‌توانید دریافت کنید:

نظرات خوانندگان

نویسنده: مهدی نقدی
تاریخ: ۱۳۹۳/۰۶/۳۰ ۲۰:۵۶

خیلی موضوع خوبی بود. واقعا خسته نباشید.
اگر از کتابخانه ای استفاده کنیم که DLL دات نت ۴ و ۴.۵ جداگانه ای رو ارائه داده باشه، چطور میشه این موضوع را پوشش داد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۳۰ ۲۲:۰۱

ویژگی‌های Condition ذکر شده در متن، در مورد ارجاعات هم قابل تنظیم است:

```
<Reference Include="MyAssembly" Condition=".....">  
  <SpecificVersion>False</SpecificVersion>  
  <HintPath>path\to\MyAssembly.dll</HintPath>  
</Reference>
```

همچنین در مورد فایل‌های سورس:

```
<Compile Include="Class20.cs" Condition="....." />
```