本篇文章带你了解目标框架，项目，包之间的关系

**.NET 的框架体系**

下表列表了常见的 .net的框架

| **名称** | **TFM（目标框架）/TxM** |
| --- | --- |
| .NET framework | net11 |
|  | net20 |
|  | ...(一系列的版本) |
|  | net472 |
|  | net48 |
| Windows Phone (SL) | wp [wp7] |
|  | ...(一系列的版本) |
|  | wp81 |
| .NET Standard | netstandard1.0 |
|  | netstandard1.1 |
|  | ...(一系列的版本) |
|  | netstandard2.0 |
|  | netstandard2.1 |
| .NET 5 及更高版本（和 .NET Core） | netcoreapp1.0 |
|  | netcoreapp1.1 |
|  | ...(一系列的版本) |
|  | netcoreapp3.1 |
|  | net5.0 |
|  | net6.0 |

1. .NET Standard

.NET Standard 为规范框架，基于该框架的项目（包）可以被其他框架项目（包）使用，如项目A的框架为net6.0，包B的框架为netstandard2.1，则项目A可以引用包B

1. 版本兼容性

高版本项目（包）可以引用低版本项目（包），如项目A的框架为net6.0，包B的框架为netcoreapp2.0，则A可以引用B，反之则不行

1. NET framework的兼容

从 .NET Standard 2.0 开始，引入了 .NET Framework 兼容性模式。 此兼容性模式允许 .NET Standard 和 .NET Core 项目引用 .NET Framework 库。但本人不推荐这种做法

**.NET Standard与其他框架的关系**

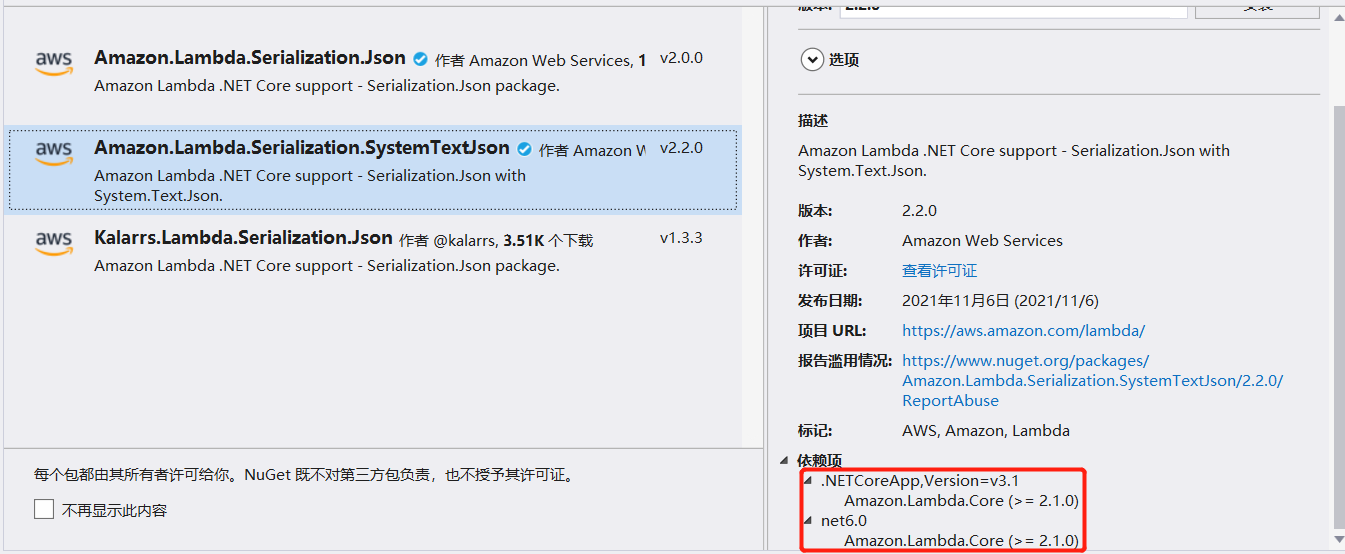
| .NET Standard | [1.0](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard1.0.md) | [1.1](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard1.1.md) | [1.2](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard1.2.md) | [1.3](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard1.3.md) | [1.4](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard1.4.md) | [1.5](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard1.5.md) | [1.6](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard1.6.md) | [2.0](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard2.0.md) | [2.1](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard2.1.md) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .NET 和.NET Core | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.0 | 3.0 |
| .NET Framework 1 | 4.5 | 4.5 | 4.5.1 | 4.6 | 4.6.1 | 4.6.1 2 | 4.6.1 2 | 4.6.1 2 | N/A3 |

举个例子，如果项目（包）B的目标框架为 .NET Standard 2.0，那么要想引用B，那么当前项目的目标框架必须大于或等于 .net core 2.0或 .NET Framework 4.6.1

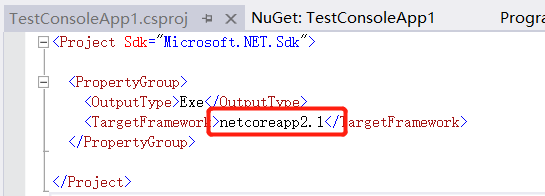
**Nuget包与框架**

每个包都有其支持的目标框架

如下，Amazon.Lambda.Serialization.SystemTextJson包支持的目标框架有 .netcoreapp3.1 和 net6.0

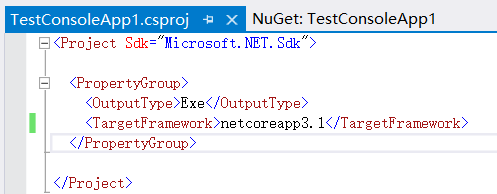


如果我们项目的目标框架是netcoreapp2.1，那么我们无法安装该包



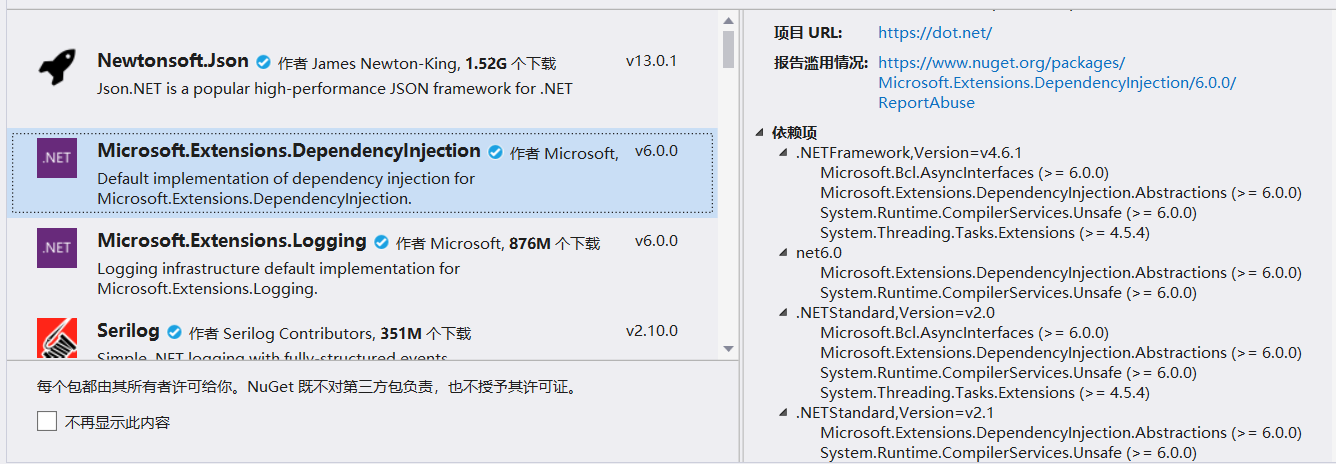


如果我们将项目的目标框架改为netcoreapp3.1，则可以正常安装



**Nuget包依赖项分析**

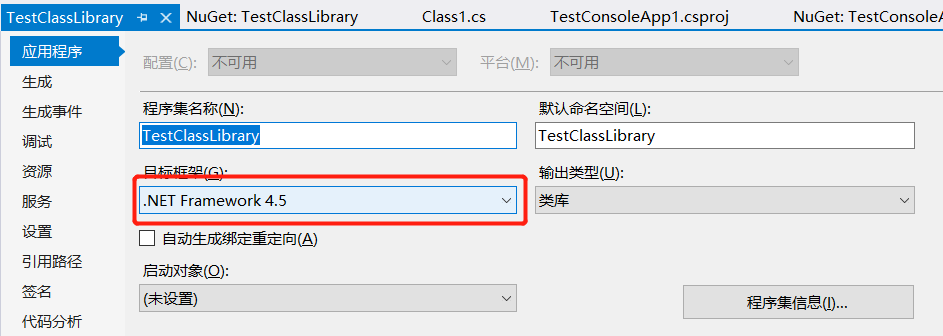
我们以Microsoft.Extensions.DependencyInjection包的依赖项进行分析



我们以下面几个项目进行分析

1. 项目目标框架 .NET Framework 4.5

如果我们的项目是 .NET Framework 4.5，对应的 .NET Standard 为1.0



依赖项分析：

包依赖项 .NET Framework 4.6.1，该选项大于项目的目标框架，不适合

包依赖项 .NET Standard 2.0，该选项大于项目的 .NET Standard 1.0，不适合

...(其他选项不说了)

所以项目无法安装该包

1. 项目目标框架 .netcoreapp2.1

如果我们的项目是 .netcoreapp2.1，对应的 .NET Standard 为2.0

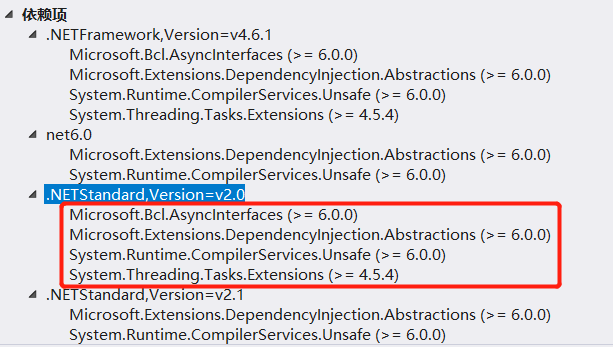
依赖项分析：

包依赖项 .net6.0，该选项不适合

包依赖项 .NET Standard 2.1，该选项不适合

包依赖项 .NET Standard 2.0，该选项适合

所以项目会安装该包，并且安装该包在该框架下所依赖的4个包



1. 项目目标框架 net5.0

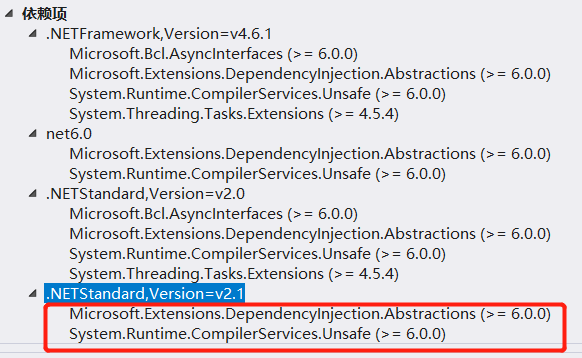
如果我们的项目是 net5.0，对应的 .NET Standard 为2.1

依赖项分析：

包依赖项 .net6.0，该选项不适合

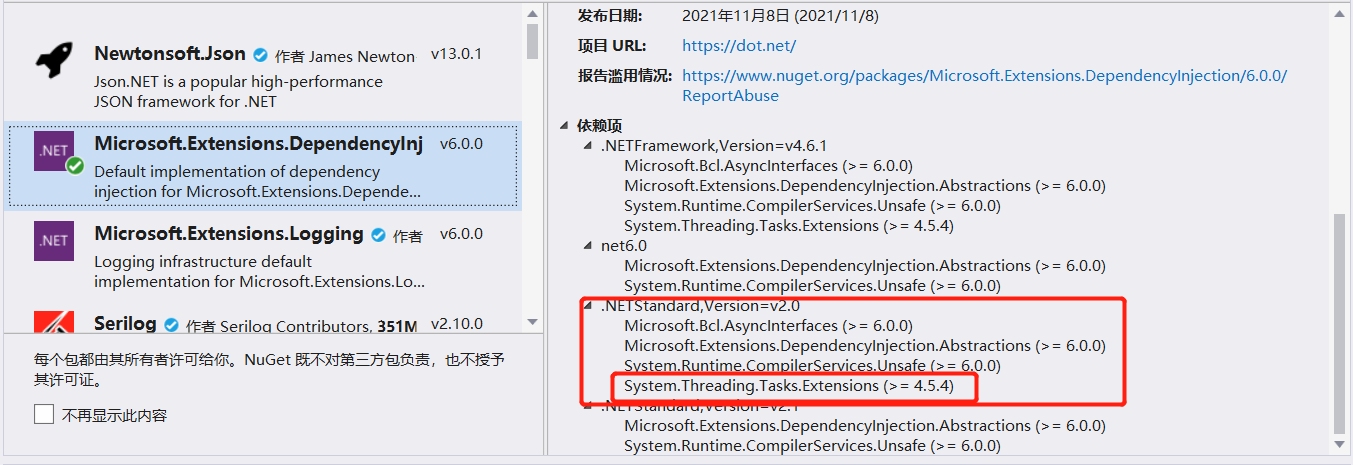
包依赖项 .NET Standard 2.1，该选项适合

所以项目会安装该包，并且安装该包在该框架下所依赖的2个包

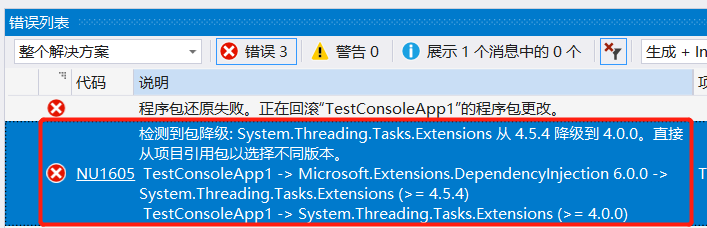


**包版本冲突**

如下，我们在 .netcoreapp2.1 下安装了包Microsoft.Extensions.DependencyInjection



现在我们安装System.Threading.Tasks.Extensions 4.0 版本，会收到错误提示



意思就是下面这2个依赖包有冲突

TestConsoleApp1 -> Microsoft.Extensions.DependencyInjection 6.0.0 -> System.Threading.Tasks.Extensions (>= 4.5.4)

TestConsoleApp1 -> System.Threading.Tasks.Extensions (>= 4.0.0)