以应用或库中的框架为目标时，需要指定想要向应用或库提供的 API 集。 使用目标框架名字对象 (TFM) 在项目文件中指定目标框架。

应用或库可以使用 .NET Standard 版本作为目标。 .NET Standard 版本表示所有 .NET 实现中的标准化 API 集。 例如，库可以使用 .NET Standard 1.6 作为目标，并获得对可使用相同基本代码跨 .NET Core 和 .NET Framework 工作的 API 的访问权限。

应用或库还能以一个特定 .NET 实现为目标，获得特定于实现的 API 的访问权限。 例如，面向 Xamarin.iOS 的应用（如 `Xamarin.iOS10`）有权访问 Xamarin 提供的适用于 iOS 10 的 iOS API 包装器；面向通用 Windows 平台 (UWP) 的应用（如 `uap10.0`）有权访问为运行 Windows 10 的设备编译的 API。

对于某些目标框架（例如 .NET Framework），API 由框架在系统上安装的程序集定义，并且可能包括应用程序框架 API（例如 ASP.NET）。

对于基于包的目标框架（例如 .NET 5、.NET Core 和 .NET Standard），API 由包含在应用或库中的包定义。 元包 是一个 NuGet 包，NuGet 包本身不包含任何内容，只是一个依赖项列表（其他包）。 基于 NuGet 包的目标框架隐式指定一个元包，该元包引用一起构成框架的所有包。

最新版本

下表定义了最常见的目标框架、如何引用这些框架，以及它们实现的 .NET Standard 版本。 这些目标框架版本是最新的稳定版本。 预览版不会显示。 目标框架名字对象 (TFM) 是一个标准化令牌格式，用于指定 .NET 应用或库的目标框架。

目标框架	最新 稳定版本	目标框架名字对象 (TFM)	已实现 .NET Standard 版本
.NET 5	5.0	net5.0	空值
.NET Standard	2.1	netstandard2.1	空值
.NET Core	3.1	netcoreapp3.1	2.1
.NET Framework	4.8	net48	2.0

支持的目标框架

目标框架通常由 TFM 引用。 下表显示 .NET SDK 和 NuGet 客户端支持的目标框架。 等效项显示在括号内。 例如，`win81` 对于 `netcore451` 来说等效于 TFM。

目标 Framework	TFM
.NET 5（和 .NET Core）	netcoreapp1.0 netcoreapp1.1 netcoreapp2.0 netcoreapp2.1 netcoreapp2.2 netcoreapp3.0 netcoreapp3.1 net5.0*
.NET Standard	netstandard1.0

	netstandard1.1 netstandard1.2 netstandard1.3 netstandard1.4 netstandard1.5 netstandard1.6 netstandard2.0 netstandard2.1
.NET Framework	net11 net20 net35 net40 net403 net45 net451 net452 net46 net461 net462 net47 net471 net472 net48
Windows 应用商店	netcore [netcore45] netcore45 [win] [win8] netcore451 [win81]
.NET Micro Framework	netmf
Silverlight	sl4 sl5
Windows Phone	wp [wp7] wp7 wp75 wp8 wp81 wpa81
通用 Windows 平台	uap [uap10.0] uap10.0 [win10] [netcore50]

* .NET 5.0 及更高版本的 TFM 包含特定于操作系统的变体。 有关详细信息，请参阅下一节：.NET 5 特定于 OS 的 TFM。

.NET 5 特定于 OS 的 TFM

对于每个 .NET 5.0 及更高版本的 TFM（例如 `net5.0`），都存在包含特定于 OS 的绑定的 TFM 变体。 下表中显示了这些变体。

特定于 OS 的格式	示例
<base-tfm>-android	net5.0-android
<base-tfm>-ios	net5.0-ios
<base-tfm>-macos	net5.0-macos

<base-tfm>-tvos	net5.0-tvos
<base-tfm>-watchos	net5.0-watchos
<base-tfm>-windows	net5.0-windows

`net5.0` TFM 仅包括跨平台工作的技术。 指定特定于 OS 的 TFM 使特定于操作系统的 API 可供应用使用，例如 Windows 窗体或 iOS 绑定。 特定于 OS 的 TFM 还继承 `net5.0` TFM 可用的每个 API。
若要使应用可跨不同平台移植，你可以定位多个特定于 OS 的 TFM，并使用 `#if` 预处理器指令围绕特定于 OS 的 API 调用添加平台保护。
下表显示了 .NET 5 TFM 与旧 .NET 版本的 TFM 的兼容性。

TFM	可兼容对象	备注
net5.0	net1..4（带有 NU1701 警告） netcoreapp1..3.1 （引用 WinForms 或 WPF 时出现警告） netstandard1..2.1	
net5.0-android	xamarin.android（以及从 `net5.0` 继承的所有其他内容）	
net5.0-ios	xamarin.ios（以及从 `net5.0` 继承的所有其他内容）	
net5.0-macos	xamarin.mac（以及从 `net5.0` 继承的所有其他内容）	
net5.0-tvos	xamarin.tvos（以及从 `net5.0` 继承的所有其他内容）	
net5.0-watchos	xamarin.watchos（以及从 `net5.0` 继承的所有其他内容）	
net5.0-windows	netcoreapp1..3.1（以及从 `net5.0` 继承的所有其他内容）	包括 WinForms、WPF 和 UWP API。 有关信息，请参阅在桌面应用中调用 Windows 运行时 API。

建议的目标
使用以下准则确定在应用中使用哪种 TFM：

- 可移植到多个平台的应用应面向 `net5.0`。 这包括大多数库，但也包含 ASP.NET Core 和实体框架。
- 特定于平台的库应面向特定于平台的风格。 例如，WinForms 和 WPF 项目应面向 `net5.0-windows`。
- 跨平台应用程序模型（Xamarin Forms、ASP.NET Core）和网桥包 (Xamarin Essentials) 应至少面向 `net5.0`，但也可以面向其他特定于平台的风格来支持更多的 API 或功能。

TFM 中的 OS 版本

你还可以在 TFM 的末尾指定可选的 OS 版本，例如 net5.0-ios13.0，该版本指示应用可用的 API。 （.NET 5 SDK 将更新，以在发布时包含对较新 OS 版本的支持。）若要访问新发布的 API，请增加 TFM 中的 OS 版本。 通过将 SupportedOSPlatformVersion 元素添加到项目文件，仍可以使应用与旧 OS 版本兼容（并在调用更高版本的 API 时添加防护）。 SupportedOSPlatformVersion 元素指示运行应用所需的最低 OS 版本。

例如，以下项目文件摘录指定 iOS 14 API 可用于应用，但它可以在 iOS 13 或更高版本的计算机上运行。

XML

```
<Project Sdk="Microsoft.NET.Sdk"> <PropertyGroup> <TargetFramework>net5.0-ios14.0</TargetFramework> <SupportedOSPlatformVersion>13.0</SupportedOSPlatformVersion> (minimum os platform version) </PropertyGroup> ... </Project>
```

如何指定目标框架

在项目文件中指定目标框架。 指定单个目标框架时，使用 TargetFramework 元素。 以下控制台应用项目文件演示了如何面向 .NET 5.0：

XML

```
<Project Sdk="Microsoft.NET.Sdk"> <PropertyGroup> <OutputType>Exe</OutputType> <TargetFramework>net5.0</TargetFramework> </PropertyGroup> </Project>
```

指定多个目标框架时，可有条件地为每个目标框架引用程序集。 在代码中，可使用具有 -if-then-else 逻辑的预处理器符号，有条件地针对这些程序集进行编译。

以下库项目面向 .NET Standard (netstandard1.4) 和 .NET Framework （net40 和 net45）的 API。 将复数形式的 TargetFrameworks 元素与多个目标框架一起使用。 为两个 .NET Framework TFM 编译库时，Condition 属性包括特定于实现的包：

XML

```
<Project Sdk="Microsoft.NET.Sdk"> <PropertyGroup> <TargetFrameworks>netstandard1.4;net40;net45</TargetFrameworks> </PropertyGroup> <!-- Conditionally obtain references for the .NET Framework 4.0 target --> <ItemGroup Condition=" '$(TargetFramework)' == 'net40' "> <Reference Include="System.Net" /> </ItemGroup> <!-- Conditionally obtain references for the .NET Framework 4.5 target --> <ItemGroup Condition=" '$(TargetFramework)' == 'net45' "> <Reference Include="System.Net.Http" /> <Reference Include="System.Threading.Tasks" /> </ItemGroup> </Project>
```

在库或应用中，使用预处理器指令编写条件代码，针对每个目标框架进行编译：

C#

```
public class MyClass { static void Main() { #if NET40 Console.WriteLine("Target framework: .NET Framework 4.0"); #elif NET45 Console.WriteLine("Target framework: .NET Framework 4.5"); #else Console.WriteLine("Target framework: .NET Standard 1.4"); #endif } }
```

使用 SDK 样式项目时，生成系统可识别预处理器符号，这些符号表示支持的目标框架版本表中所示的目标框架。 使用表示 .NET Standard、.NET Core 或 .NET 5 TFM 的符号时，请用下划线替换点和连字符，并将小写字母更改为大写字母（例如，netstandard1.4 的符号为 NETSTANDARD1_4）。

.NET 目标框架的预处理器符号的完整列表如下：

目标框架	符号
.NET Framework	NETFRAMEWORK, NET48, NET472, NET471, NET47, NET462, NET461, NET46, NET452, NET451, NET45, NET40, NET35, NET20
.NET Standard	NETSTANDARD, NETSTANDARD2_1, NETSTANDARD2_0, NETSTANDARD1_6, NETSTANDARD1_5, NETSTANDARD1_4, NETSTANDARD1_3, NETSTANDARD1_2, NETSTANDARD1_1, NETSTANDARD1_0
.NET 5（和 .NET Core）	NET5_0, NETCOREAPP, NETCOREAPP3_1, NETCOREAPP3_0, NETCOREAPP2_2, NETCOREAPP2_1, NETCOREAPP2_0, NETCOREAPP1_1, NETCOREAPP1_0

已弃用的目标框架

以下目标框架已弃用。 面向这些目标框架的包应迁移到指定的替代框架。

已弃用的 TFM	Replacement
aspnet50 aspnetcore50 dnxcore50 dnx dnx45 dnx451 dnx452	netcoreapp
dotnet dotnet50 dotnet51 dotnet52 dotnet53 dotnet54 dotnet55 dotnet56	netstandard
netcore50	uap10.0
win	netcore45
win8	netcore45
win81	netcore451
win10	uap10.0
winrt	netcore45

另请参阅

- .NET 5 中的目标框架名称
- 使用跨平台工具开发库
- .NET Standard
- .NET Core 版本控制
- dotnet/standard GitHub 存储库
- NuGet 工具 GitHub 存储库
- .NET 中的框架配置文件
- 平台兼容性分析器