

NuGet 系列之三



# NuGet包的创建与发布

---

北京理工大学计算机学院  
金旭亮



# 如何创建自己的NuGet包?

---


# 创建NuGet包的三种方式

- 1 使用dotnet CLI工具
- 2 使用Visual Studio
- 3 使用NugetPackageExplorer工具

相关技术细节参考微软网站上的NuGet文档中的“创建包”部分

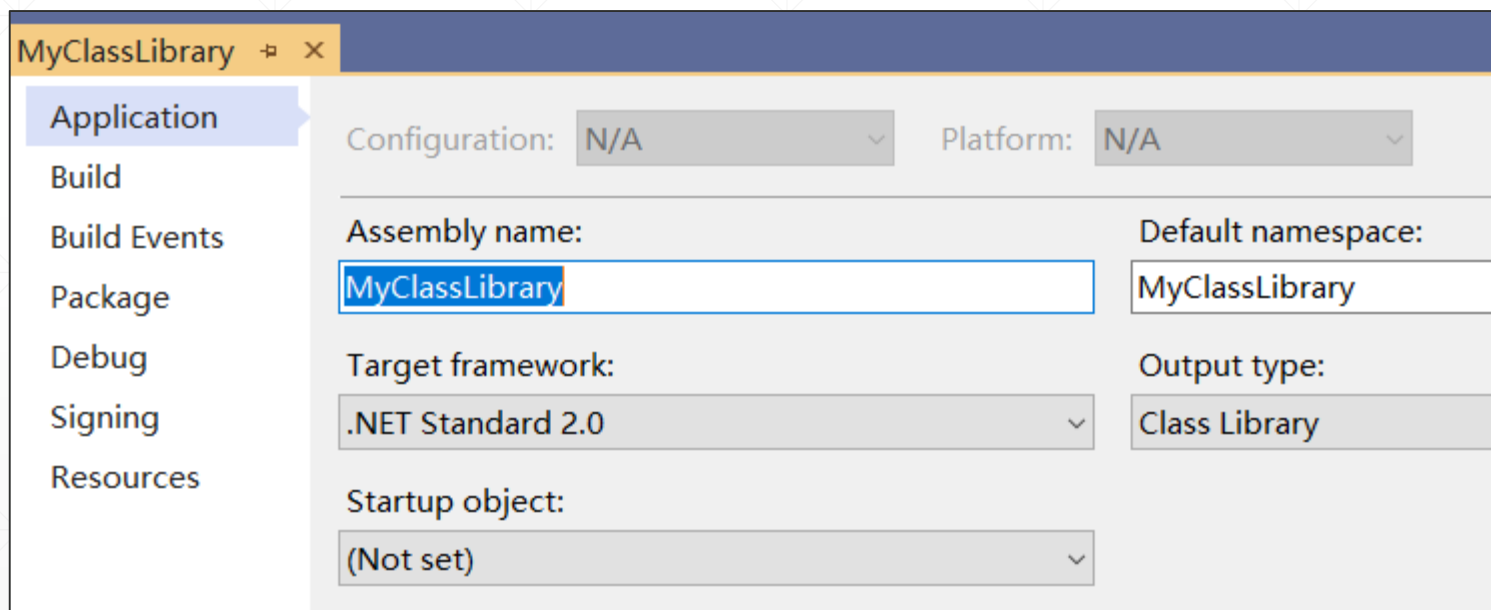
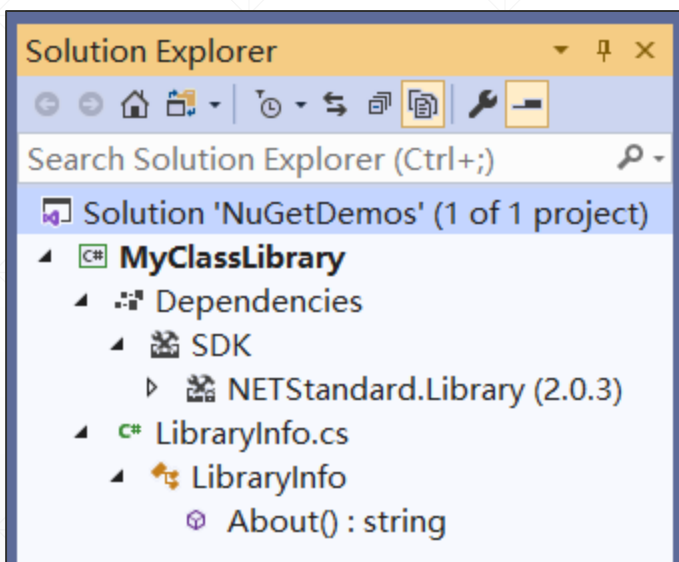
<https://docs.microsoft.com/zh-cn/nuget/create-packages/overview-and-workflow>

# 创建示例类库项目

 Class Library (.NET Standard)  
A project for creating a class library that targets .NET Standard.

C# Android iOS Linux macOS Windows Library

本类库项目将用于保存那些需要被其他项目重用的代码。



# 设定NuGet包必要的参数信息

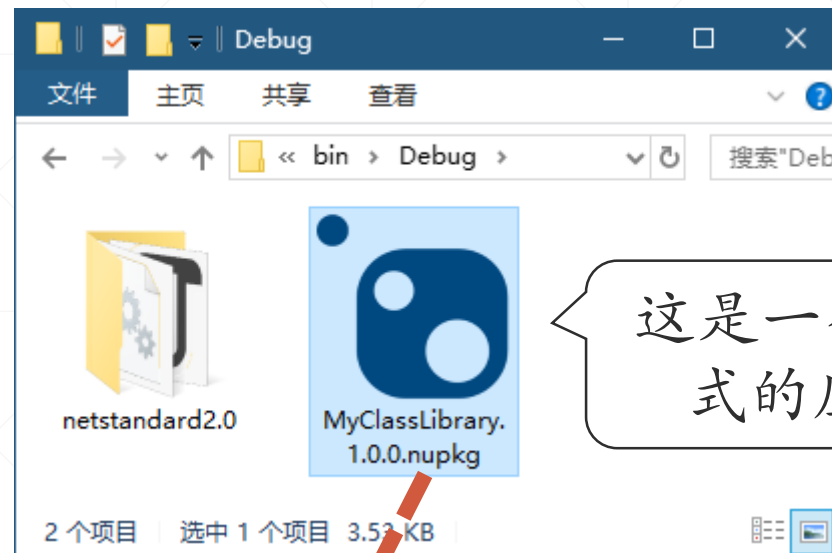
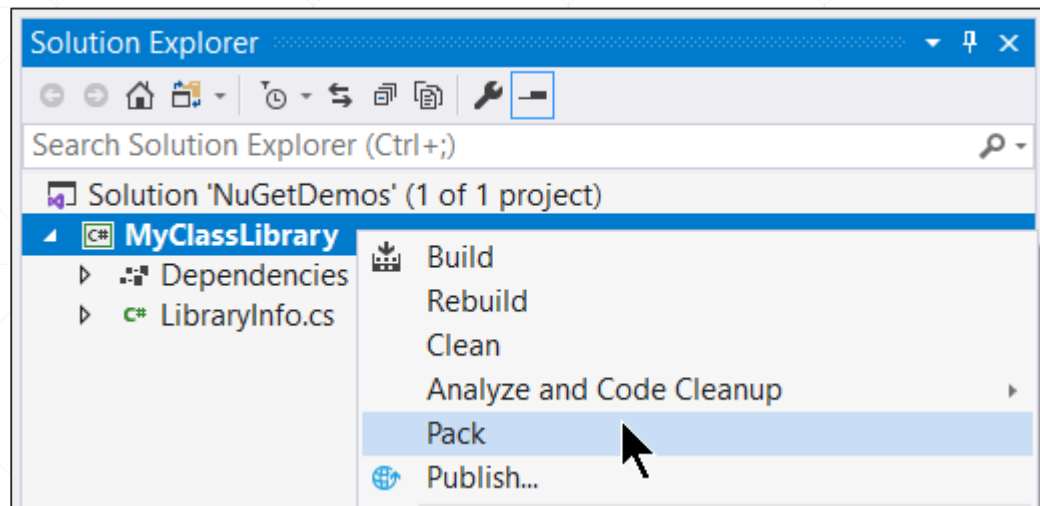
在项目属性窗口的Package选项卡中设置相关的属性，这些设置将保存于.csproj文件中。

```
MyClassLibrary.csproj
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <TargetFramework>netstandard2.0</TargetFramework>
5     <Authors>JinXuLiang</Authors>
6     <Description>一个用于展示NuGet相关特性的示例用包</Description>
7     <Copyright />
8   </PropertyGroup>
9
10 </Project>
```

The screenshot shows the 'Package\*' tab in the Visual Studio project properties window for 'MyClassLibrary'. The left sidebar lists 'Application', 'Build', 'Build Events', 'Package\*', 'Debug', 'Signing', and 'Resources', with 'Package\*' selected. The main area shows configuration options for generating a NuGet package. A dashed blue box highlights the fields for Package id, Package version, Authors, Company, Product, and Description, which are populated with 'MyClassLibrary', '1.0.0', 'JinXuLiang', 'JinXuLiang', 'MyClassLibrary', and '一个用于展示NuGet相关特性的示例用包' respectively. The 'Generate NuGet package on build' and 'Require license acceptance' checkboxes are unchecked. The Configuration and Platform dropdowns are set to 'N/A'.

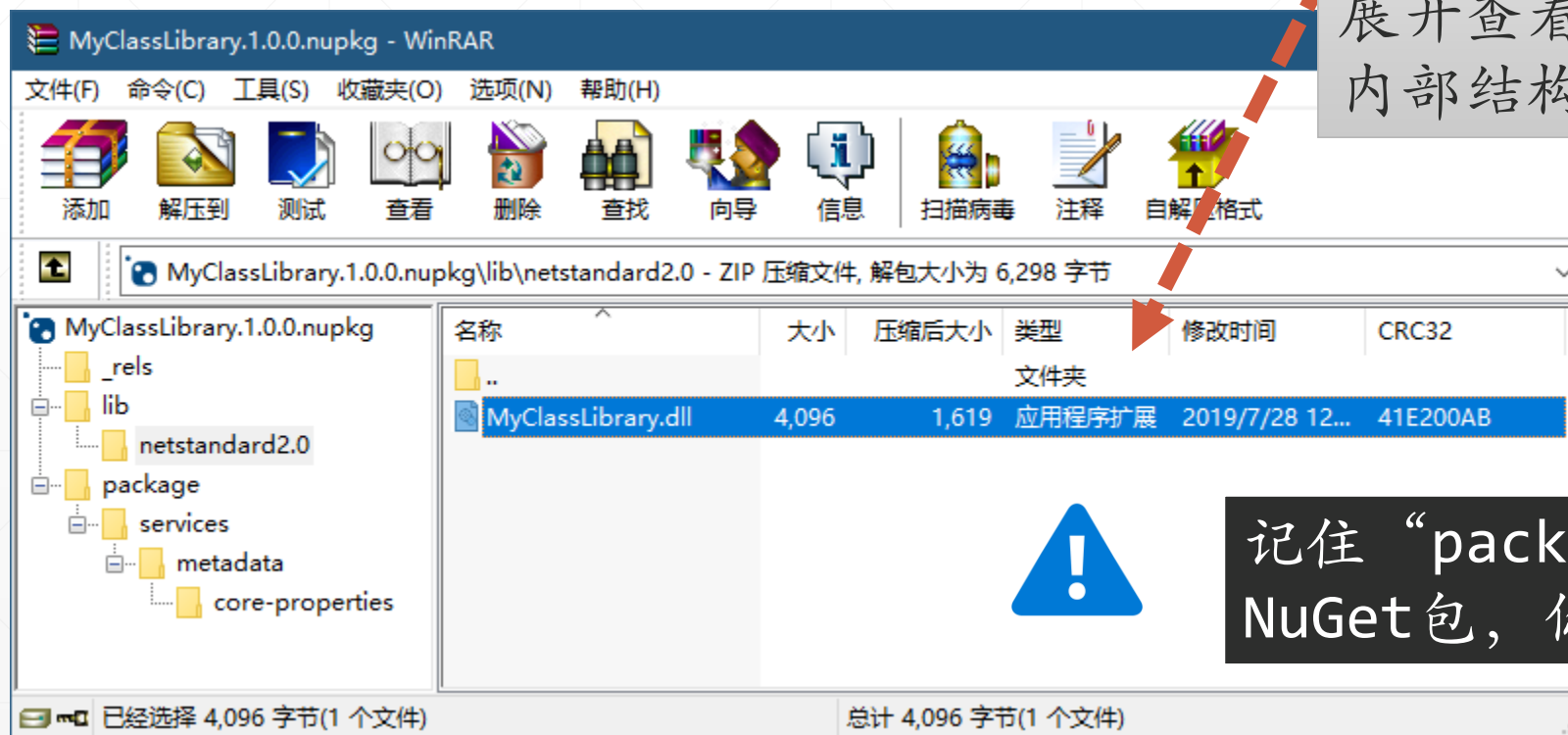
Property	Value
Package id:	MyClassLibrary
Package version:	1.0.0
Authors:	JinXuLiang
Company:	JinXuLiang
Product:	MyClassLibrary
Description:	一个用于展示NuGet相关特性的示例用包

Visual Studio负责保证这两者的数据一致性



这是一个Zip格式的压缩包

展开查看其内部结构



记住“pack”命令可用于创建NuGet包，你会经常用到它的。



# 如何让其他人也能使用自己创建好的NuGet包?

1



使用本地（或网络共享）文件夹构建  
NuGet源

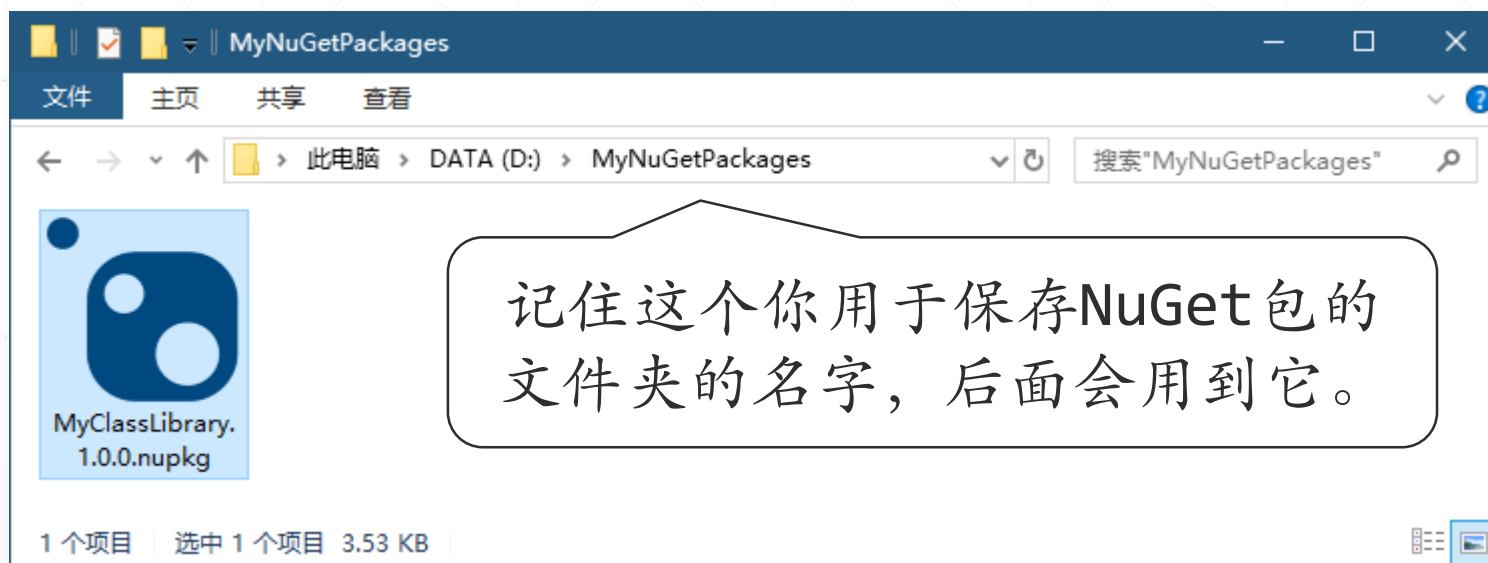
2



使用NuGet帐号和密钥发布到NuGet.org

# 发布私有NuGet包

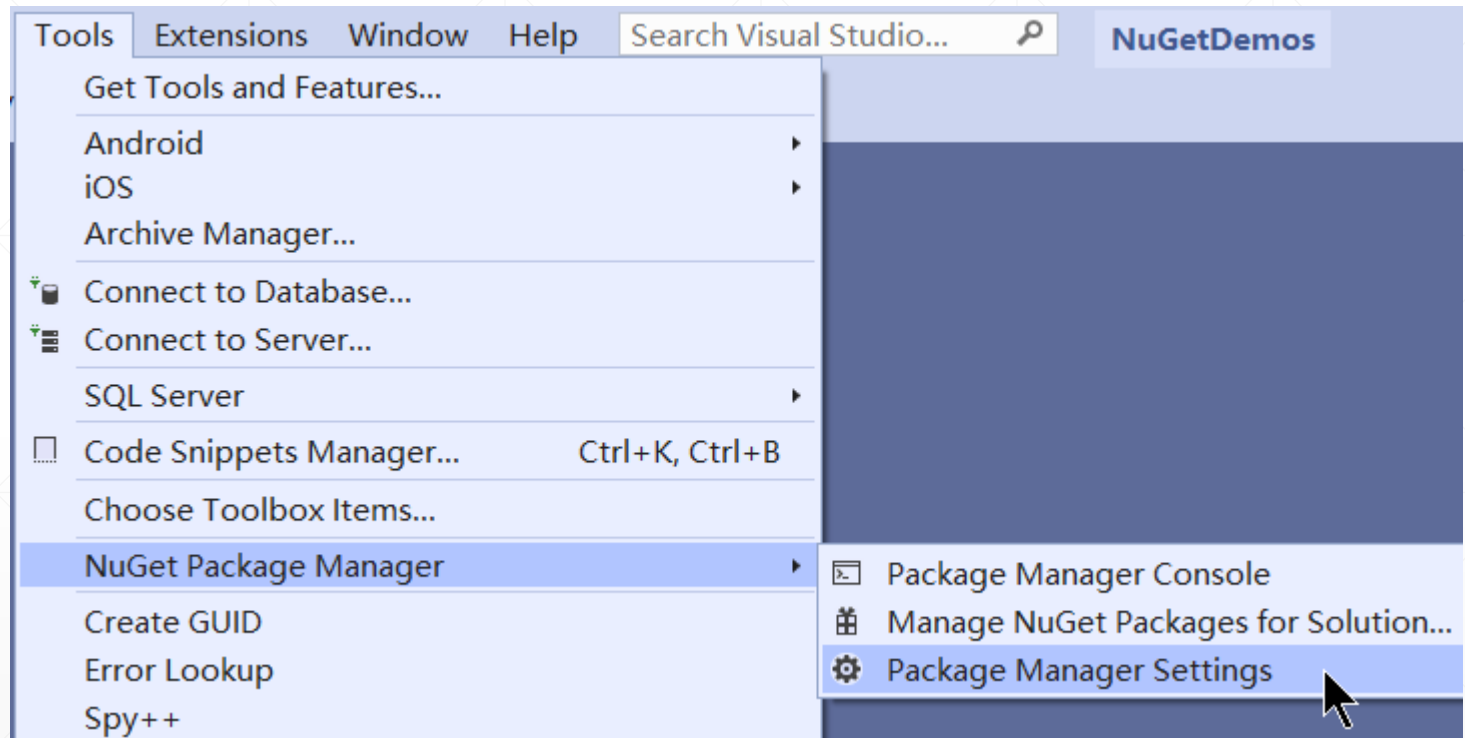
将.nupkg文件移入一个文件夹.....



这个文件夹将专用于保存自己开发的NuGet包。如果是一个局域网共享文件夹，则其他人也可以使用。

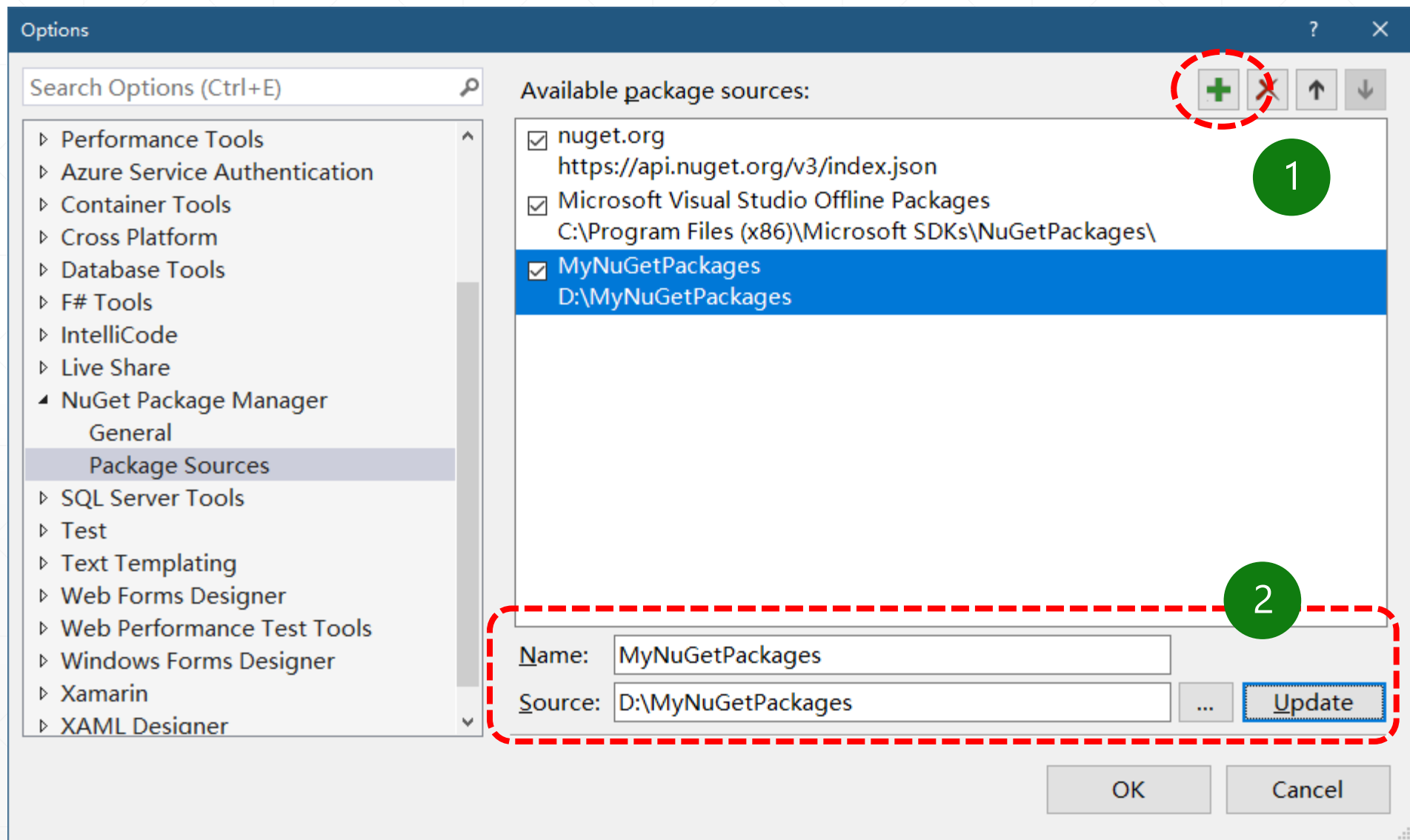


# 准备设置NuGet源



在打开的窗口中设置  
NuGet Package  
Source .....

# 使用私有NuGet包源



# 创建一个.NET Core控制台项目



## Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C#

Linux

macOS

Windows

Console

Program.cs

```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <OutputType>Exe</OutputType>
5     <TargetFramework>netcoreapp3.0</TargetFramework>
6   </PropertyGroup>
7
8 </Project>
```

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'NuGetDemos' (2 of 2 projects)

CoreConsoleApp

Dependencies

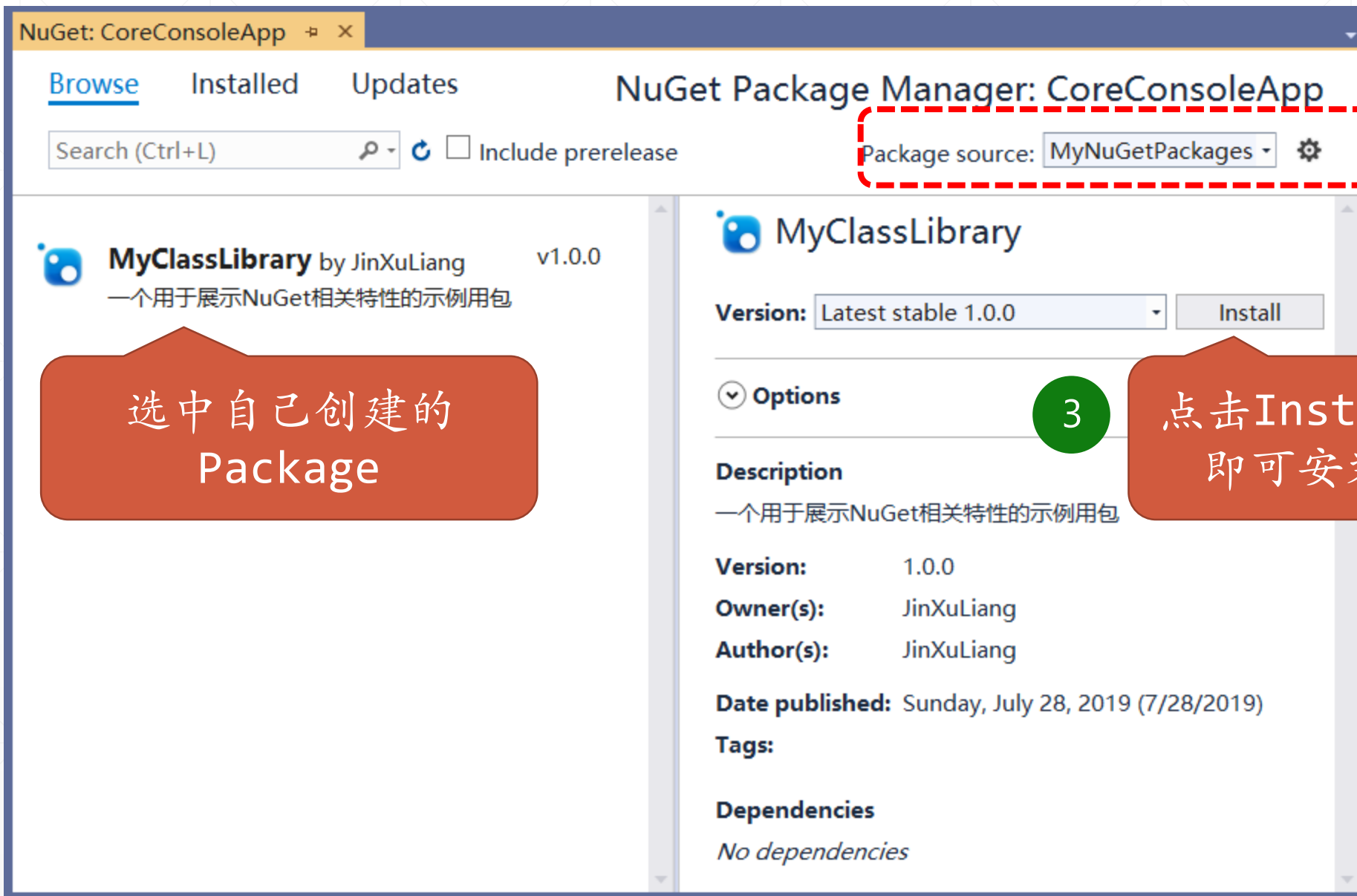
Frameworks

Microsoft.NETCore.App

Program.cs

MyClassLibrary

这个控制台项目将尝试从保存NuGet包的那个文件夹中检索并安装你前面创建的那个示例NuGet包。



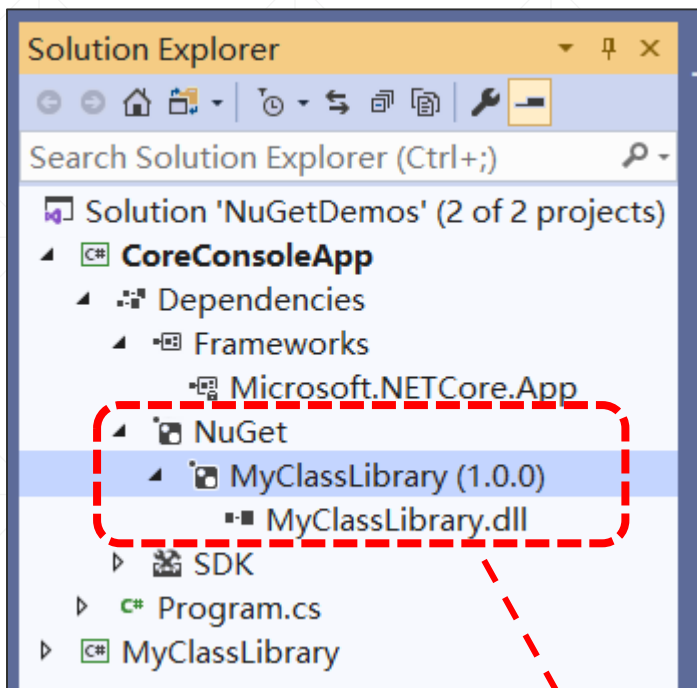
2

选中自己创建的  
Package

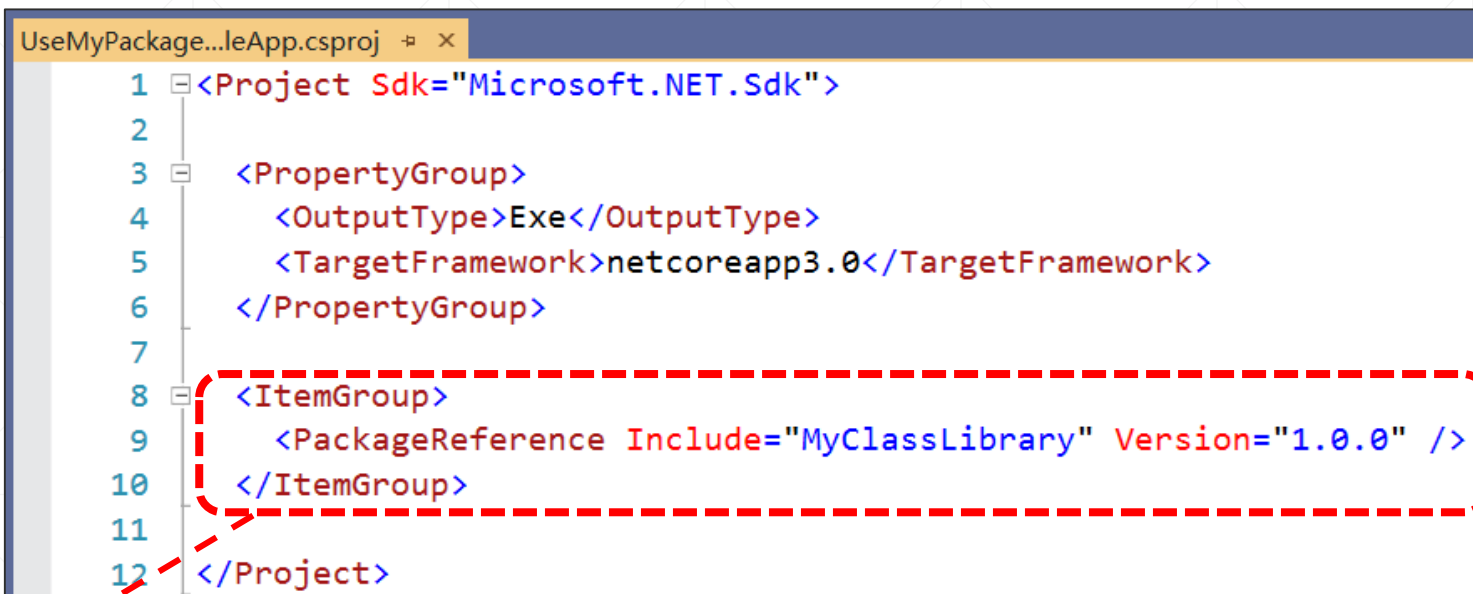
1

3

点击Install  
即可安装



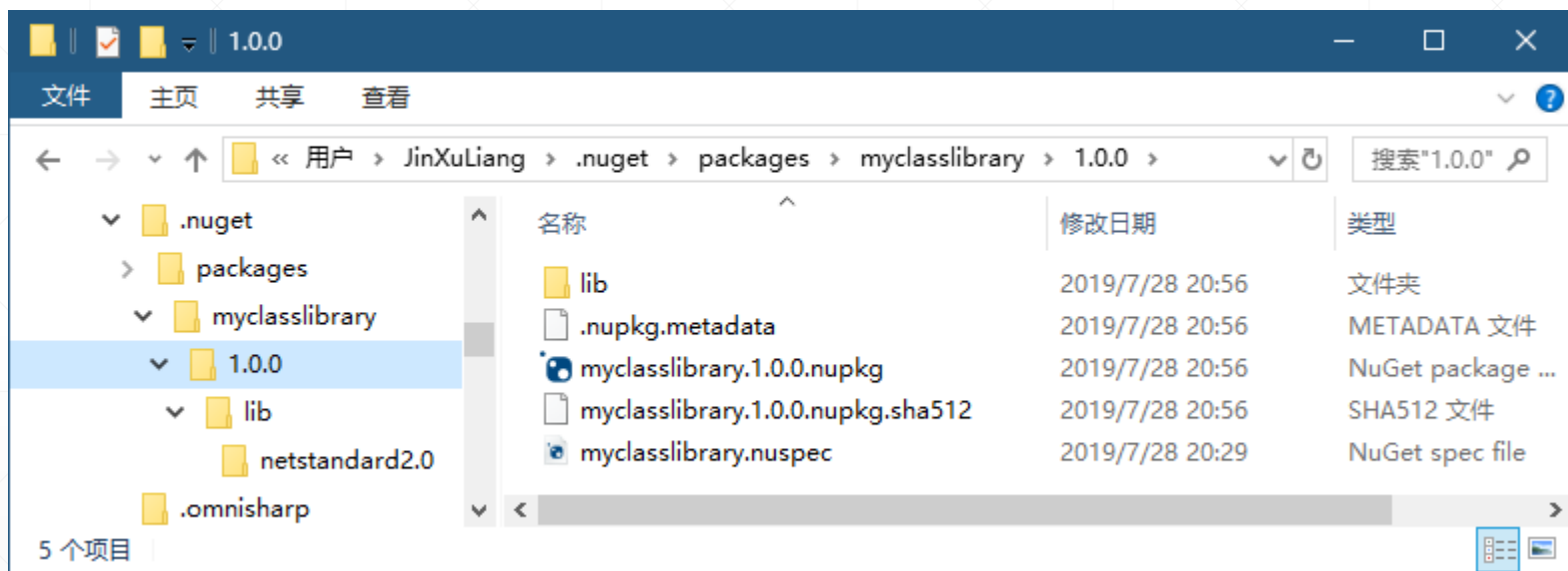
NuGet包安装成功之后，可以在项目文件和解决方案管理器中看到相应的信息



现在可以使用NuGet包中定义的类.....

# 检查一下NuGet包的缓存文件夹

C:\Users\JinXuLiang\.nuget\packages\myclasslibrary\1.0.0



即使这个包是从本地文件夹中检索出来并安装的，NuGet仍然会将它放到全局的缓存文件夹中再保存一份.....

# 如果要发布到Nuget.org.....



1 先使用dotnet CLI工具或Visual Studio创建NuGet包，得到.nupkg文件。

2 在Nuget.org网站注册帐号，获取密钥，之后，就可以使用dotnet nugget push命令发布NuGet包：



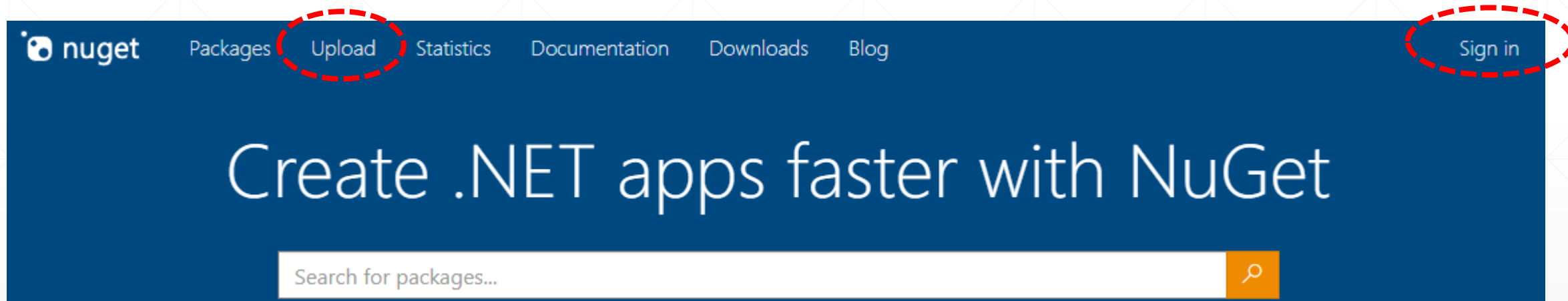
```
dotnet nugget push 包名.nupkg -k 密钥  
-s https://api.nuget.org/v3/index.json
```



# 在线发布到NuGet.org

2 在线直接上传.nupkg文件

1 登录网站






# 创建“多目标框架”的包

---

# 创建一个传统的Window Forms项目

 Windows Forms App (.NET Framework)  
A project for creating an application with a Windows Forms (WinForms) user interface

C# Windows Desktop

Assembly name: TraditionalWinForm

Default namespace: TraditionalWinForm

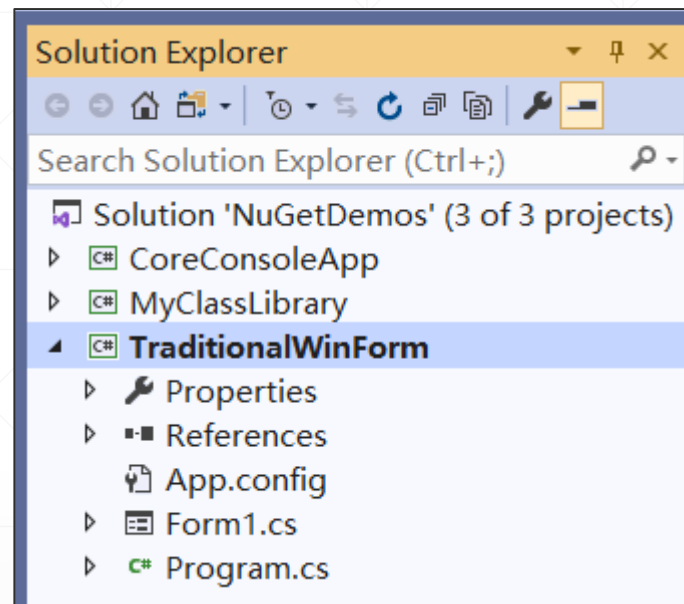
Target framework: .NET Framework 4.5

Output type: Windows Application

☐ Auto-generate binding redirects

Startup object: (Not set)

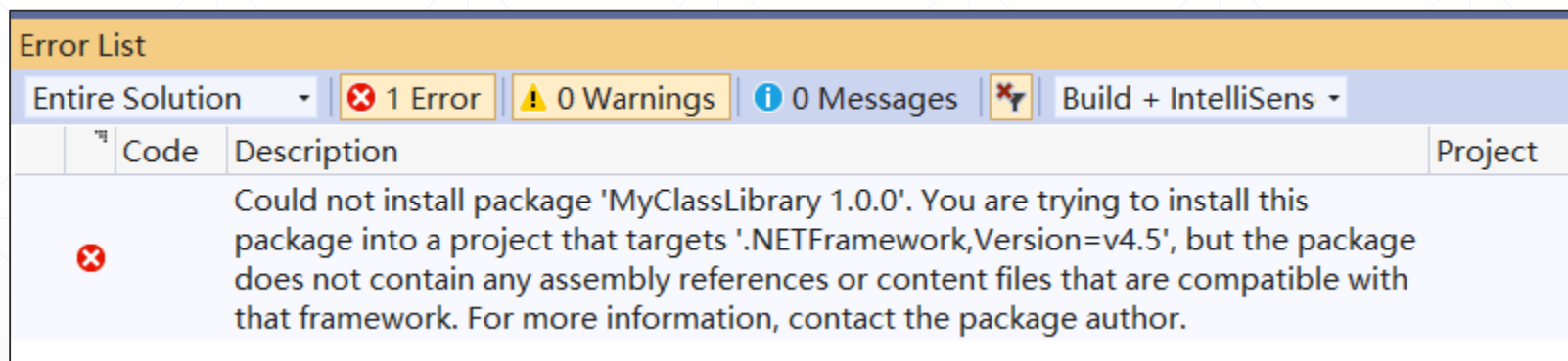
Assembly Information...



注意其目标框架为 .NET Framework 4.5

# 动手试一试

如果尝试使用Nuget Package Manager给TraditionalWinForm项目添加对MyClassLibrary包的引用，Visual Studio会报告以下错误：



# 修改WinForm项目的目标框架

Assembly name:	Default namespace:
<input type="text" value="TraditionalWinForm"/>	<input type="text" value="TraditionalWinForm"/>
Target framework:	Output type:
<input type="text" value=".NET Framework 4.7.2"/>	<input type="text" value="Windows Application"/>



.NET Framework 4.5没有实现.NET Standard 2.0，而.NET Framework 4.7.2实现了，所以针对4.7版本的WinForm项目可以调用MyClassLibrary中的组件。



# 让MyClassLibrary支持.NET Framework 4.5

## 1 修改.csproj的目标框架集合:

```
MyClassLibrary.csproj
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <TargetFrameworks>netstandard2.0;net45</TargetFrameworks>
5     <Authors>JinXuLiang</Authors>
6     <Description>一个用于展示NuGet相关特性的示例用包</Description>
7     <Copyright />
8     <Version>1.0.1</Version>
9   </PropertyGroup>
10
11 </Project>
```

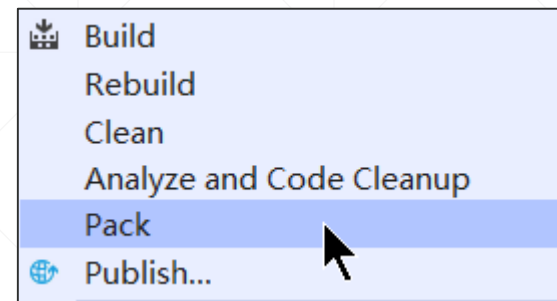
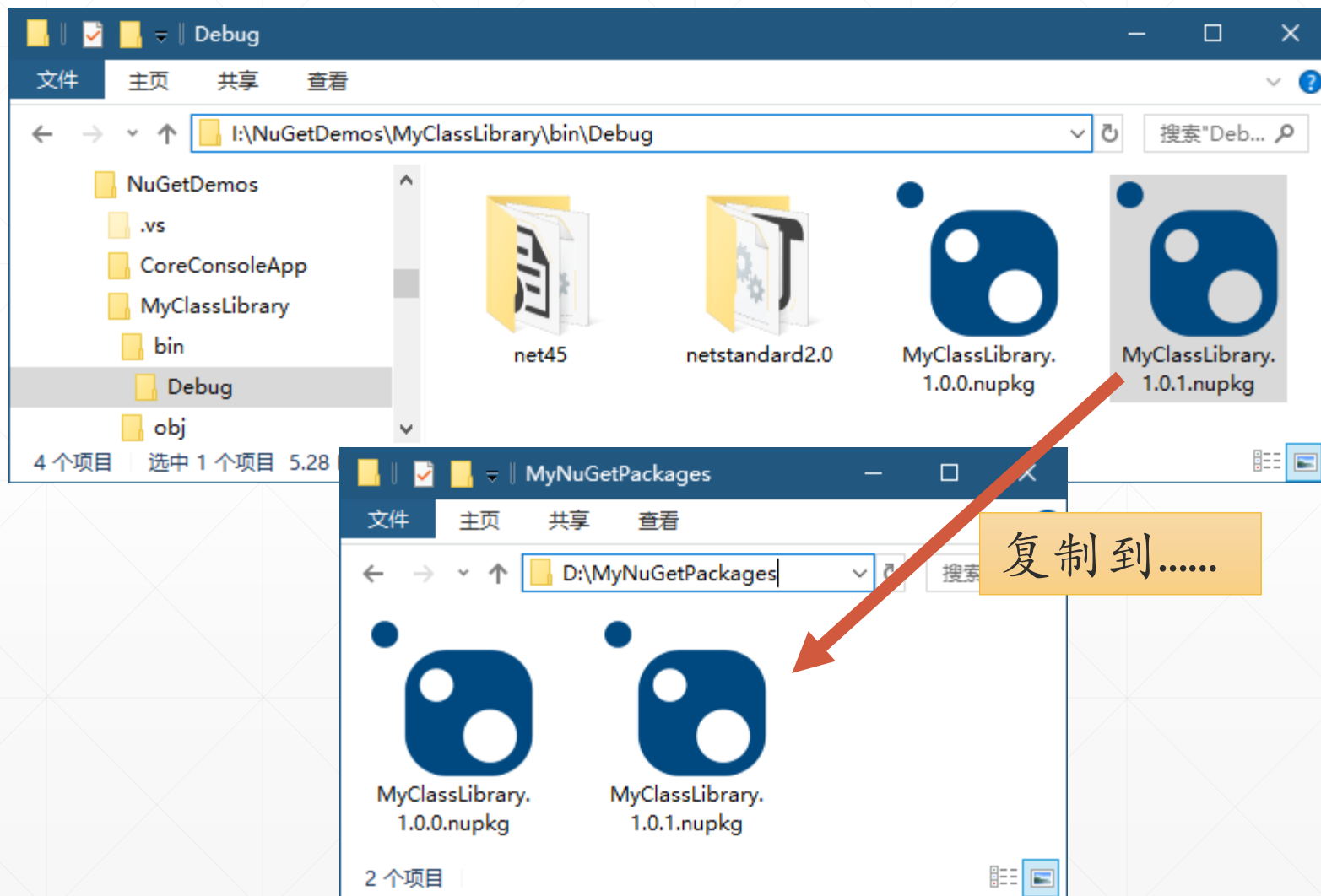


注意XML元素名为  
<TargetFrameworks>，  
是复数形式，另外，包的  
版本号也增加了。

## 2 使用条件编译或有可适配性 代码 (adaptive code) 针 对不同的目标框架编写代码。

```
namespace MyClassLibrary
{
    0 references
    public class LibraryInfo
    {
        0 references
        public static string About()
        {
            #if NET45
                return "一个目标框架为.NET Framework 4.5的类库";
            #elif NETSTANDARD2_0
                return "一个目标框架为.NET Standard的类库";
            #endif
        }
    }
}
```

# 使用Pack命令重新生成NuGet包.....



重新打包之后，会发现项目文件夹中多了一个NuGet包，其版本为1.0.1。



选择本地私有包源


NuGet: TraditionalWinForm

Browse Installed **Updates 1**


Search (Ctrl+L)    ☐ Include prerelease

Package source: **MyNuGetPackages**

---

**MyClassLibrary** by JinXuLiang  
一个用于展示NuGet相关特性的示例用包

v1.0.0  
v1.0.1

**MyClassLibrary**

Installed:

Version:

**Description**  
一个用于展示NuGet相关特性的示例用包

**Version:** 1.0.1

**Owner(s):** JinXuLiang

**Author(s):** JinXuLiang

**Date published:** Sunday, July 28, 2019 (7/28/2019)

**Tags:**

**Dependencies**  
*No dependencies*

# 重新将WinForm项目目标框架改回原值

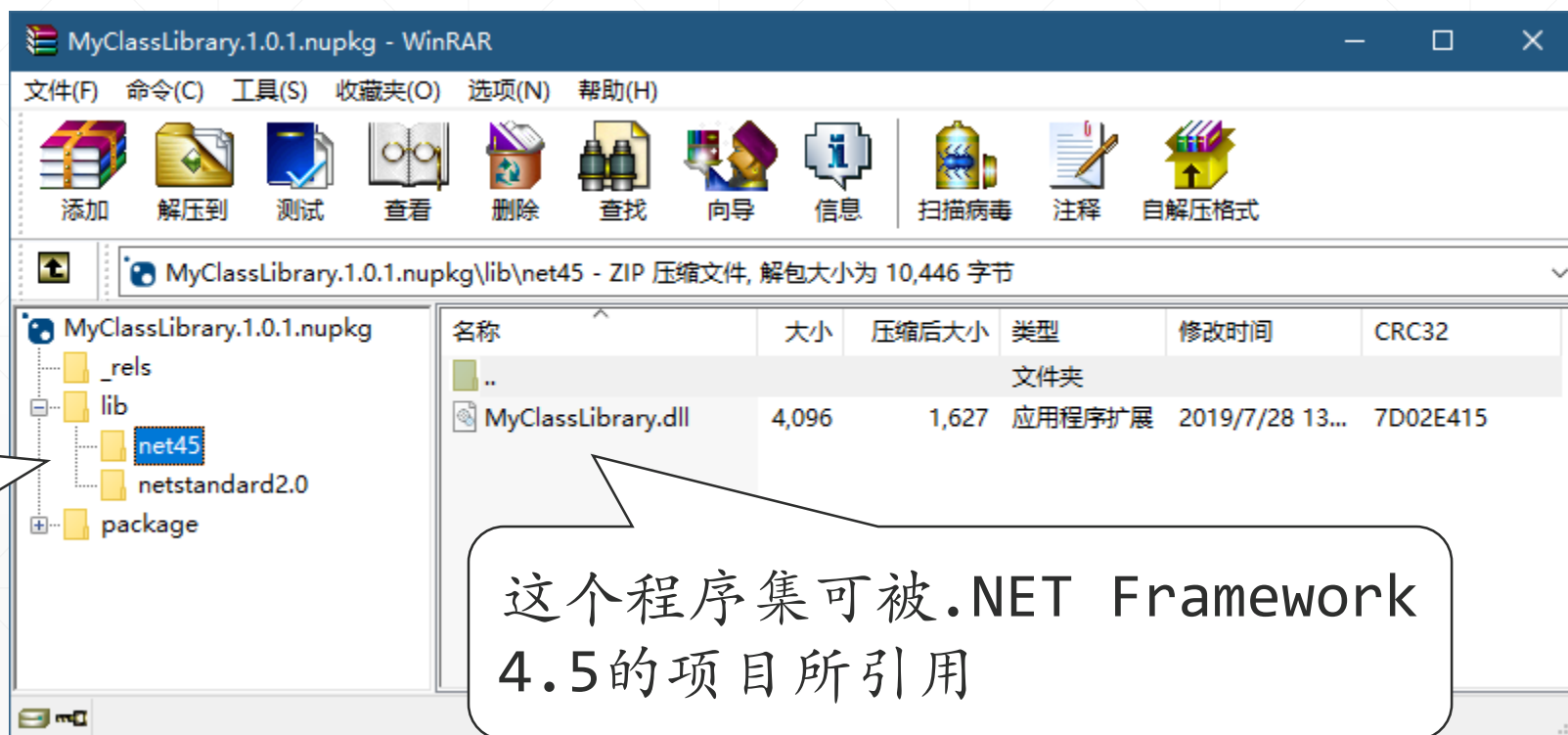
Assembly name:	Default namespace:
<input type="text" value="TraditionalWinForm"/>	<input type="text" value="TraditionalWinForm"/>
Target framework:	Output type:
<input type="text" value=".NET Framework 4.5"/>	<input type="text" value="Windows Application"/>
<input type="checkbox"/> Auto-generate binding redirects	



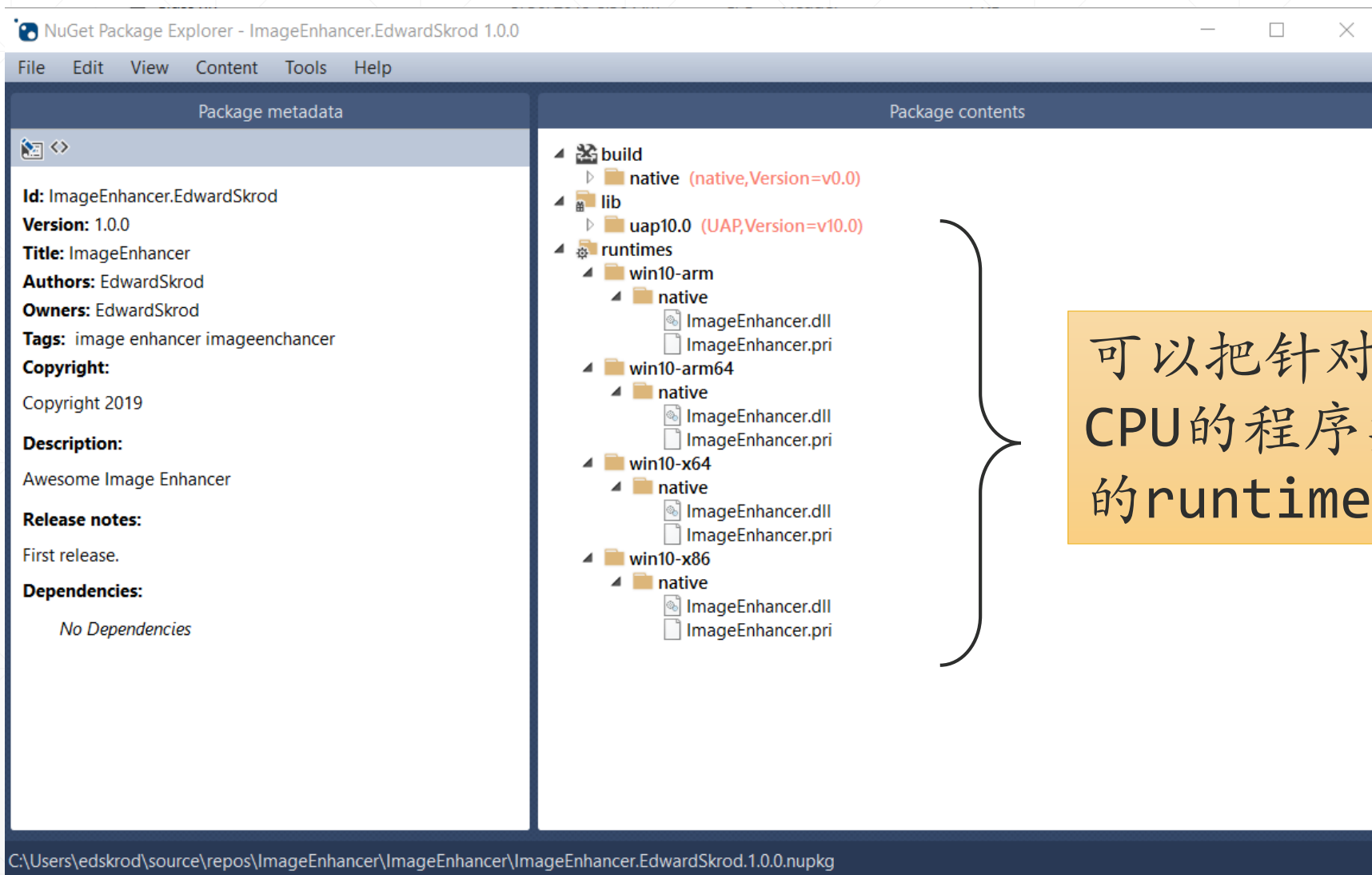
由于MyClassLibrary支持多框架，所以，即使WinForms项目改回使用.NET Framework 4.5作为目标框架，仍然可以调用MyClassLibrary中的组件。

# 新版NuGet包内部有何变化?

查看1.0.1版的NuGet包:



# 其实NuGet包中还支持放置针对特定CPU的程序集.....



可以把针对不同体系结构CPU的程序集放到NuGet包的runtimes文件夹下。







# 使用专门的App来创建包







---

# NuGetPackageExplorer

<https://github.com/NuGetPackageExplorer/NuGetPackageExplorer>






 **NuGetPackageExplorer** / **NuGetPackageExplorer**

 Watch 89  Star 1,267  Fork 246


 Code  Issues 34  Pull requests 0  Wiki  Security  Insights





Create, update and deploy Nuget Packages with a GUI

[nuget-packages](#) [gui-application](#) [nuget-client](#)

 1,519 commits  5 branches  86 releases  39 contributors  MIT

Branch: master ▾ [New pull request](#) [Find File](#) [Clone or download ▾](#)

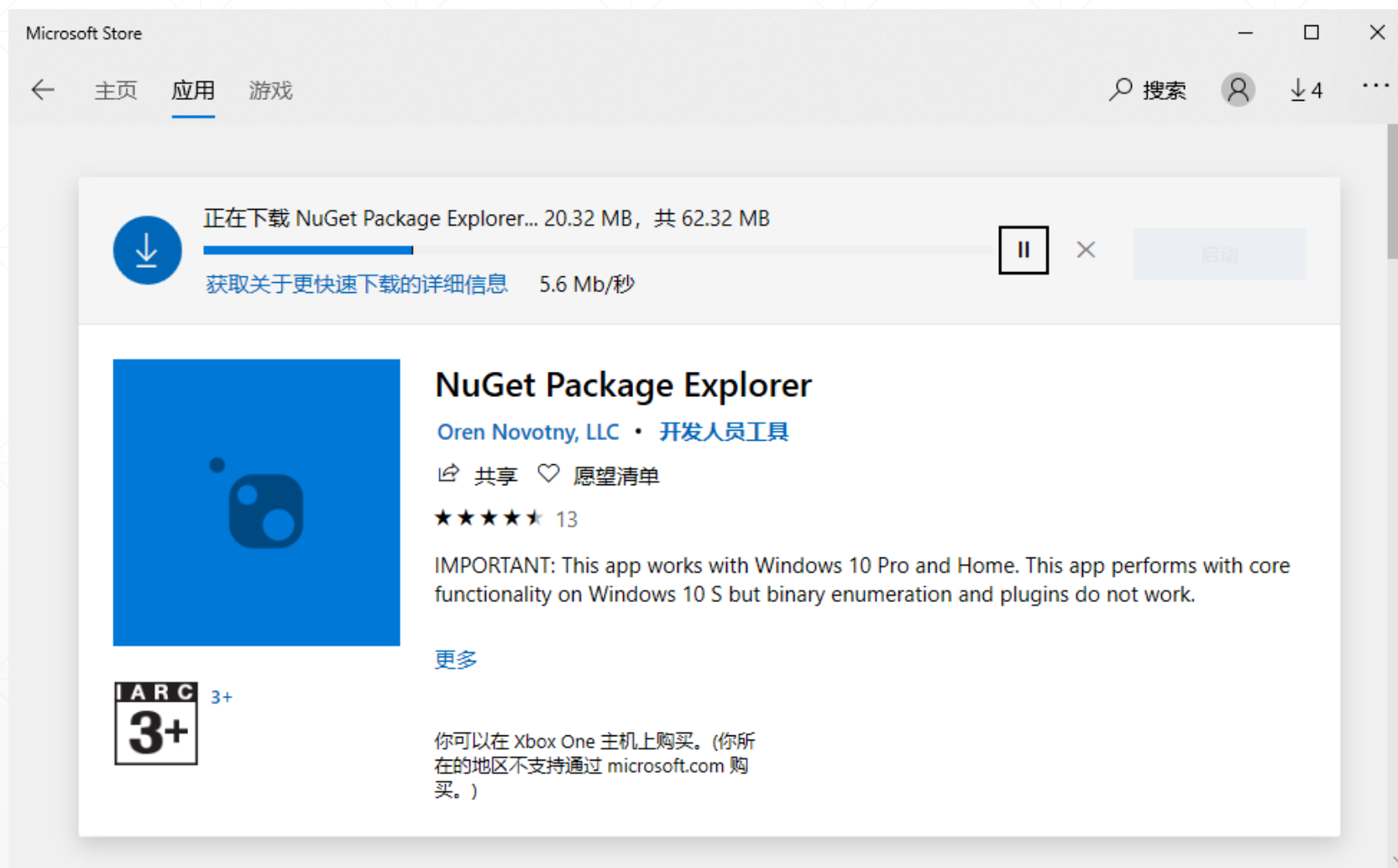
 **onovotny** Update to latest private nuget Latest commit aa478e4 14 days ago

 <a href="#">.github</a>	Update ISSUE_TEMPLATE.md	8 months ago
 <a href="#">Build</a>	Add workaround for updating appinstaller	3 months ago
 <a href="#">Common</a>	Move to NuGet.Packaging for creation and reading of packages (#252)	2 years ago
 <a href="#">Core</a>	supressions	last month

# 可以到Win 10软件商店搜索NuGetPackageExplorer

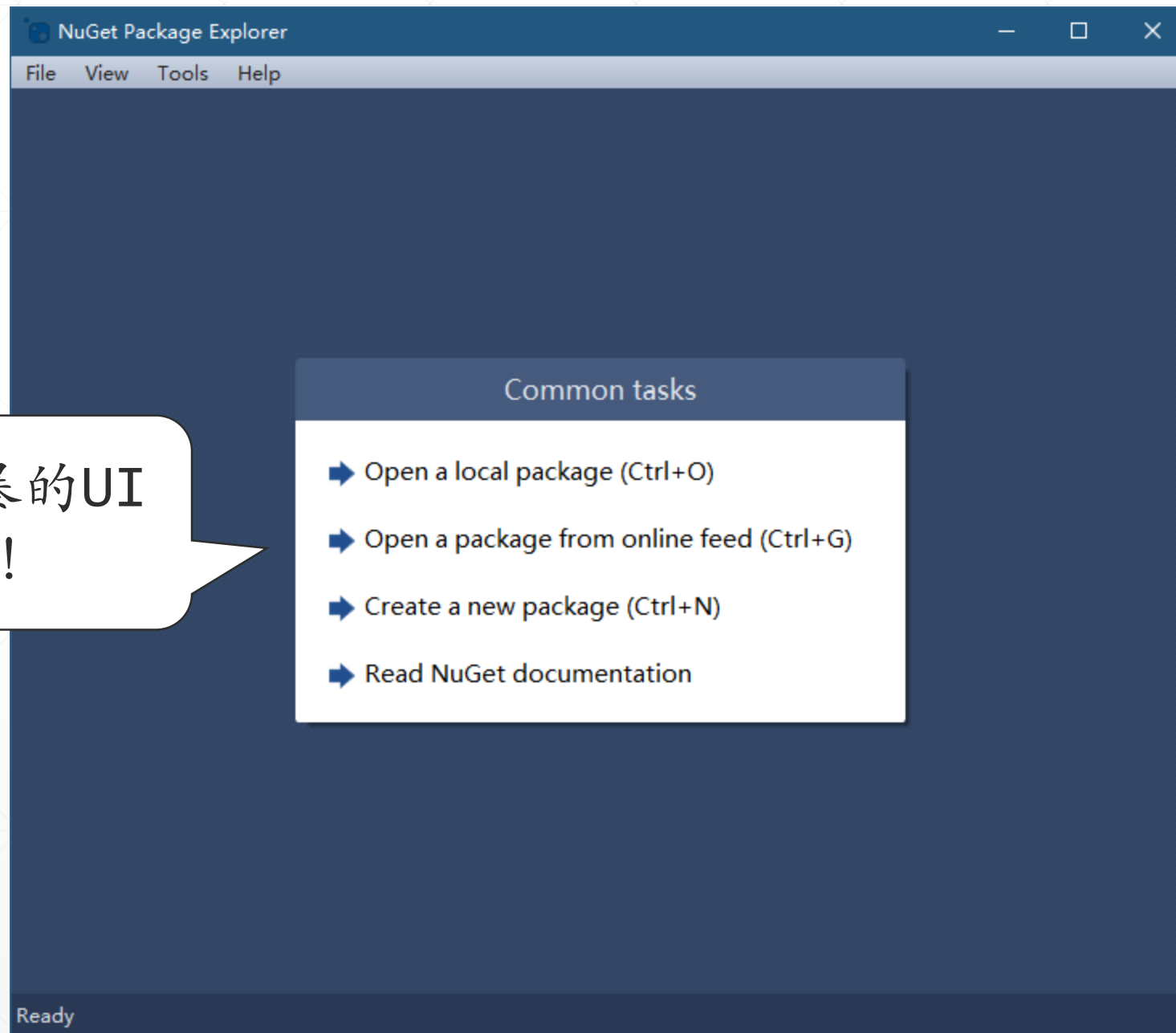


Microsoft Store



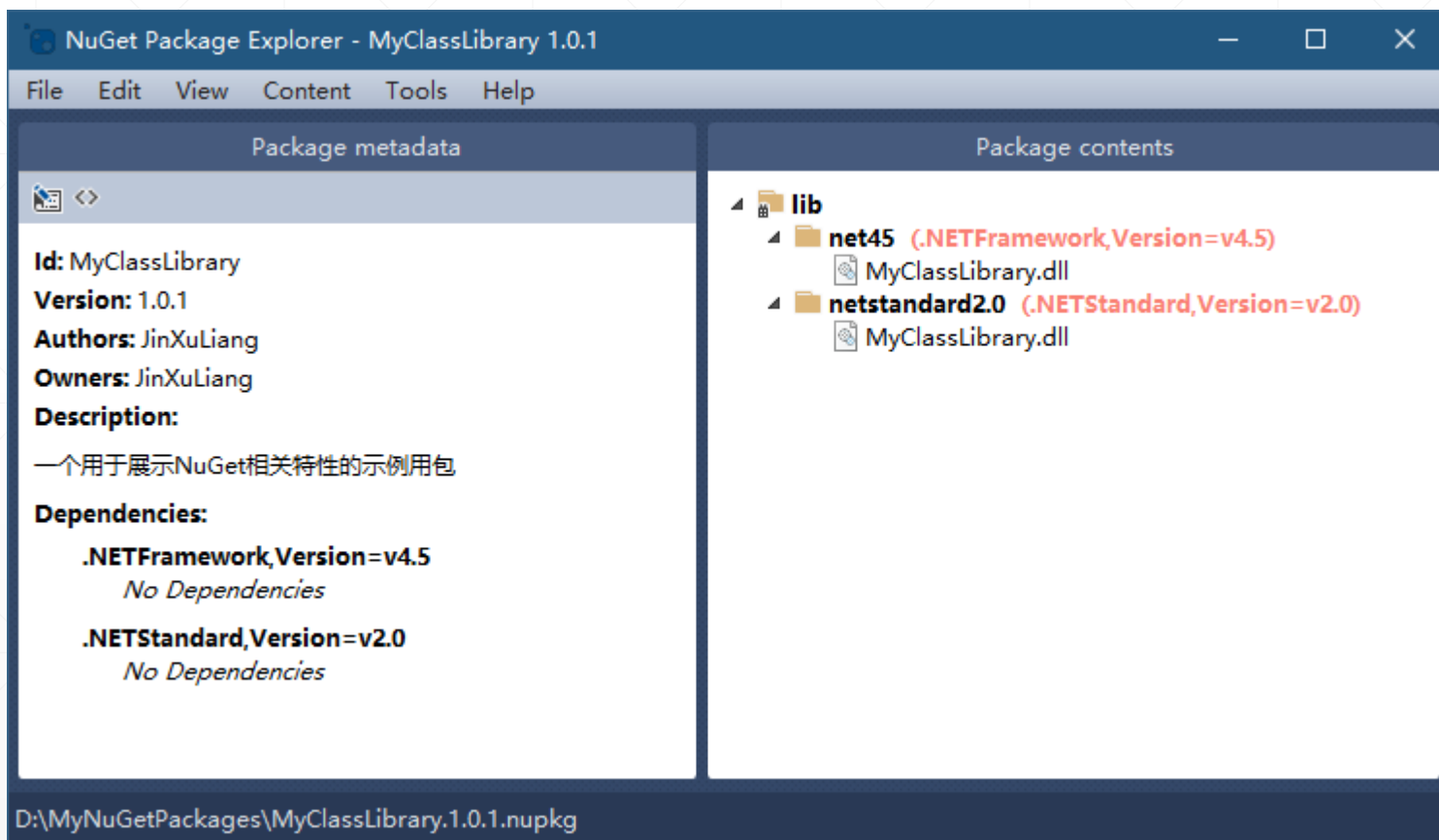


简单粗暴的UI  
界面！



# Open a local package

使用NuGetPackageExplorer查看我们自己开发的NuGet包



# Open a package from online feed

The screenshot shows the 'Select Package' dialog in Visual Studio. The package source is set to 'https://api.nuget.org/v3/index.json'. A list of packages is displayed on the left, with 'Newtonsoft.Json' selected. The right pane shows details for 'Newtonsoft.Json', including its description, version (12.0.2), author (James Newton-King), license, project URL, and tags.

**Select Package**

Search (Ctrl+E) ☐ Show pre-release packages Package source: <https://api.nuget.org/v3/index.json>

Package Icon	Package Name	Author	Downloads	Version
	<b>NUnit</b>	by Charlie Poole, Rob Prouse	47.3M	v3.12.0
	<b>Newtonsoft.Json</b>	by James Newton-King	254.5M	v12.0.2
	<b>EntityFramework</b>	by Microsoft	67.4M	v6.3.0-preview7-19363-02
	<b>MySQL.Data</b>	by Oracle	8.1M	v8.0.17
	<b>NuGet.Core</b>	by Outercurve Foundation	4.8M	v2.14.0
	<b>bootstrap</b>	by The Bootstrap Authors, Twitter Inc.	29.2M	v4.3.1
	<b>Microsoft.Web.Xdt</b>	by Microsoft	3.8M	v3.0.0

**Newtonsoft.Json**

[open](#) [download](#) [show all versions](#)

**Description**  
Json.NET is a popular high-performance JSON framework for .NET

**Version:** 12.0.2

**Author(s):** James Newton-King

**License:** <https://www.nuget.org/packages/Newtonsoft.Json/12.0.2/license>

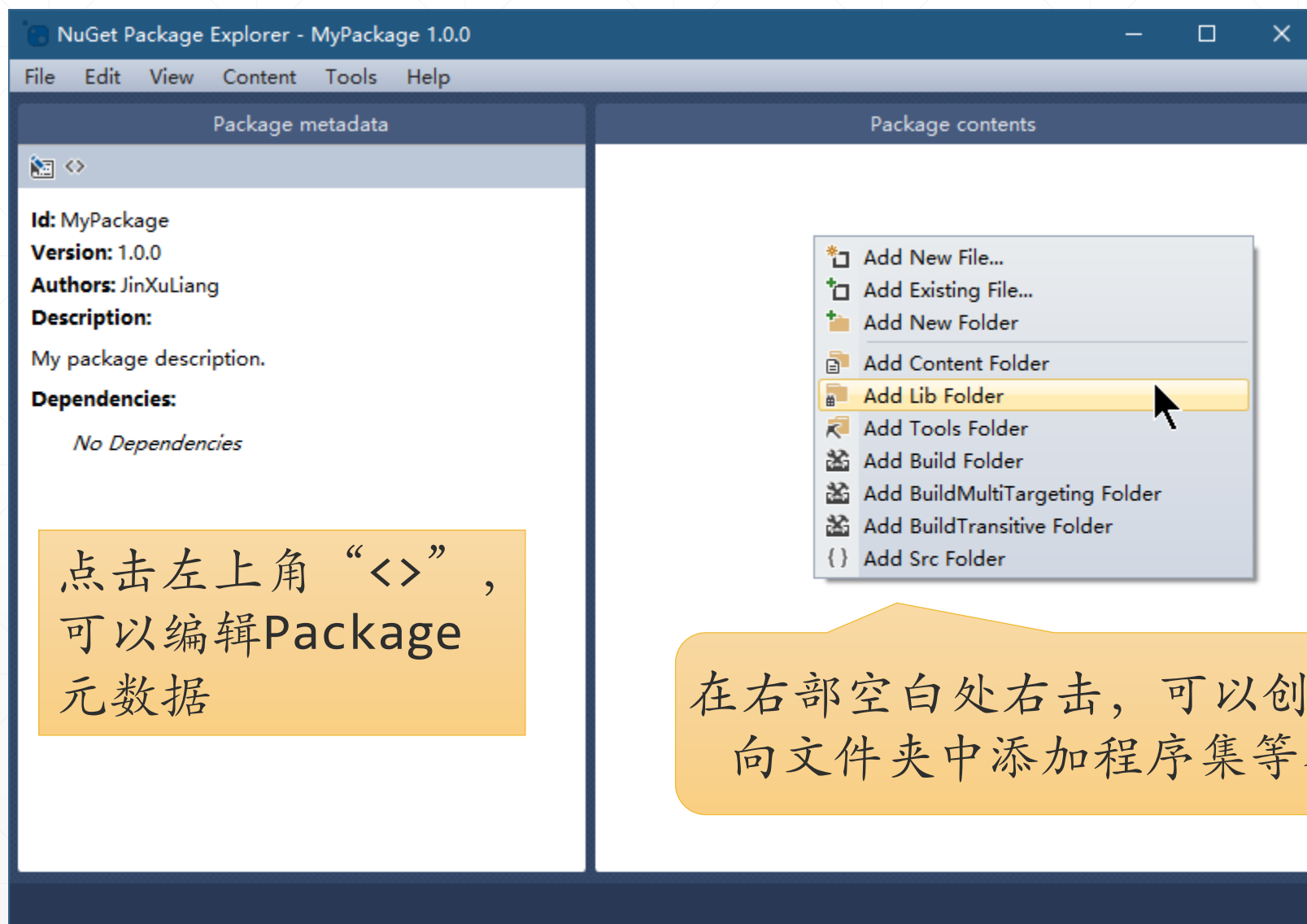
**Project URL:** <https://www.newtonsoft.com/json>

**Report Abuse:**

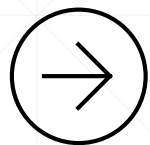
**Tags:** json

<< < 1 to 15 > >>

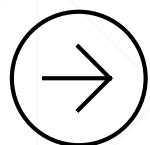
# 使用NuGetPackageExplorer创建NuGet包



# 小结



这一讲PPT包容丰富的内容，请课后在PPT的引导下将示例流程走一遍，以加深理解。



这一讲PPT的重点是介绍了NuGet包的创建方式，以及Target Framework在.NET组件化开发中的重要作用。



NuGet包既支持本地或局域网内的共享，也支持跨互联网的共享，如果你做了有意义的工作，建议将它发布到NuGet.org，为推动.NET生态圈的繁荣尽一把力。

