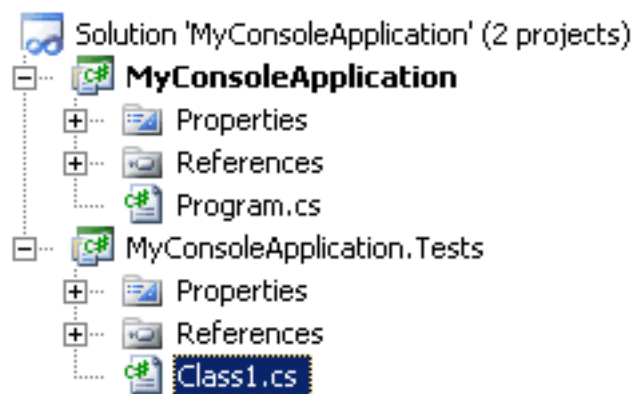


در ویژوال استودیو ذیل منوی File، گزینه‌ای وجود دارد به نام Export template که کار آن تهیه یک قالب، بر اساس ساختار پروژه جاری است. این قابلیت جهت تهیه قالب‌های سفارشی، برای کاهش زمان تهیه پروژه‌ها بسیار مفید است. به این ترتیب می‌توان بسیاری از نکات مدنظر را، در یک قالب ویژه لحاظ کرد و به دفعات بدون نیاز به copy/paste مداوم فایل‌ها و تنظیمات اولیه، بسیار سریع یک پروژه جدید دلخواه را ایجاد نمود.

اما ... این قالب تهیه شده، صرفاً بر اساس یکی از چندین پروژه Solution جاری تهیه می‌شود و همچنین نصب و توزیع آن نیز دستی است. در ادامه قصد داریم با نحوه تهیه یک قالب جدید پروژه متشکل از چندین پروژه، به همراه تهیه فایل VSI نصاب آن، آشنا شویم.

تهیه یک ساختار نمونه

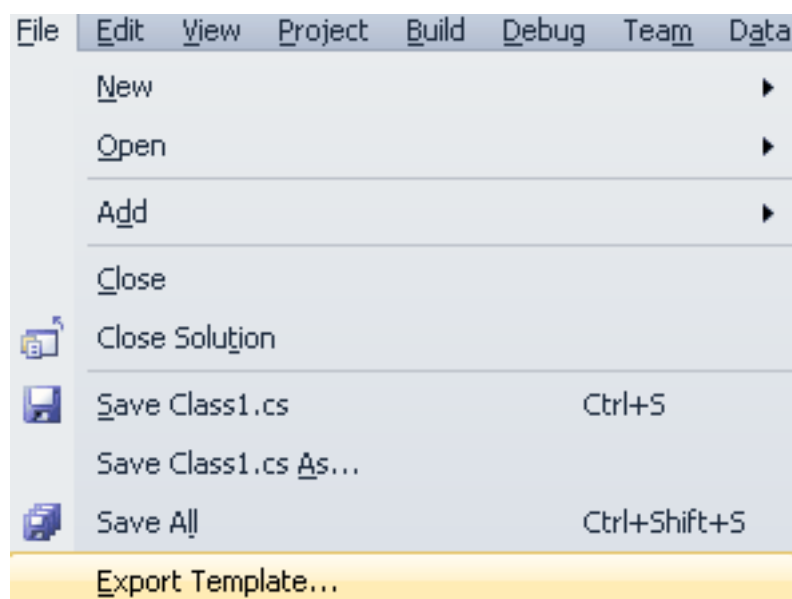
یک پروژه جدید کنسول را به نام MyConsoleApplication ایجاد کنید. سپس به Solution جاری، یک Class library جدید را به نام مثلاً MyConsoleApplication.Tests اضافه نمائید. تا اینجا به شکل زیر خواهیم رسید:



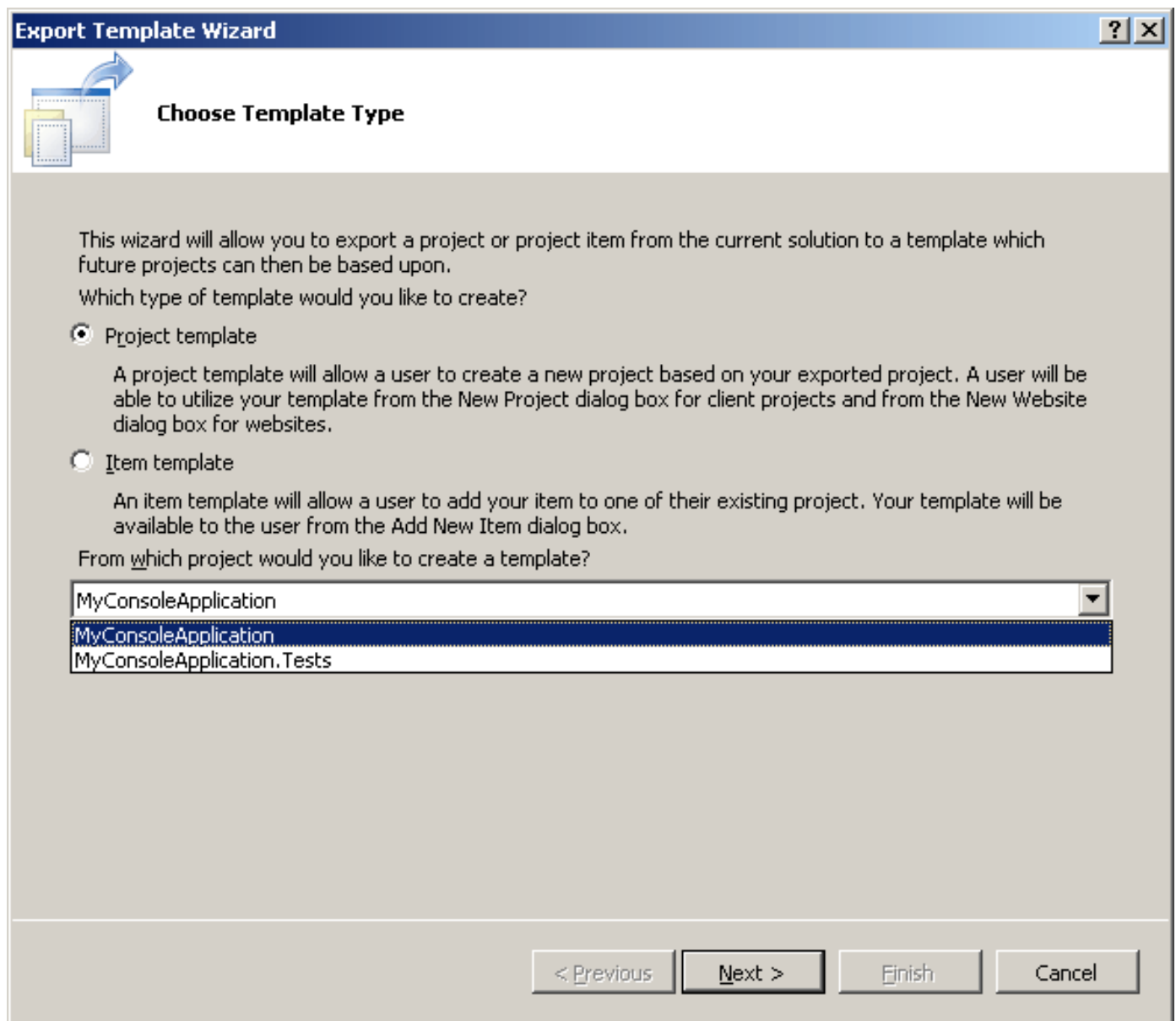
اکنون قصد داریم از این پروژه خاص، یک قالب تهیه کنیم؛ تا هر بار نخواهیم یک چنین مرحله‌ای را تکرار کنیم.

تهیه قالب به ازای هر پروژه در Solution

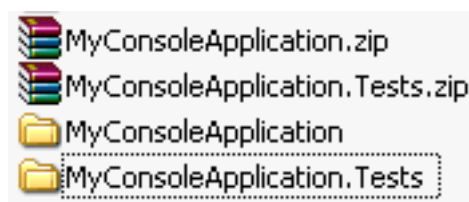
در همین حال که Solution باز است، به منوی File و گزینه Export template مراجعه کنید.



در اینجا تنها امکان انتخاب یک پروژه وجود دارد. به همین جهت این مرحله را باید به ازای هر تعداد پروژه موجود در Solution یکبار تکرار کرد.



اکنون در پوشه My Documents\Visual Studio 2010\My Exported Templates دو فایل zip به نام‌های MyConsoleApplication.Tests.zip و MyConsoleApplication.zip وجود دارند. هر دو فایل را توسط برنامه‌های مخصوص گشودن فایل‌های Zip گشوده و تبدیل به دو پوشه باز شده MyConsoleApplication.Tests و MyConsoleApplication کنید.



افزودن فایل MyTemplate.vstemplate چند پروژه‌ای

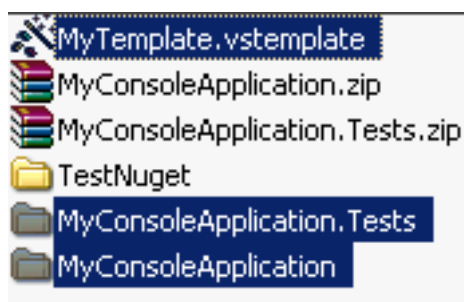
در همین پوشه جاری که اکنون حاوی دو پوشه باز شده است، یک فایل متنی جدید را با محتوای ذیل به نام

[MyTemplate.vstemplate](#) ایجاد کنید:

```
<VSTemplate Version="3.0.0" Type="ProjectGroup"
xmlns="http://schemas.microsoft.com/developer/vstemplate/2005">
  <TemplateData>
    <Name>MyConsoleApplication</Name>
    <Description>MyConsoleApplication Desc</Description>
    <ProjectType>CSharp</ProjectType>
  </TemplateData>
  <TemplateContent>
    <ProjectCollection>
      <ProjectTemplateLink ProjectName="MyConsoleApplication">
        MyConsoleApplication\MyTemplate.vstemplate</ProjectTemplateLink>
      <ProjectTemplateLink ProjectName="MyConsoleApplication.Tests">
        MyConsoleApplication.Tests\MyTemplate.vstemplate</ProjectTemplateLink>
      </ProjectCollection>
    </TemplateContent>
  </VSTemplate>
```

در اینجا به ازای هر پروژه، یک ProjectTemplateLink ایجاد خواهد شد که به فایل MyTemplate.vstemplate موجود در قالب آن اشاره می‌کند.

در ادامه این دو پوشه باز شده و فایل MyTemplate.vstemplate فوق را انتخاب کرده:



و همگی را تبدیل به یک فایل zip جدید کنید؛ مثلاً به نام MyConsoleApplicationTemplates.zip.

تهیه فایل نصاب از قالب پروژه جدید

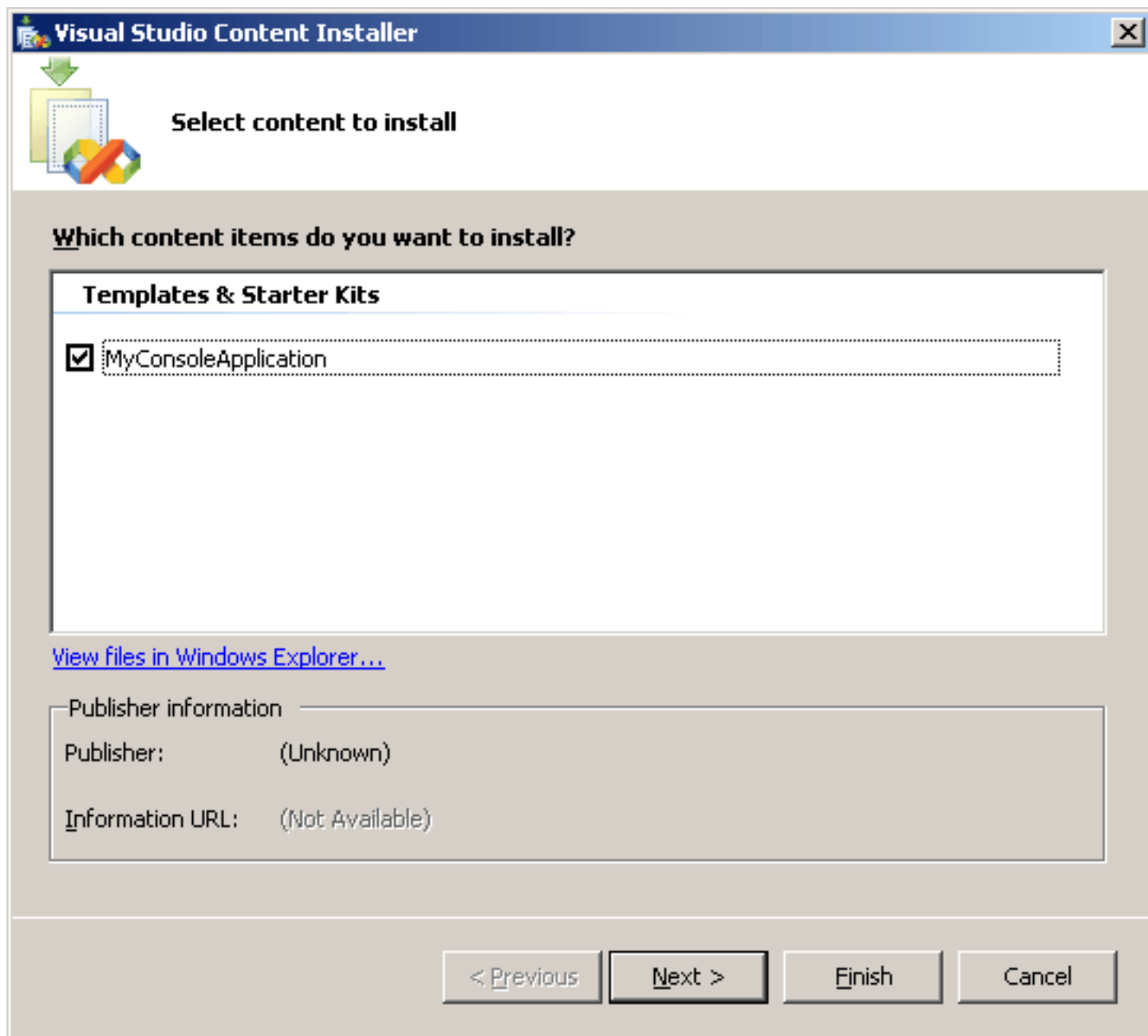
تا اینجا موفق شدیم، چندین قالب پروژه تهیه شده را به هم متصل کرده و تبدیل به یک فایل zip نهایی کنیم. مرحله بعد ایجاد فایلی است متنی به نام [MyConsoleApplicationTemplates.vscontent](#) با محتویات زیر:

```
<VSContent xmlns="http://schemas.microsoft.com/developer/vscontent/2005">
  <Content>
    <FileName>MyConsoleApplicationTemplates.zip</FileName>
    <DisplayName>MyConsoleApplication</DisplayName>
    <Description>A C# project that ...</Description>
    <FileContentType>VSTemplate</FileContentType>
    <ContentVersion>1.0</ContentVersion>
    <Attributes>
      <Attribute name="ProjectType" value="Visual C#" />
      <Attribute name="ProjectSubType" value="Web" />
      <Attribute name="TemplateType" value="Project" />
    </Attributes>
  </Content>
</VSContent>
```

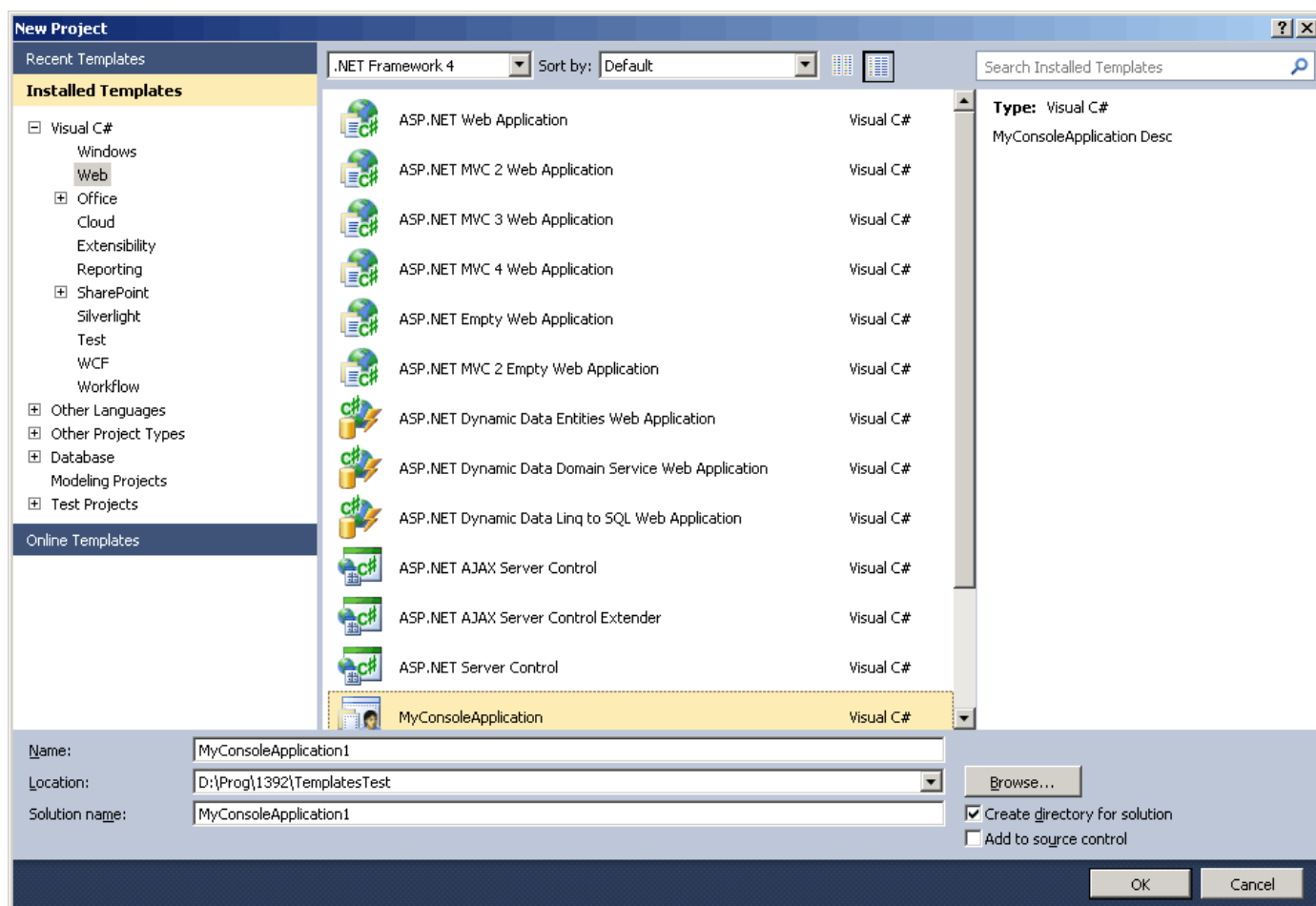
در اینجا توسط قسمت Attributes مشخص می‌کنیم که قالب پروژه جدید باید در صفحه new project، در کدام مدخل قرار گیرد. بنابراین مطابق تنظیمات فوق، قالب جدید ذیل پروژه‌های وب سی‌شارپ قرار خواهد گرفت. مقدار FileName آن دقیقاً معادل نام

فایل zip ایی است که در مرحله قبل ایجاد کردیم.

مرحله بعد انتخاب دو فایل MyConsoleApplicationTemplates.vsix و MyConsoleApplicationTemplates.zip و تبدیل ایندو به یک فایل zip جدید است. پس از ایجاد فایل جدید، پسوند آنرا به VSI تغییر دهید؛ برای مثال نام آنرا به MyConsoleApplicationTemplates.vsi تغییر دهید. اکنون این فایل نهایی با دوبار کلیک بر روی آن قابلیت اجرا و نصب خودکار را پیدا می‌کند.



پس از نصب، بلافاصله ذیل قسمت پروژه‌های وب قابل دسترسی و استفاده خواهد بود:



بنابراین به صورت خلاصه:

- (1) به ازای هر پروژه، یک فایل قالب zip معادل آن باید تهیه شود.
 - (2) تمام این فایل‌های zip را گشوده و تبدیل به پوشه‌های متناظری کنید.
 - (3) یک فایل MyTemplate.vstemplate را در پوشه ریشه مرحله 2 جهت تعریف ProjectTemplateLink ها اضافه کنید.
 - (4) فایل جدید MyTemplate.vstemplate مرحله 3 و تمام پوشه‌های قالب‌های باز شده مرحله 2 را zip کنید.
 - (5) سپس یک فایل vscontent نصاب را تهیه و آنرا با فایل zip مرحله 4 مجددا zip کرده و پسوند آنرا به VSI تغییر دهید.
- اکنون می‌توان این فایل VSI را توزیع کرد.

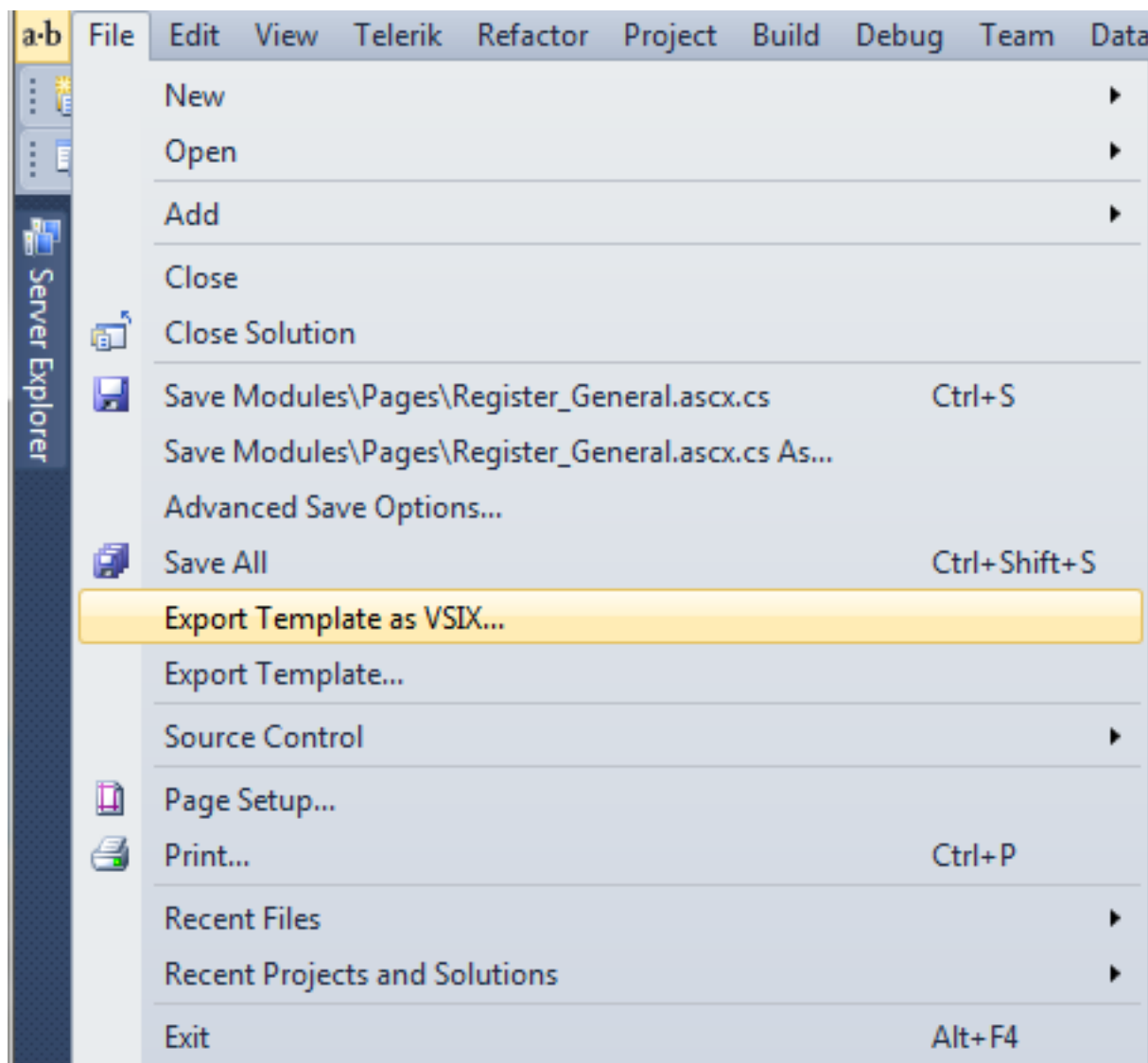
نظرات خوانندگان

نویسنده: مجتبی صحرائی
تاریخ: ۱۱:۵۷ ۱۳۹۲/۰۲/۲۹

سلام و ممنون

بنده از این روش استفاده کرده بودم و نهایتاً برای خودکار سازی این اعمال از افزونه [ExportTemplate\(vsix\).vsix](#) ویژوال استودیو استفاده کردم

طریقه استفاده اون هم به این صورت هستش که پس از نصب گزینه Export Template as VSIX... در منوی فایل ظاهر میشه و با کلیک بر روی اون تمامی پروژه‌های موجود در Solution جاری رو لیست می‌کنه و می‌تونید انتخاب کنید و Export کنید



نویسنده: مجتبی صحرائی
تاریخ: ۱۲:۲ ۱۳۹۲/۰۲/۲۹

نکته تکمیلی اینکه این امکان وجود دارد که پس از انتخاب پروژه(ها)، میتونید برای نصاب خودتون آیکون قرار بدید، لایسنس گذاری کنید و در ضمن ساخت نصاب عمل import شدن به VS هم به طور خودکار انجام بشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۳:۱۰

- ممنون. افزونه خیلی کاربردی و مفیدی است.
- البته در حالت دستی عنوان شده امکان تعریف آیکون و غیره هم هست. در متن، لینک داده شده به مراجع تولید فایل‌های vscontent و vstemplate که برای نمونه یک مدخل اضافه‌تر برای آیکون پیدا می‌کند :

<Icon>__Template_small.png</Icon>

در کل بد نیست یک برنامه نویس بدون پشت صحنه این اعمال به چه صورتی هست.

نویسنده: مجتبی صحرایی
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۳:۴۱

بله دقیقا با نظر شما موافقم

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۷:۵۳

سلام ... خیلی ممنون بابت این افزونه ...
گویا این افزونه روی VS 2012 کار نمیکنه! ... راهی هست که بشه کار کنه؟!

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۸:۰۰

با سلام و عرض تشکر

یک سوال : اینکه بخواهیم از روشی که شما ارائه دادید استفاده کنیم ولی اگر بخواهیم بدین گونه باشد که :
پس از درست کردن قالب مد نظر پروژه بخواهیم یک ساختاری مانند پروژه MVC یا یک ساختار دلخواه داشته باشیم بصورتی که همواره می‌خواهیم در قالب پروژه ، 3 فولدر وجود داشته باشند که فولدر اول همانام پروژه + Module باشد : اگر نام پروژه هنگام ایجاد یا Add کردن Factor باشد یک فولدر در داخل همان پروژه جدید بنام FactorModule ایجاد شود و دو فولدر دیگه FactorAdmin و FactorUser در آن ایجاد شود و به همین ترتیب...؟!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۸:۰۱

فایل extension.vsixmanifest [افزونه رو](#) باید ویرایش کنید (فایل vsix در اصل یک فایل zip است). مثلا VisualStudio
Version آن الان 10 است که باید بشود 11. بعد در SupportedFrameworkRuntimeEdition آن باید MaxVersion به 4.5 تنظیم شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۹ ۲۰:۰۴

امکان سفارشی سازی قالب ساز با کدنویسی هم میسر است. نیاز است اینترفیس IWizard پیاده سازی شود. در اینجا هر نوع کدی رو که لازم بود می‌شود در متد ProjectFinishedGenerating آن تدارک دید. مثلا پوشه درست کند، تنظیمات پروژه را تغییر دهد و امثال آن.

- یک مثال از پیاده سازی اینترفیس IWizard:

[Creating custom project template with wizard for Visual Studio](#)

- مثلاً پروژه sharp-architecture از [همین روش](#) استفاده می‌کند.

نویسنده: وحید نصیری
تاریخ: ۲۲:۴۳ ۱۳۹۲/۰۲/۲۹

یک نکته: روش دیگر ساخت قالب، استفاده از برنامه [Templify](#) است.

نویسنده: وحید نصیری
تاریخ: ۰:۲۱ ۱۳۹۲/۰۲/۳۰

یک نکته تکمیلی دیگر:

با نصب SDK ویژوال استودیو ([^](#) و [^](#)) یک قالب جدید تولید فایل‌های VSIX به مجموعه قالب‌های پروژه‌ها اضافه می‌شود.

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۰ ۱۳۹۲/۰۲/۳۰

یک نکته مهم!

اگر روش فوق را امتحان کنید (چه استفاده از افزونه یاد شده یا حتی روش دستی مقدماتی فوق)، هر نامی را که در ابتدای کار ایجاد Solution جدید وارد کنید، به زیر پروژه‌های اضافه شده اعمال نمی‌شود. یعنی همان نام ابتدایی خودشان را خواهند داشت که این مورد اصلاً جالب نیست.

برای رفع آن نیاز است از متغیری به نام \$safeprojectname\$ استفاده شود (هرجایی که نام پروژه به صورت مستقیم استفاده شده، حتی نام پوشه‌ها یا فایل‌ها) به همراه ReplaceParameters=true. یک مثال را در این مورد در پیوست ذیل می‌توانید دریافت کنید:

[MyConsoleApplicationTemplates.zip](#)

روش نصب دستی این قالب با کپی کردن آن در پوشه My Documents\Visual Studio xyz\Templates\ProjectTemplates است.

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۴ ۱۳۹۲/۰۲/۳۰

و یا از \$safeprojectname\$ باید استفاده شود به روشی که [در اینجا توضیح دادم](#).

نویسنده: وحید نصیری
تاریخ: ۱۵:۲ ۱۳۹۲/۰۳/۰۴

نکات مطرح شده در این مطلب، تبدیل به یک پروژه شد: « [Solution template generator](#) »

نویسنده: ایلیا اکبری فرد
تاریخ: ۸:۴۵ ۱۳۹۲/۰۳/۳۱

با سلام.

یک پروژه class library در solution خود دارم. چگونه می‌شود فضای نام پیش فرض این پروژه و فایل‌های درون آنرا براساس \$safeprojectname\$ تغییر داد.

نویسنده: وحید نصیری
تاریخ: ۸:۵۶ ۱۳۹۲/۰۳/۳۱

- اینکار باید جداگانه (جدای از پروژه در حال کار) و به صورت دستی انجام شود (یک search و replace است).

- یا پروژه « [Solution template generator](#) » این کارها رو به صورت خودکار انجام می‌ده.

نویسنده: جواد جوادی
تاریخ: ۱۰:۴۹ ۱۳۹۴/۰۴/۰۲

سلام؛ در بخش آخر که Visual Studio Content Installer می باشد [طبق لینک](#) ارجاعی شما برای vs 2013 پشتیبانی ندارد در قسمت نسخه‌های پشتیبانی 2005 و 2008 و 2010 و 2012 است ولی 2013 نیست و در صورت اجرای فایل vsi. با خطای زیر مواجه می‌شویم :

```
Installation stoped because the directory for projectType value did not exsist .. the projectType is invalid for your installation of Visual Studio
```

در صورت حذف ProjectType/Attribute نصب انجام می‌شود ولی در لیست پروژه‌ها نمایش داده نمی‌شود.
با تشکر از شما

نویسنده: وحید نصیری
تاریخ: ۱۱:۰۰ ۱۳۹۴/۰۴/۰۲

نیازی نیست تا این کارها را دستی انجام دهید. نکات مطرح شده در این مطلب، تبدیل به یک پروژه شد: « [Solution template generator](#) »