# 如何从Toolkit5.5升级到Toolkit5.6

Toolkit5.5是基于.net standard2.0和.net core2.2的。由于.net core2.2的生命周期临近，因此，Toolkit5.6基于.net standard2.1和.net core3.1用于取代Toolkit5.5。

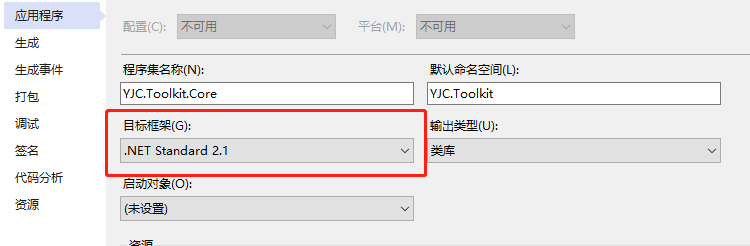
原则上，Toolkit5.5和Toolkit5.6没有太大差别。下面介绍升级方法。

## .net standard库升级

### 目标修改

直接修改项目文件，将csproj中的<TargetFramework>netstandard2.0</TargetFramework>修改为<TargetFramework>netstandard2.1</TargetFramework>

或者在项目属性中修改目标框架，如下图：



### 多余引用删除

如果项目中，因为Linq的问题，引用了如下包：

<PackageReference Include="System.Data.DataSetExtensions" Version="4.5.0" />

那么可以直接删除，因为.net standard2.1已经包含该库了。

### 更新输出路径

由于Web项目取消了.net core版本的输出路径，所以，原先的输出路径 ..\..\ManageWeb\bin\Debug\netcoreapp2.2\Modules\ 需要改为 ..\..\ManageWeb\bin\Debug\Modules\。

在csproj文件如下：

<PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|AnyCPU'">

<OutputPath>..\..\ManageWeb\bin\Debug\Modules\</OutputPath>

</PropertyGroup>

### 更新Toolkit包

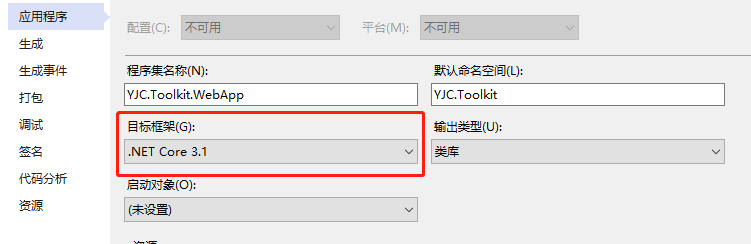
使用update-package更新Toolkit包到5.6.X的版本，然后编译即可。

## .net core库升级

### 目标修改

直接修改项目文件，将csproj中的<TargetFramework>netcoreapp2.2</TargetFramework>修改为<TargetFramework>netcoreapp3.1</TargetFramework>

或者在项目属性中修改目标框架，如下图：



### 多余引用删除

如果项目中，因为Linq的问题，引用了如下包：

<PackageReference Include="System.Data.DataSetExtensions" Version="4.5.0" />

那么可以直接删除，因为.net core3.1已经包含该库了。

此外，如果使用字体引用了

<PackageReference Include="System.Drawing.Common" Version="4.7.0" />

那么也可以直接删除，因为.net core3.1已经包含该库了。

### AspNetCore引用修改

如果.net core项目要使用Web的Request，Response之类的类，原来的方式是使用：

<PackageReference Include="Microsoft.AspNetCore" Version="2.2.0" />

现在需要改为：

<ItemGroup>

<FrameworkReference Include="Microsoft.AspNetCore.App" />

</ItemGroup>

但是，Web项目不需要引用，因为自身就内含该模块（如果有上面的内容，编译时应该会生成警告）。

### 更新输出路径

由于Web项目取消了.net core版本的输出路径，所以，原先的输出路径 ..\..\ManageWeb\bin\Debug\netcoreapp2.2\Modules\ 需要改为 ..\..\ManageWeb\bin\Debug\Modules\。

在csproj文件如下：

<PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|AnyCPU'">

<OutputPath>..\..\ManageWeb\bin\Debug\Modules\</OutputPath>

</PropertyGroup>

### 更新Toolkit包

使用update-package更新Toolkit包到5.6.X的版本，然后编译即可。

## Web项目升级

### 目标库修改/多余引用删除/更新Toolkit包

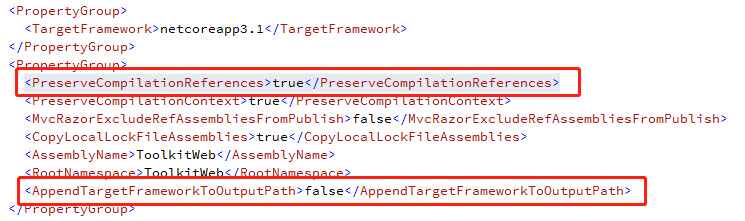
首先，Web项目也是.net core库，需要按照.net core库的升级方法升级。

### Razor条件修改

然后，在项目中，因为Razor库的更新，我们需要增加<PreserveCompilationReferences>true</PreserveCompilationReferences>，否则，运行Razor时会出错。

此外，为了避免产生netcore2.2或者netcore3.1之类的目录，在项目中可以增加<AppendTargetFrameworkToOutputPath>false</AppendTargetFrameworkToOutputPath>。

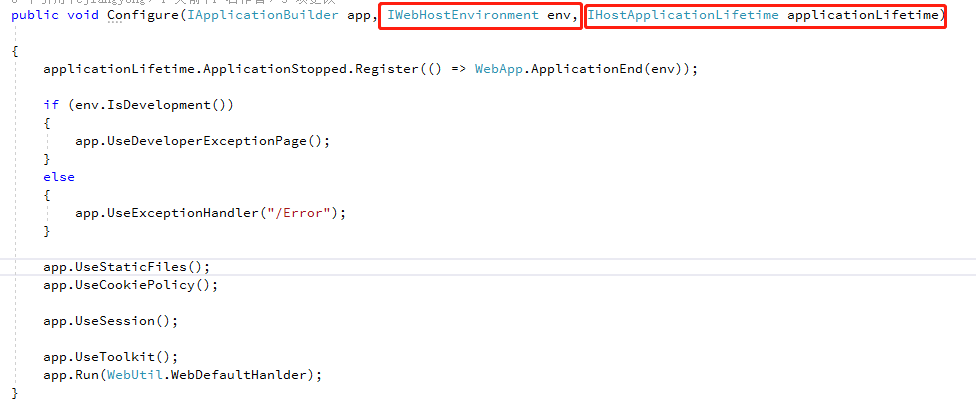
下图是Toolkit5.6的项目文件截图，注意红框的内容，是新增的：



### Startup.cs修改

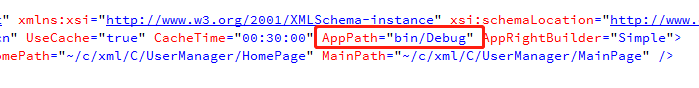
由于IHostingEnvironment和IApplicationLifetime在.net core3.1中过期。因此，根据注释提示，Toolkit5.6修订了相应的API，使用IWebHostEnvironment来替代IHostingEnvironment，使用IHostApplicationLifetime来替代IApplicationLifetime。

因此，Startup.cs中的Config函数需要修改一下，如下图：



### Application.xml修改

由于编译生成不再生成netcore22或者netcore31之类的目录。因此Application.xml中的AppPath需要调整，调整如下图：



### 删除Razor的Temp文件

由于Toolkit的库全面升级，以及Razor库做了一些变动。保险起见，清空Xml/Razor/Temp下全部文件，在运行时，Toolkit会根据需要重新生成相应的razor文件。