

```
public class HelloWorld {
    #include<iostream>
    using std::cout;
    using std::endl;
    public static void main(String args[]) {
        System.out.println("Hello World!");
    }
}
DSEG SEGMENT
STR DB "Hello World!"
DSEG ENDS
CSEG SEGMENT
ASSUME DS:DSEG,CSEG
START: MOV AX,DSEG
MOV DS,AX
MOV DX,OFFSET STR
MOV AH,09H
INT 21H
MOV AH,4CH
INT 21H
CSEG ENDS
END START
```

Hello World!

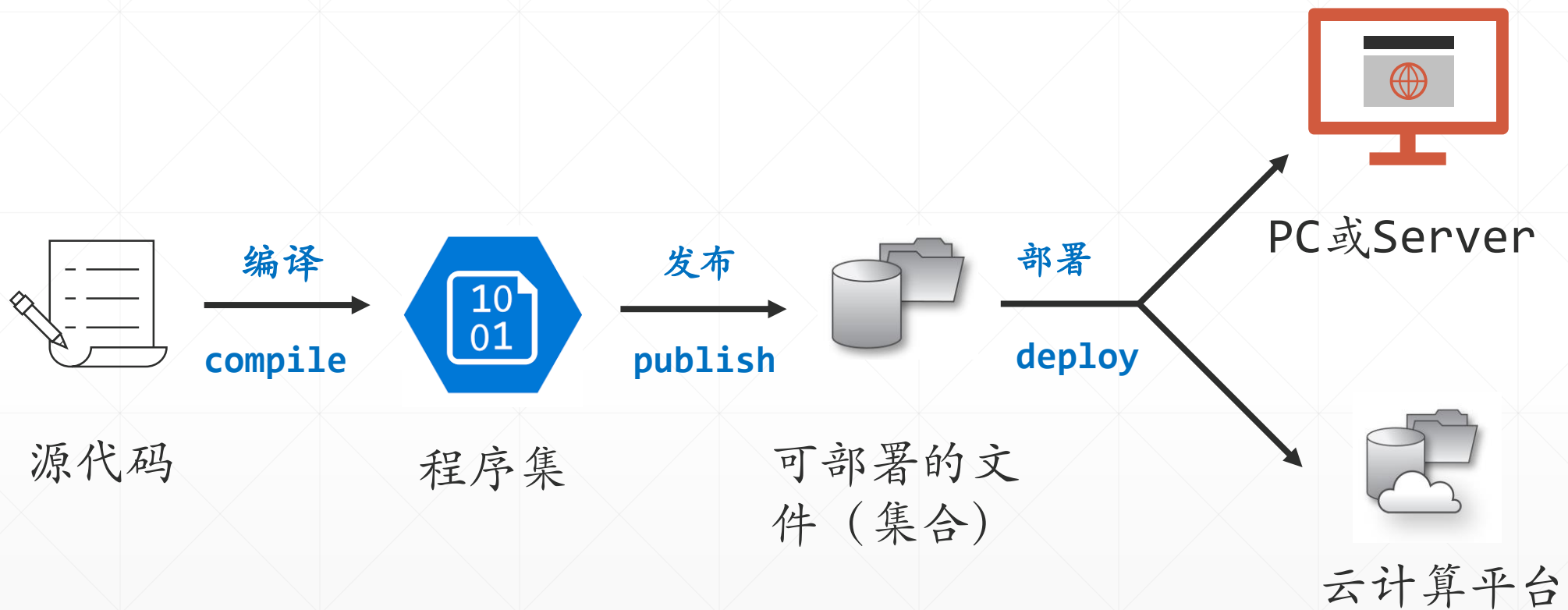
```
#include<iostream>
using std::cout;
using std::endl;
void main() {
    cout<<"Hello World!"<<endl;
    return 0;
}
printf("Hello World!");
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace hello
{
    class Program
    {
        static void Main(string[] arg
        {
            Console.WriteLine("Hello
        }
    }
}
```

.NET Core 的 Hello World 之旅

2 .NET core项目的发布和部署

北京理工大学计算机学院
金旭亮

.NET Core应用程序的“发布”与“部署”



.NET core程序的发布 (publish) 方式

1

Framework Dependent
(框架依赖的发布模式)

只部署本App相关的程序集及相关资源，要求目标机器上已经安装有兼容的.NET core运行时。

2

Self-contained
(自包容的发布模式)

除了部署本App相关的程序集及相关资源之外，还把.NET core运行时环境一并打包，以便在目标机器上无需安装.NET core而直接运行。

以“框架依赖”方式发布



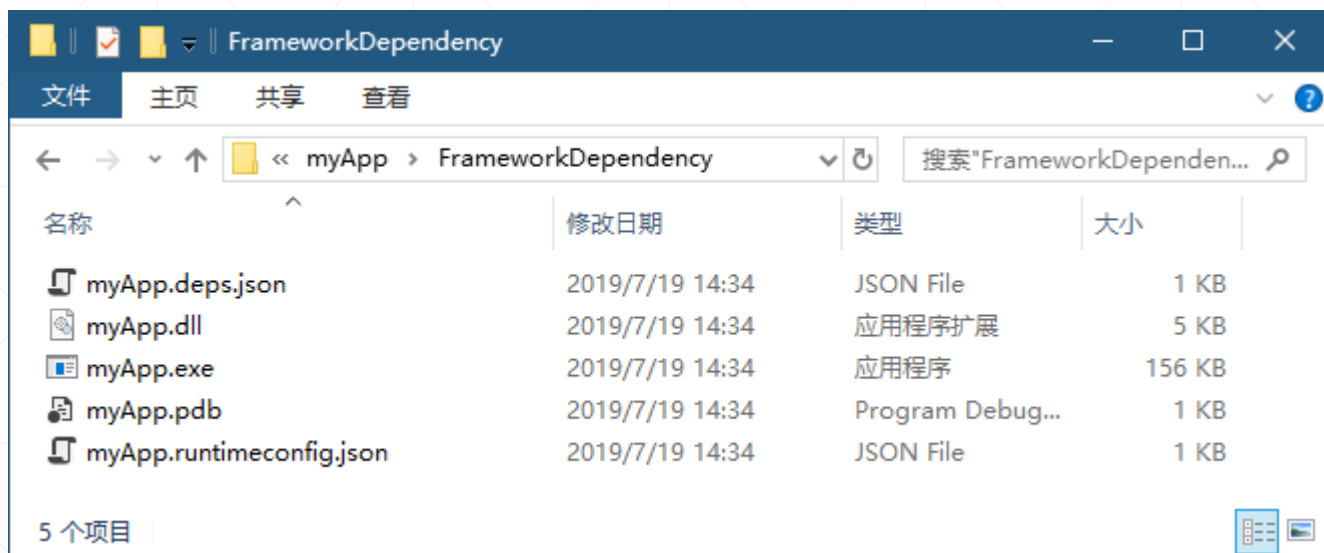
使用“**dotnet publish**”命令发布 .NET Core 应用：

```
C:\> 命令提示符
D:\myApp>dotnet publish -c Release -o FrameworkDependency
用于 .NET Core 的 Microsoft (R) 生成引擎版本 16.2.0-preview-19278-01+d635043bd
版权所有(C) Microsoft Corporation。保留所有权利。

D:\myApp\myApp.csproj 的还原在 68.25 ms 内完成。
C:\Program Files\dotnet\sdk\3.0.100-preview6-012264\Sdks\Microsoft.NET.Sdk\targets\Microsoft.NET.RuntimeIdentifierInference.targets(158,5): message NETSDK1057:
你正在使用 .NET Core 的预览版。请查看 https://aka.ms/dotnet-core-preview [D:\myApp\myApp.csproj]
    myApp -> D:\myApp\bin\Release\netcoreapp3.0\myApp.dll
    myApp -> D:\myApp\FrameworkDependency\

D:\myApp>
```

.NET core 应用程序的发布结果



在发布文件夹中，除了可执行程序文件，还有两个json文件，其中包容了一些配置信息。

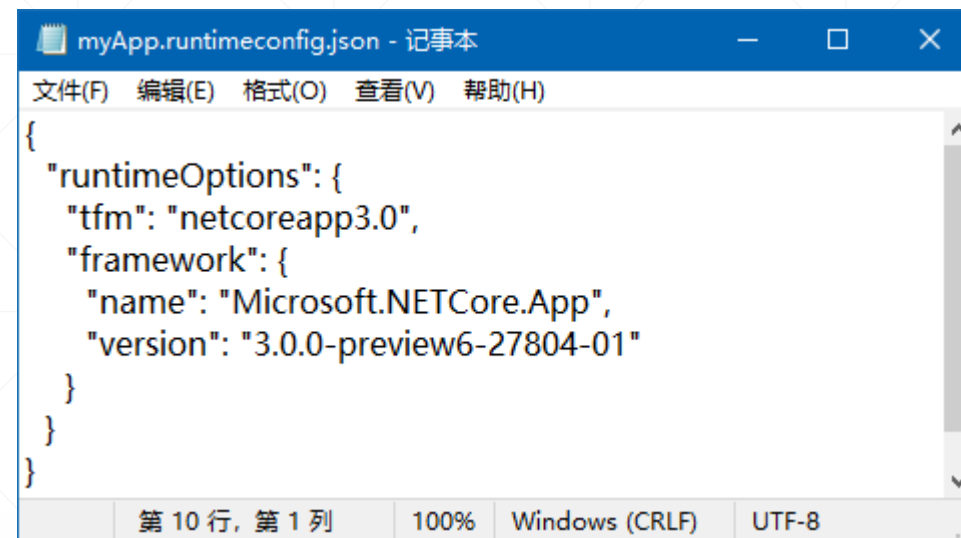
.NET core程序运行时所需的配置参数



```
myApp.deps.json - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
{
  "runtimeTarget": {
    "name": ".NETCoreApp,Version=v3.0",
    "signature": ""
  },
  "compilationOptions": {},
  "targets": {
    ".NETCoreApp,Version=v3.0": {
      "myApp/1.0.0": {
        "runtime": {
          "myApp.dll": {}
        }
      }
    }
  },
  "libraries": {
    "myApp/1.0.0": {
      "type": "project",
      "serviceable": false,
      "sha512": ""
    }
  }
}
```

第 24 行, 第 1 列 100% Windows (CRLF) UTF-8

指明所依赖
的组件信息



```
myApp.runtimeconfig.json - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
{
  "runtimeOptions": {
    "tfm": "netcoreapp3.0",
    "framework": {
      "name": "Microsoft.NETCore.App",
      "version": "3.0.0-preview6-27804-01"
    }
  }
}
```

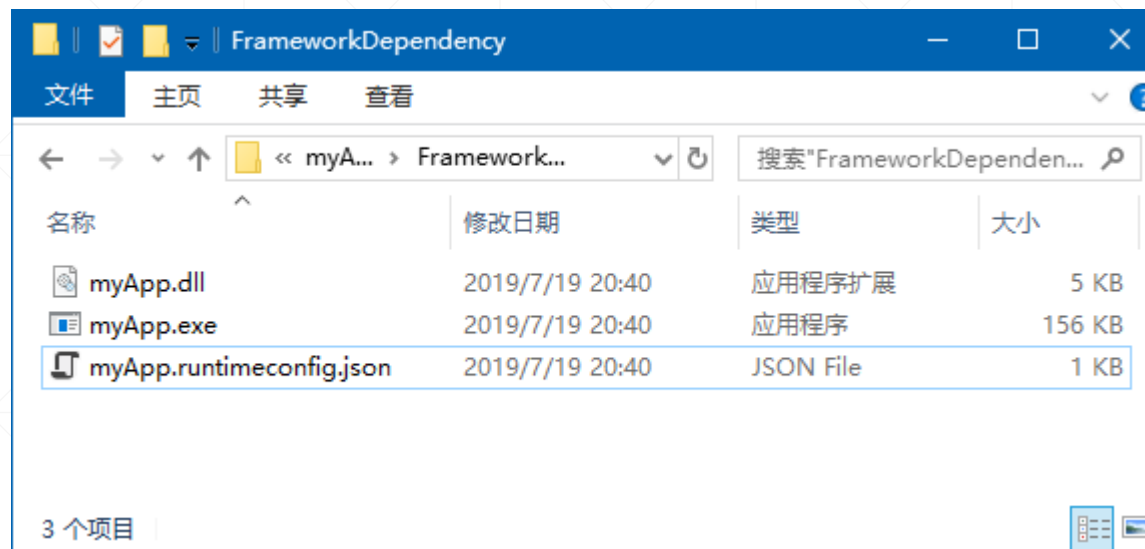
第 10 行, 第 1 列 100% Windows (CRLF) UTF-8

指明运行环境要求

当使用dotnet命令运行程序时，它需要从runtimeconfig.json文件中读取相应的配置参数以启动相应的运行时，因此，这个文件是不能缺失的。

进一步“精简”.....

五个文件中的.pdb和deps.json文件可以被删除，不影响myApp程序的运行。



```
命令提示符

D:\myApp\FrameworkDependency>dotnet myApp.dll
A fatal error was encountered. The library 'hostpolicy.dll' required to ex
ecute the application was not found in 'D:\myApp\FrameworkDependency\'.
Failed to run as a self-contained app. If this should be a framework-depen
dent app, add the D:\myApp\FrameworkDependency\myApp.runtimeconfig.json fi
le specifying the appropriate framework.

D:\myApp\FrameworkDependency>_
```

runtimeconfig.json不能删除，如
果删除，程序运行时引发致命错误。





以“自包容”方式发布

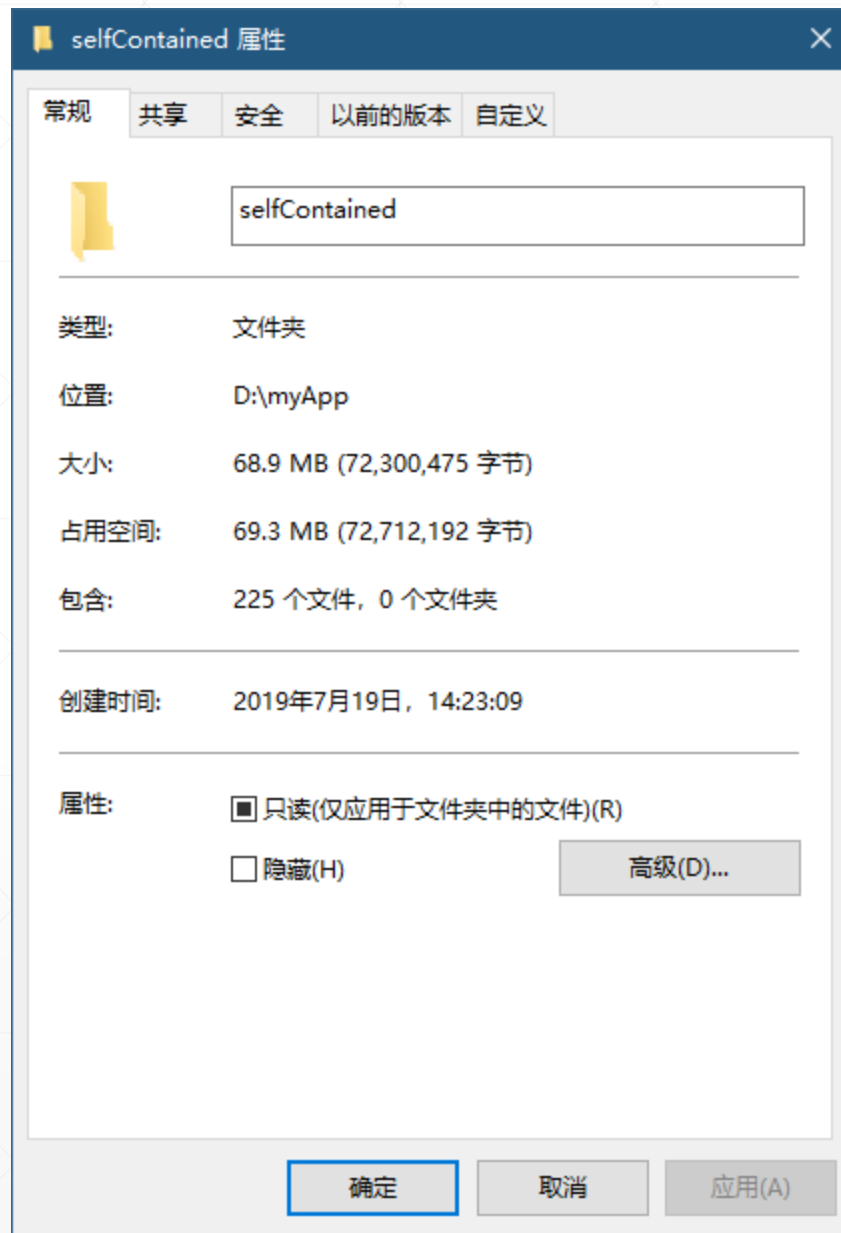
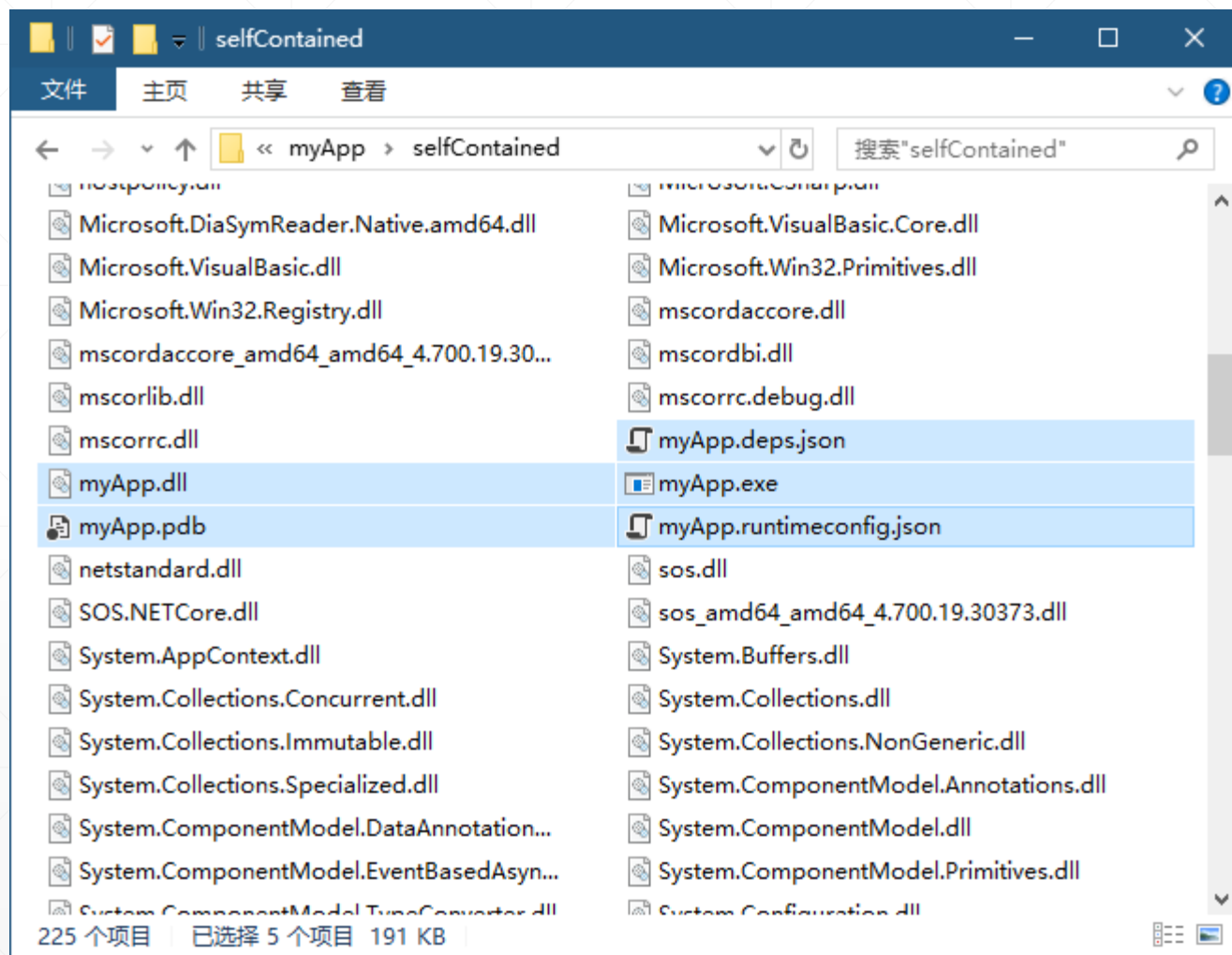
```
C:\> 命令提示符

D:\myApp>dotnet publish -c Release -o selfContained -r win-x64 --self-contained
true
用于 .NET Core 的 Microsoft (R) 生成引擎版本 16.2.0-preview-19278-01+d635043bd
版权所有(C) Microsoft Corporation。保留所有权利。

D:\myApp\myApp.csproj 的还原在 26.05 ms 内完成。
C:\Program Files\dotnet\sdk\3.0.100-preview6-012264\Sdks\Microsoft.NET.Sdk\targe
ts\Microsoft.NET.RuntimeIdentifierInference.targets(158,5): message NETSDK1057:
你正在使用 .NET Core 的预览版。请查看 https://aka.ms/dotnet-core-preview [D:\myA
pp\myApp.csproj]
myApp -> D:\myApp\bin\Release\netcoreapp3.0\win-x64\myApp.dll
myApp -> D:\myApp\selfContained\

D:\myApp>_
```


“自包容”方式发布的文件



第三种方式—单文件部署

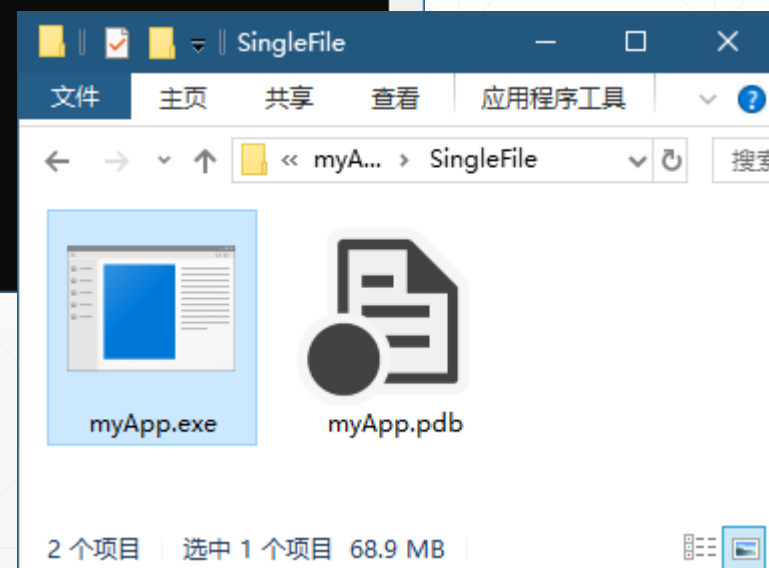


```
命令提示符

D:\myApp>dotnet publish -r win-x64 -o SingleFile /p:PublishSingleFile=true
用于 .NET Core 的 Microsoft (R) 生成引擎版本 16.2.0-preview-19278-01+d635043bd
版权所有(C) Microsoft Corporation。保留所有权利。

D:\myApp\myApp.csproj 的还原在 21.44 ms 内完成。
C:\Program Files\dotnet\sdk\3.0.100-preview6-012264\Sdks\Microsoft.NET.Sdk\targe
ts\Microsoft.NET.RuntimeIdentifierInference.targets(158,5): message NETSDK1057:
你正在使用 .NET Core 的预览版。请查看 https://aka.ms/dotnet-core-preview [D:\myA
pp\myApp.csproj]
myApp -> D:\myApp\bin\Debug\netcoreapp3.0\win-x64\myApp.dll
myApp -> D:\myApp\SingleFile\

D:\myApp>
```



myApp三种发布方式的对比



	框架依赖	自包容	单文件
发布之后文件数目	5	255	2
发布文件的总容量	164KB	约70M	约70M
目标计算机上必须安装.NET core?	Yes	No	No