**[.NET Core全面扫盲贴](http://www.cnblogs.com/Wddpct/p/5694596.html)**

标签： .NETCore Asp.NETCore

* [1. 前言](http://www.cnblogs.com/Wddpct/p/5694596.html#1.)
* [2. .NET Core 简介](http://www.cnblogs.com/Wddpct/p/5694596.html#2.)
  + [2.1 .NET Core是什么](http://www.cnblogs.com/Wddpct/p/5694596.html#2.1)
  + [2.2 .NET Core的组成](http://www.cnblogs.com/Wddpct/p/5694596.html#2.2)
  + [2.3 .NET Core的特性](http://www.cnblogs.com/Wddpct/p/5694596.html#2.3)
  + [2.4 开发语言](http://www.cnblogs.com/Wddpct/p/5694596.html#2.4)
* [3. Get Started](http://www.cnblogs.com/Wddpct/p/5694596.html#3.)
  + [3.1 Win 10](http://www.cnblogs.com/Wddpct/p/5694596.html#3.1)
    - [3.1.1 安装.NET Core SDK和.NET Core Runtime](http://www.cnblogs.com/Wddpct/p/5694596.html#3.1.1)
    - [3.1.2 简单的运行结果](http://www.cnblogs.com/Wddpct/p/5694596.html#3.1.2)
  + [3.2 CentOS 7.2（本地Hyper-V）](http://www.cnblogs.com/Wddpct/p/5694596.html#3.2)
    - [3.2.1 安装和运行](http://www.cnblogs.com/Wddpct/p/5694596.html#3.2.1)
    - [3.2.2 Self-contained applications 发布](http://www.cnblogs.com/Wddpct/p/5694596.html#3.2.2)
      * [（1） 修改project.json文件](http://www.cnblogs.com/Wddpct/p/5694596.html#3.2.2.1)
      * [（2） 执行restore和publish操作](http://www.cnblogs.com/Wddpct/p/5694596.html#3.2.2.2)
      * [（3） 在Linux平台上运行](http://www.cnblogs.com/Wddpct/p/5694596.html#3.2.2.3)
* [4. 总结](http://www.cnblogs.com/Wddpct/p/5694596.html#4.)
  + [4.1 尾声](http://www.cnblogs.com/Wddpct/p/5694596.html#4.1)
  + [4.2 相关学习资料](http://www.cnblogs.com/Wddpct/p/5694596.html#4.2-1)
* [5. 参考资料](http://www.cnblogs.com/Wddpct/p/5694596.html#5.)

**1. 前言**

.NET发行至今已经过了十四个年头。随着版本的不断迭代更新，.NET在Windows平台上的表现也是越来越好，可以说Windows平台上所有的应用类型.NET几乎都能完成。

只是成也Windows，败也Windows，这十四年来，除了部分“民间”版本，.NET一直没能在官方支持下摆脱Windows平台的局限，“开源”和“跨平台”这两个词语也是所有.NET开发者心中的痛楚。最终，.NET Core出现了，它让开发者们在官方和社区的支持走出了Windows，可以在macOS，Linux主流distributions上编写调试并部署.NET程序。

**2. .NET Core 简介**

**2.1 .NET Core是什么**

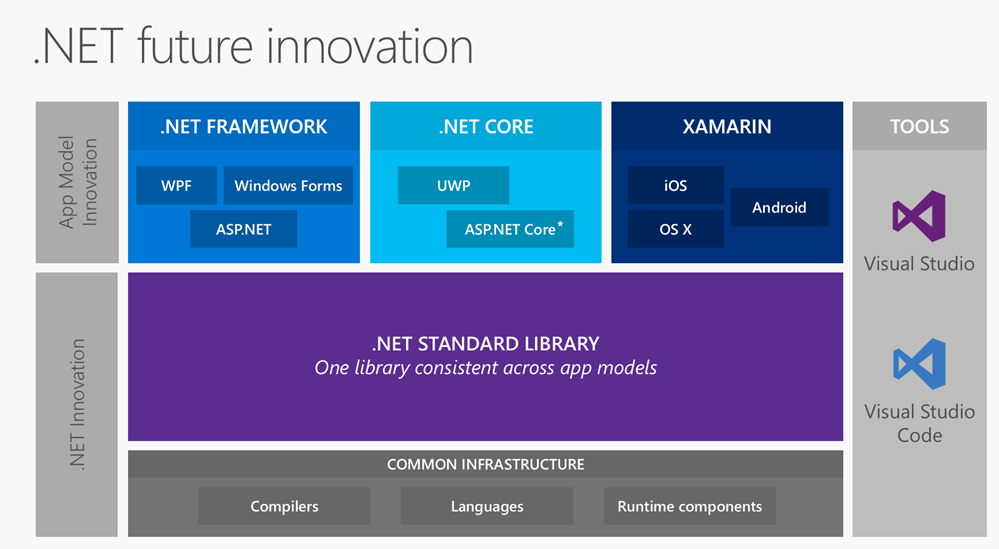
.NET Core是一个开源通用的开发框架，支持跨平台，即支持在Window，macOS，Linux等系统上的开发和部署，并且可以在硬件设备，云服务，和嵌入式/物联网方案中进行使用。.NET Core的源码放在[GitHub](https://github.com/dotnet/core)上，由微软官方和社区共同支持。

它和传统的.NET Framework，属于“子集—超集”的关系，或者你也可以简单地认为它就是.NET Framework的跨平台版本（基于[BCL](https://zh.wikipedia.org/wiki/%E5%9F%BA%E7%A1%80%E7%B1%BB%E5%BA%93)的层面上看）。这是因为在当前版本中（1.0），.NET Core中的大部分核心代码都是从.NET Framework中继承重写的，包括Runtime和Libraries（如GC, JIT, 部分类型）。

吐槽：只能感谢微软《CLR via C#》不用白看呀，我之前看了得有小半年才看完

现在的.NET Core 1.0版本是一个很小的核心，APIs和工具也并不完整，但是随着.Net Core的不断完善，补充的Apis和创新也会一起整合到.NET Framework中。也就是说，.NET Core微软会同时更新.NET Framework和.NET Core，他们就像俩兄弟，共同努力致富（致谁的富？当然是.NET开发者们），实现所谓的天下大同，也就是[.NET 标准2.0](http://www.cnblogs.com/Wddpct/p/5685341.html)

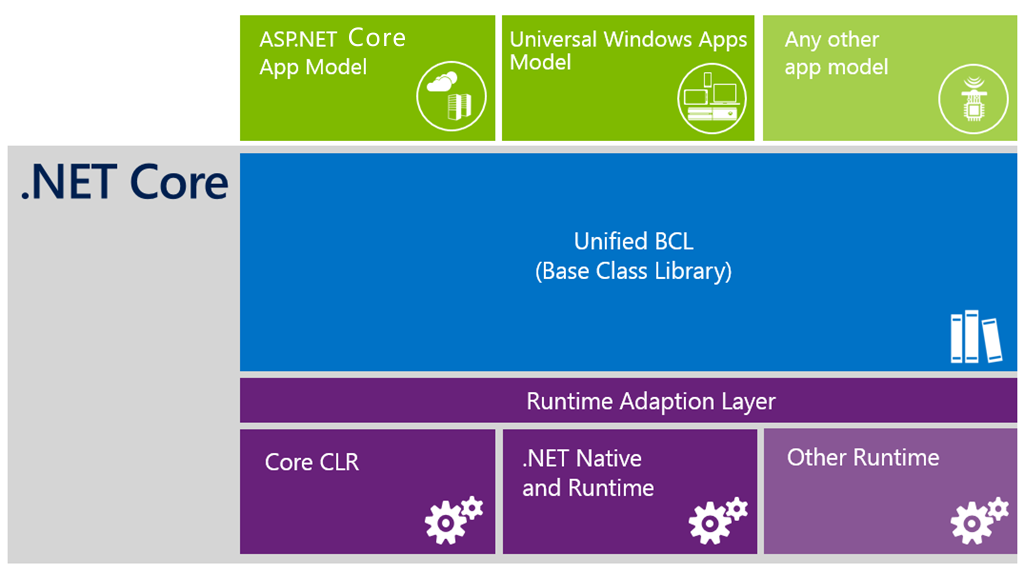
这里不得不提到一个叫做[.NET Standard Library](https://docs.microsoft.com/zh-cn/dotnet/articles/standard/library" \l "targeting-net-standard-library)概念。作为.NET平台APIs开发官方支持标准，它要求所有的.NET框架APIs都遵循向下兼容。比如说.NET Framwork 4.6支持.NET Standard Library 1.3，.NET Framwork 4.6.2框架支持.NET Standard Library 1.5，而.NET Core 1.0框架支持1.6标准。

最终展望如下：  


**2.2 .NET Core的组成**

* .NET Runtime  
  即[CoreCLR](https://github.com/dotnet/coreclr)。如之前所说，CoreCLR与.NET Framework的CLR并没有什么区别，进程管理，GC，JIT（RyuJIT 编译器）这些部分也都是一样的，只是针对服务器系统做了相应优化。现在CLR和CoreCLR也在进行同步更新，只是可以肯定的是，CoreCLR才是.NET的未来，CLR将作为兼容手段而存在
* Framework Libraries,  
  即[CoreFX](https://github.com/dotnet/corefx)。包括集合类，文件系统处理类，XML处理类，异步Task类等
* SDK Tools 和 Language Compilers （SDK工具和编译器）  
  即[CLI](https://github.com/dotnet/cli)工具和[Roslyn编译器](https://github.com/dotnet/roslyn)。可以通过[.NET Core SDK](https://docs.microsoft.com/dotnet/articles/core/sdk)(.NET Core开发工具包)获取。
* dotnet’app host  
  用来选取并执行对应运行时，提供组件载入原则和启动.NET Core应用程序。SDK也是通过相同程序启动。

Tips:是不是想起了MSCorEE.dll这个垫片，它同样承载着Windows平台上为.NET应用程序选择CLR版本的工作



**2.3 .NET Core的特性**

* 跨平台  
  可以在Windows，macOS，Linux上运行
* 灵活的部署机制

1.Portable applications(便携式应用)  
这种部署机制和传统的.NET Framework相似，只要目标平台上存在.NET Core Runtime即可。

2.Self-contained application（自宿主应用）  
顾名思义，这种部署机制将应用和运行时共同打包，即便目标平台上没有安装.NET Core Runtime也能正常使用

第二种方式和.NET Native也是不一样的，仍然使用CoreCLR，而.NET Native使用CoreRT作为运行时，详细信息请见[dotnet/corert](https://github.com/dotnet/corert/blob/master/Documentation/intro-to-corert.md)

* 命令行工具  
  .NET程序所有的运行脚本都可以用命令行工具执行(cmd，bash)这里有几个常见的donnet命令

| **指令** | **帮助** |
| --- | --- |
| dotnet new | 产生新的基本 .NET 项目内容 (包含 project.json、Program.cs 以及 NuGet.config |
| dotnet restore | 还原所参考的 NuGet 包 |
| dotnet build | 建造 .NET 项目 |
| dotnet publish | 产生可发行的 .NET 项目 (包含所属的 Runtime) |
| dotnet run | 编译与立即运行 .NET 项目 (较适用于 exe 型项目) |
| dotnet repl | 引导交互式的对话 |
| dotnet pack | 将项目的产出封装成 NuGet 包 |

* 兼容性  
  通过[.NET Standard Library](https://docs.microsoft.com/dotnet/articles/standard/library)与.NET Framework，Xamarin，Mono兼容
* 开源  
  .NET Core从属于[.NET基金会](http://www.dotnetfoundation.org/projects)，由微软进行官方支持。使用MIT和Apache 2开源协议，文档协议遵循[CC-BY](https://zh.wikipedia.org/wiki/%E5%88%9B%E4%BD%9C%E5%85%B1%E7%94%A8#.E5.85.AD.E7.A8.AE.E6.A0.B8.E5.BF.83.E6.8E.88.E6.AC.8A.E6.A2.9D.E6.AC.BE)

**2.4 开发语言**

.NET Core 1.0版本中支持的变成语言仅有C#（F#和VB暂未实现），这里还要提到一个开源的语言编译器Roslyn，它负责将代码编译成我们熟悉的IL语言，然后再借由AOT或JIT编译器编译成机器熟悉的机器语言。

**3. Get Started**

以下内容演示在Windows10和CentOS 7.2下的命令行生成和发布demo

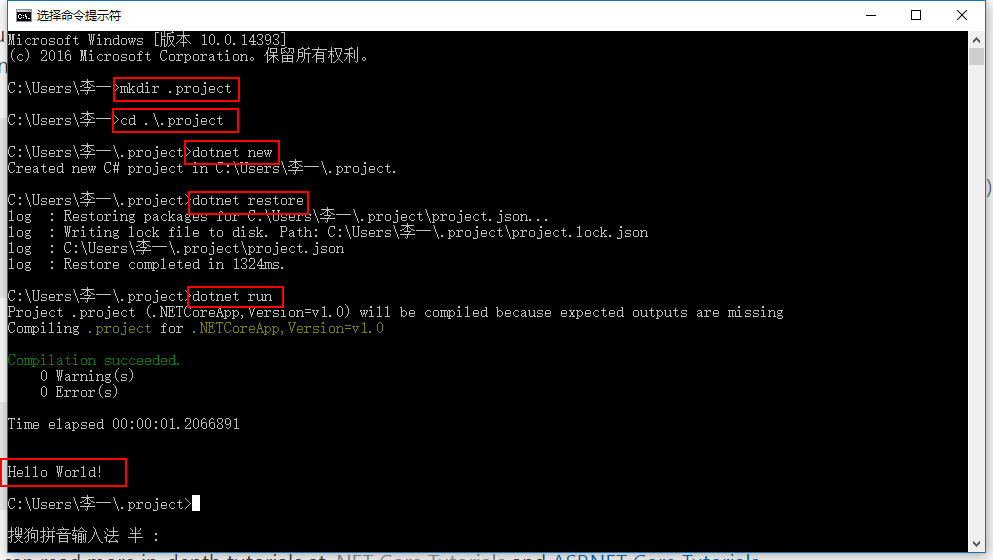
**3.1 Win 10**

**3.1.1 安装**[**.NET Core SDK**](https://www.microsoft.com/net/download#core)**和**[**.NET Core Runtime**](https://www.microsoft.com/net/download#core)

.NET Core SDK = 使用.NET Core开发应用.NET Core Runtime 和 SDK+CLI 工具

**3.1.2 简单的运行结果**

打开cmd，依次输入mkdir .project（创建目录），cd .\.project（进入目录），dotnet new（新建初始项目），dotnet restore（还原依赖），dotnet run（运行）即可运行第一个Hello World程序



**3.2 CentOS 7.2（本地Hyper-V）**

**3.2.1 安装和运行**

详情请见：[.NET Core in CentOS](https://www.microsoft.com/net/core#centos)，大致命令如下

sudo yum install libunwind libicu #安装libunwind，libicu包

curl -sSL -o dotnet.tar.gz https://go.microsoft.com/fwlink/?LinkID=809131 #下载dotnet-dev-centos-x64.1.0.0-preview2-003121.tar文件，有时会因为网络问题下载较慢，耐心等待即可，当然也可以手动下载后放到目录下。

sudo mkdir -p /opt/dotnet && sudo tar zxf dotnet.tar.gz -C /opt/dotnet #创建目录并解压已下载文件

sudo ln -s /opt/dotnet/dotnet /usr/local/bin #将目录链接到$PATH下，否则dotnet命令无法识别

mkdir hwapp

cd hwapp

dotnet new #创建默认.NET Core应用

dotnet restore #还原依赖包

dotnet run #运行，结果将显示Hello World!

第六行命令后可使用dotnet --info查看是否链接成功，显示如下

.NET Command Line Tools (1.0.0-preview2-003121)

Product Information:

Version: 1.0.0-preview2-003121

Commit SHA-1 hash: 1e9d529bc5

Runtime Environment:

OS Name: centos

OS Version: 7

OS Platform: Linux

RID: centos.7-x64

以上步骤在[.NET Core](https://www.microsoft.com/net/core#windows)官方网站都可以找到，可以看到应用在经过简单的dotnet new, dotnet restore, dotnet run命令后就跑起来了，但是这其实是类似于开发环境中的调试运行，而且win上new的应用此时也不能直接跨平台到Linux下运行，所以我们又得提到dotnet publish命令了

**3.2.2 Self-contained applications 发布**

**（1） 修改project.json文件**

我们现在win10下按照步骤new一个新的HW控制台应用self，按照[官方文档](https://docs.microsoft.com/zh-cn/dotnet/articles/core/tutorials/using-with-xplat-cli)要求，我们需要用以下内容替换原来project.json文件（删除"type": "platform",并增加runtimes节点）

{

"version": "1.0.0-\*",

"buildOptions": {

"debugType": "portable",

"emitEntryPoint": true

},

"dependencies": {},

"frameworks": {

"netcoreapp1.0": {

"dependencies": {

"Microsoft.NETCore.App": {

"version": "1.0.0"

}

},

"imports": "dnxcore50"

}

},

"runtimes":{

"win10-x64":{}, #win10平台

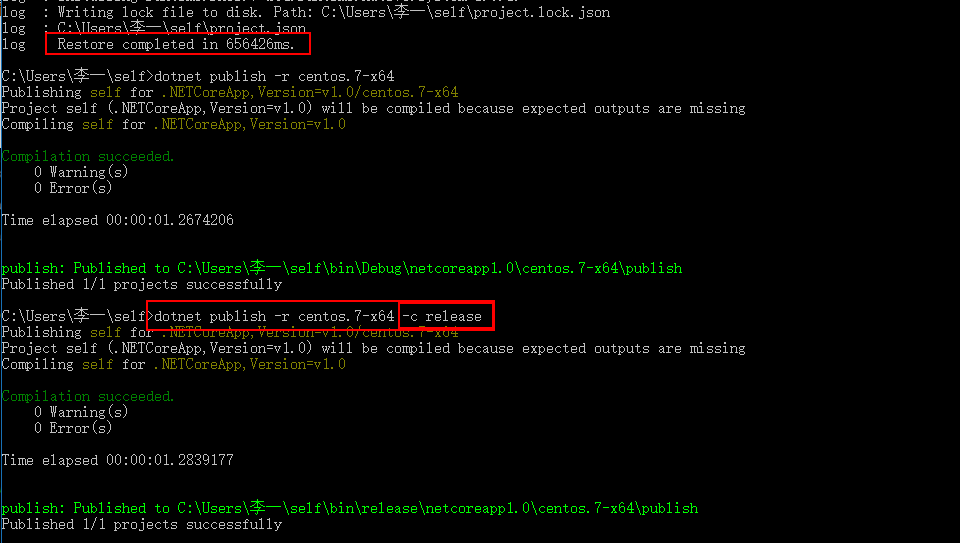
"centos.7-x64":{} #centos7.2平台

}

}

**（2） 执行restore和publish操作**

之后执行dotnet restore指令，针对平台进行还原操作。该步操作耗时较久，虽然只是两个平台，第一次也花了较长时间。紧接着进行进行dotnet publish -r centos.7-x64 -c release。



dotnet publish指令详见[dotnet-publish - Packs the application and all of its dependencies into a folder getting it ready for publishing](https://docs.microsoft.com/zh-cn/dotnet/articles/core/tools/dotnet-publish)

**（3） 在Linux平台上运行**

上述操作后，我们只需要将发布后的文件夹（bin/release/netcoreapp1.0/centos7-x64/publish，包含self.exe）上传到Linux root目录下project（新建）文件夹中， 在Shell中输入指令



修改可执行权限后，即可成功运行“Hello World”，这就是我们“Self-contained applications”方式的部署方法

**4. 总结**

**4.1 尾声**

至此，.NET Core的学习便告一段落了，以上内容简单介绍了.NET Core的组成和特性，同时也在Windows和Linux系统上通过两种不同的部署方式成功运行了示例。相比较之前的.NET Framework傻瓜式部署，.NET Core的新奇真是让我眼前一亮。接下来，我也会将.NET Core和ASP.NET Core的学习记录下来，欢迎指正。

**4.2 相关学习资料**

* [张善友的博客](http://www.cnblogs.com/shanyou/)
* [.NET Core Documentation](https://docs.microsoft.com/zh-cn/dotnet/)
* [ASP.NET Core Documentation](https://docs.asp.net/en/latest/) #园子里已经有部分中文版
* [汤姆大叔的博客：解读ASP.NET 5 & MVC6系列](http://www.cnblogs.com/TomXu/p/4496545.html)
* [.NET Blog](https://blogs.msdn.microsoft.com/dotnet/) #非常有用的.NET官方资讯
* [未完待续](http://www.cnblogs.com/Wddpct/p/5694596.html)

**5. 参考资料**

* [简析.NET Core 以及与 .NET Framework的关系](http://www.cnblogs.com/vipyoumay/p/5603928.html)
* [.NET Core SDK Overview](https://docs.microsoft.com/zh-cn/dotnet/articles/core/sdk)
* [.NET Core 1.0 is now released!](http://www.hanselman.com/blog/NETCore10IsNowReleased.aspx)
* [.NET Core](https://www.microsoft.com/net/core#windows)
* [拥抱.NET Core](http://www.cnblogs.com/ants/p/5659123.html)
* [.NET Core 维基百科](https://zh.wikipedia.org/wiki/.NET_Core)
* [让我们Core在一起：ASP.NET Core & .NET Core](http://news.cnblogs.com/n/544410/)