

# 在 Visual Studio 中生成 .NET Standard 库

2019/12/09 • 🐼 🐼 🐼

## 本文内容

[创建 Visual Studio 解决方案](#)

[创建类库项目](#)

[后续步骤](#)

类库定义的是可以由应用程序调用的类型和方法。借助面向 .NET Standard 2.0 的类库，任何支持相应 .NET Standard 版本的 .NET 实现都可以调用库。完成类库时，可以决定是要将其作为第三方组件进行分布，还是要将其作为与一个或多个应用程序捆绑在一起的组件进行添加。

### ❗ 备注

有关 .NET Standard 版本及其支持的平台列表，请参阅 [.NET Standard](#)。

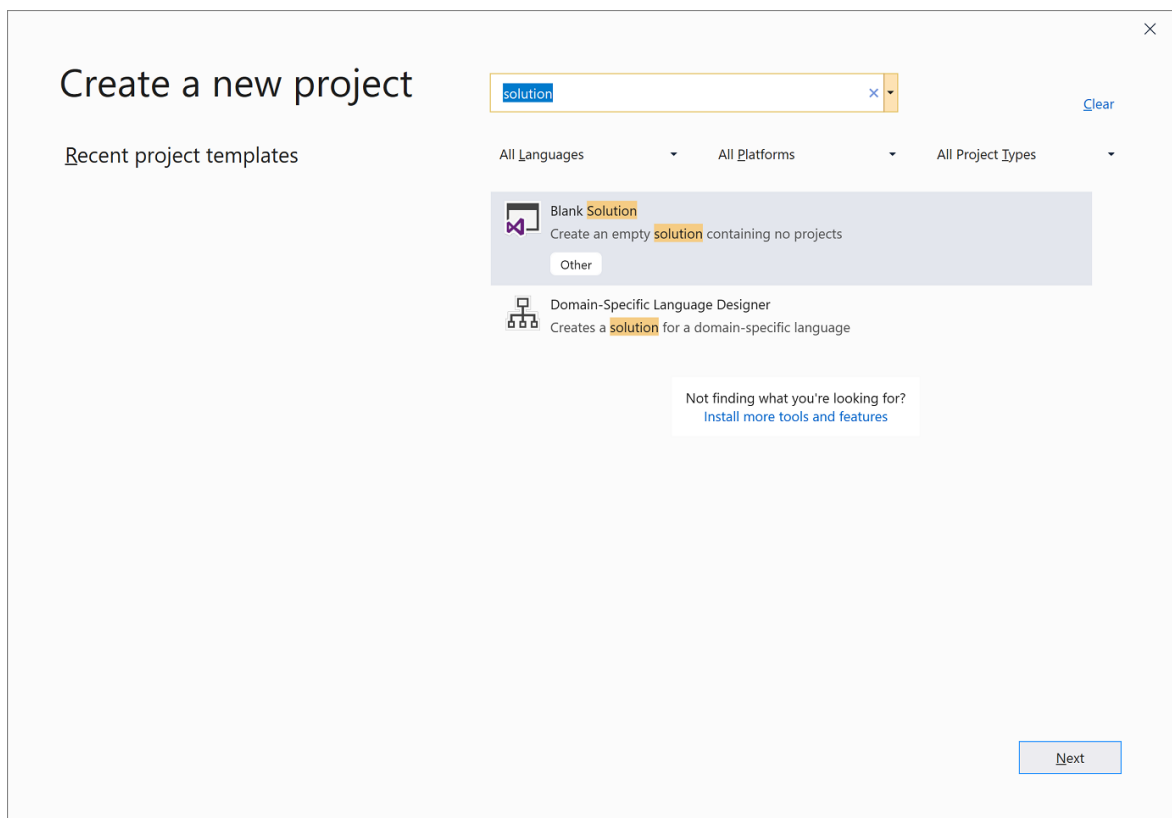
在本主题中，将创建包含一个字符串处理方法的简单实用工具库。我们将把它作为[扩展方法](#)进行实现，这样就可以把它作为 [String](#) 类成员进行调用。

## 创建 Visual Studio 解决方案

首先，创建一个空白解决方案来放置类库项目。Visual Studio 解决方案用作一个或多个项目的容器。如果继续学习本系列教程，你将向相同的解决方案添加其他相关项目。

创建空白解决方案：

1. 打开 Visual Studio。
2. 在“开始”窗口上，选择“创建新项目”。
3. 在“创建新项目”页面上，在搜索框中输入“解决方案”。选择“空白解决方案”模板，然后选择“下一步”。



4. 在“配置新项目”页面上，在“项目名称”框中输入“ClassLibraryProjects”。然后，选择“创建”。

### 💡 提示

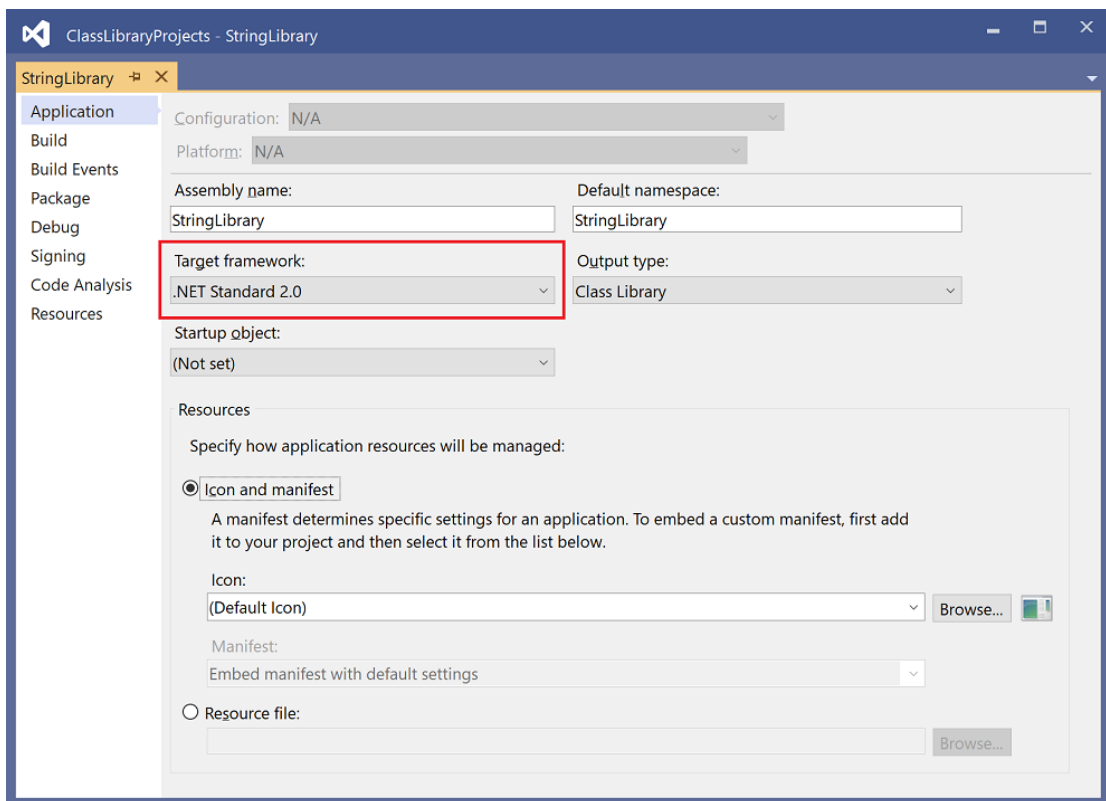
你还可以跳过此步骤，并让 Visual Studio 在下一步中创建项目时为你创建该解决方案。在“配置新项目”页上查找解决方案选项。

## 创建类库项目

C# Visual Basic

1. 将名为“StringLibrary”的新 C# .NET Standard 类库项目添加到解决方案。
  - a. 在“解决方案资源管理器”中右键单击解决方案并选择“添加” > “新建项目”。
  - b. 在“创建新项目”页面上，在搜索框中输入“库”。从“语言”列表中选择“C#”，然后从“平台”列表中选择“所有平台”。选择“类库 (.NET Standard)”模板，然后选择“下一步”。
  - c. 在“配置新项目”页面，在“项目名称”框中输入“StringLibrary”。然后，选择“创建”。

2. 请检查以确保库面向 .NET Standard 的正确版本。右键单击“解决方案资源管理器”中的库项目，然后选择“属性”。“目标框架”文本框显示项目面向 .NET Standard 2.0。



3. 将代码窗口中的代码替换为以下代码，并保存文件：

```
C# 复制

using System;

namespace UtilityLibraries
{
    public static class StringLibrary
    {
        public static bool StartsWithUpper(this String str)
        {
            if (String.IsNullOrEmpty(str))
                return false;

            Char ch = str[0];
            return Char.IsUpper(ch);
        }
    }
}
```

类库 `UtilityLibraries.StringLibrary` 包含一个名为 `StartsWithUpper` 的方法。此方法会返回 [Boolean](#) 值，以指明当前字符串实例是否以大写字符开头。Unicode 标准会区分大小写字符。如果为大写字符，[Char.IsUpper\(Char\)](#) 方法返回 `true`。

4. 在菜单栏中，选择“生成” > “生成解决方案”。

此项目的编译应该没有错误。

## 后续步骤

已成功生成库。由于尚未调用库的任何方法，因此还不知道它能否按预期运行。开发库的下一步是测试库。

创建单元测试项目

---

此页面有帮助吗？

 是  否

---