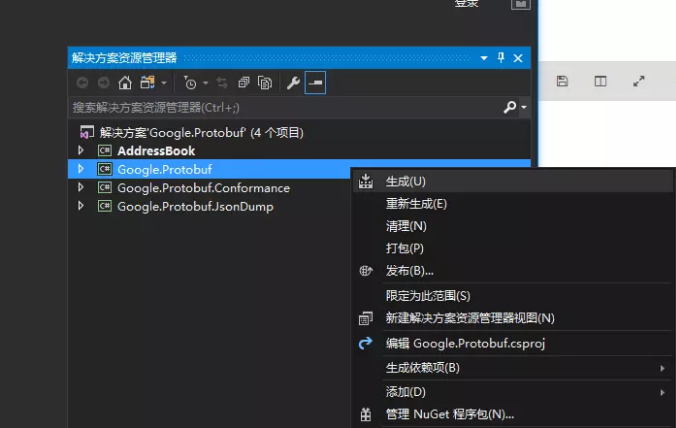
#### **使用Protobuf的流程：**

将数据对象定义在.proto文件中，用protoc.exe编译器把.proto文件编译成CS脚本，编译过程可以把.proto里定义的数据对象转换成数据类，并实现部分基本方法。实体类有了，就可以创建类对象，并给参数赋值。要实现实体类对象及数据转换就需要用到Protobuf的DLL文件里面的方法了。

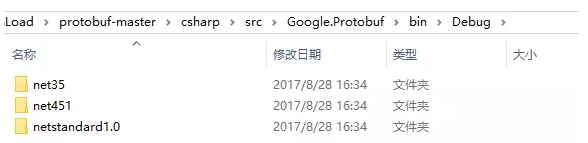
在Unity中使用Protobuf需要：C#用的protobuf的DLL文件，proto编译器（protoc.exe）。

#### **准备DLL文件：**

1. 从GitHub上下载protobuf源码 （源码链接：[https://github.com/google/protobuf](https://link.jianshu.com/?t=https://github.com/google/protobuf" \t "https://www.jianshu.com/p/_blank)）  
   2，打开工程目录下 csharp/src/Google.Protobuf.sln 文件。 PS：在下用的VS2017打开的  
   3，生成DLL文件 如下：

 Paste\_Image.png

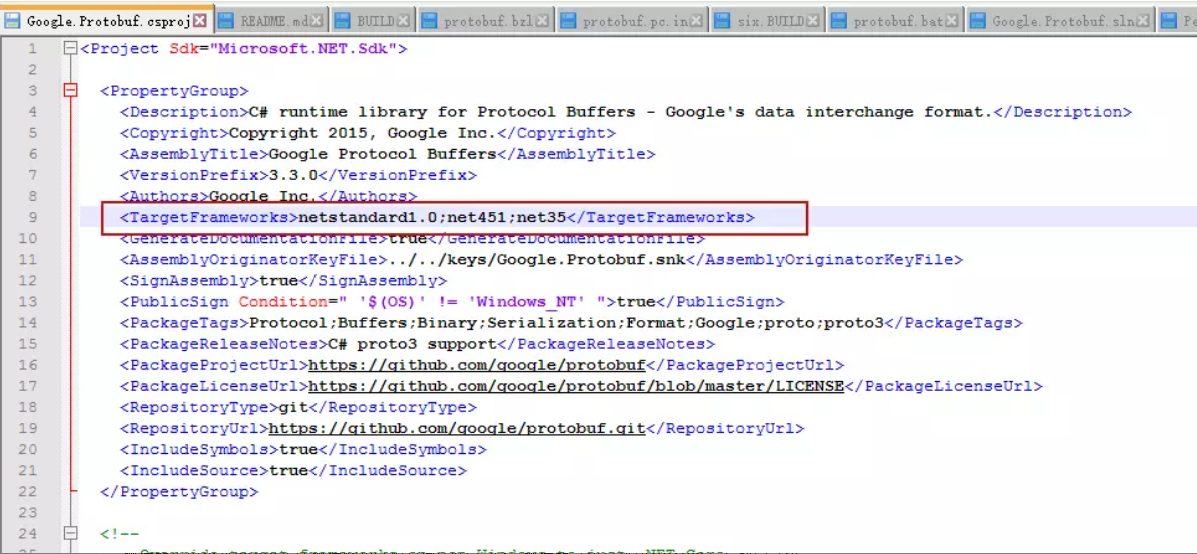
输出文件如下：

 Paste\_Image.png

4，在每个net版本文件夹下面都有Google.Protobuf.dll文件 （需要将对应得net版本DLL导入到Unity中）

PS：如何设置DLL导出版本

打开csharp/src/Google.Protobuf/Google.Protobuf.csproj 配置文件 （该文件中主要保存工程的配置信息），修改TargetFrameworks标签的参数

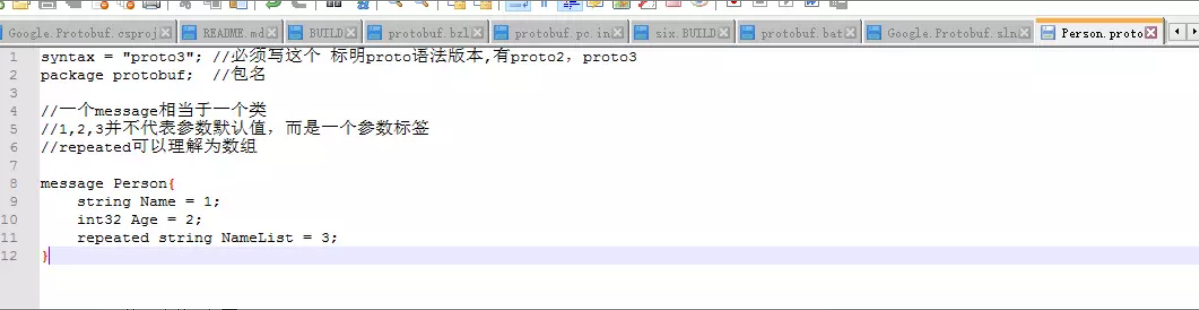
 Paste\_Image.png

#### **准备编译器：**

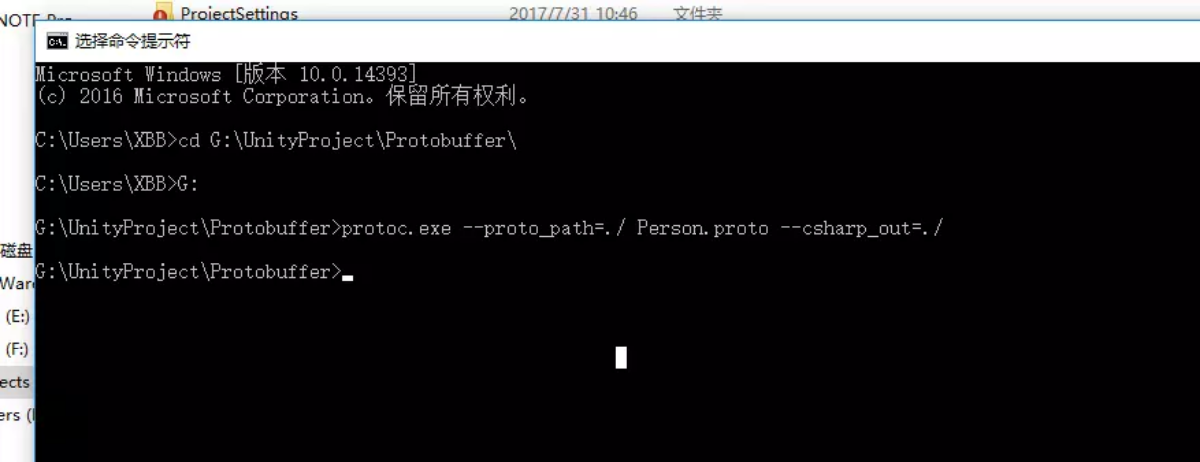
编译器是用来将.proto文件编译成相应语音脚本的工具， 编译器可以直接从GitHub上下载也可以选择自己使用工具生成。  
1，GitHub下载 （Git地址 [https://github.com/google/protobuf/releases](https://link.jianshu.com/?t=https://github.com/google/protobuf/releases" \t "https://www.jianshu.com/p/_blank)）， 下载 对应的protobuf包 （如 protoc-3.4.0-win32.zip）， 在bin文件夹下有对应得 protoc.exe 编译器  
2，自己使用工具生成protoc.exe （有兴趣可以看下在下这篇文章 [http://www.jianshu.com/p/f806faef245f](https://www.jianshu.com/p/f806faef245f" \t "https://www.jianshu.com/p/_blank)）

#### **实现protobuf数据转换**

1. 书写一个.proto文件 如下（记事本即可，将扩展名改成.proto）

 Paste\_Image.png

1. 用编译器将.proto文件编译称CS文件  
   在cmd窗口中打开protoc.exe所在文件夹，执行编译命令，成功编译的话会在该文件生成相应的.cs文件。如下：（cd到该文件夹，然后执行这行编译命令，编译命令即：protoc.exe --proto\_path=./ Wode.proto --csharp\_out=./）

 Paste\_Image.png



Paste\_Image.png

--proto\_path 指定要编译的.proto文件路径 （相对路径）  
--csharp\_out 输出cs文件路径（相对路径）

更多proto语法，命令： [https://developers.google.com/protocol-buffers/docs/proto3](https://link.jianshu.com/?t=https://developers.google.com/protocol-buffers/docs/proto3" \t "https://www.jianshu.com/p/_blank)

3，将上面生成的Google.Protobuf.dll 和 Person.cs文件导入到Unity中，  
测试代码如下：

using UnityEngine;using Protobuf; //应用CS文件的命名空间 （.proto文件中的 package 值）using Google.Protobuf; //引用DLL

public class Test : MonoBehaviour {

void Start()

{

//新建一个Person对象，并赋值

Person p = new Person();

p.Name = "IongX";

p.Age = 22;

p.NameList.Add("熊");

p.NameList.Add("棒");

p.NameList.Add("棒");

//将对象转换成字节数组

byte[] databytes = p.ToByteArray();

//将字节数据的数据还原到对象中

IMessage IMperson = new Person();

Person p1 = new Person();

p1 = (Person)IMperson.Descriptor.Parser.ParseFrom(databytes);

//输出测试

Debug.Log(p1.Name);

Debug.Log(p1.Age);

for (int i = 0; i < p1.NameList.Count; i++)

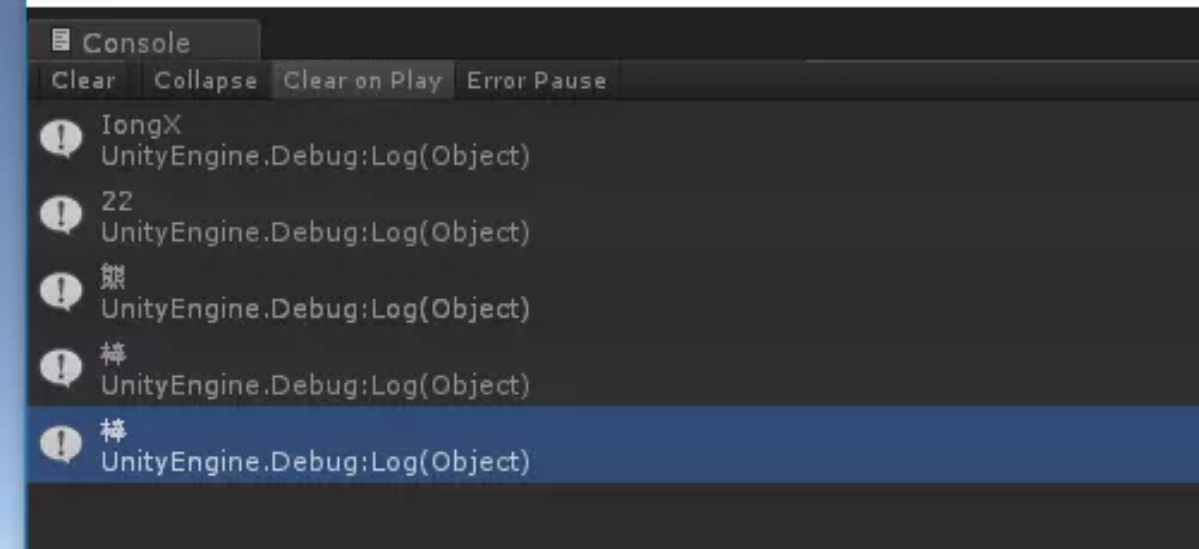
{

Debug.Log(p1.NameList[i]);

}

}}

输出结果如下：



Paste\_Image.png

#### **最后：**

以上纯属个人总结，如有不对或者更好的方法，欢迎指正，交流。  
工程文件链接 ： [https://github.com/IongX/Unity\_ProtobufTest](https://link.jianshu.com/?t=https://github.com/IongX/Unity_ProtobufTest" \t "https://www.jianshu.com/p/_blank)

原文地址：在Unity中使用Protobuf实现数据转换 - 简书

<https://www.jianshu.com/p/b135676dbe8d>