使用前请仔细阅读该文档

1. 目录结构

```
proto
|- pb 编译后的文件(客户端用)
|- proto proto源文件
|- tools 编译用工具
|- make.bat 批处理工具
|- ...
```

2. 协议命名规范

```
// 通用协议头部
syntax = "proto2";
option optimize_for = LITE_RUNTIME;

// -- 此处填写模块名称
// -- 命名规范: P + 模块名称
package PBag;

---

//-- 此处描述了客户端与服务器通信的指令id
//-- 以'c'开头的为客户端发往服务器协议
//-- 以's'开头的协议为服务器发往客户端的协议
//-- 'c'与's'开头的协议均从1开始递增
// CmdId描述(统一服务器&客户端)
// {
//     cBagUseItem = 1,//使用道具
//
//     sBagInit = 1,//初始化
//     sBagGainItem = 2,//获取道具
// };

////////////////////////////////////////////////////////////////////////
// Common message
// -- 此处放置一些通用的对象，用于在其他协议中复用，例如背包中的PBagItem就是一个通用对象
// -- 命名规范: P + 对象名称
message PBagItem
{
    required int32 itemid = 1; //道具ID
}

////////////////////////////////////////////////////////////////////////
// Client -> Server
// -- 此处放置客户端发往服务器的协议
// -- 命名规范: C + 协议用途描述
```

```
message CUseItem
{
    required int32 itemid = 1;//道具id
}

//////////////////////////////////////////////////////////////////////////
// Server -> Client
// -- 此处放置服务器发往客户端的协议
// -- 命名规范: S + 协议用途描述
message SInit
{
    repeated PBagItem itemlist = 1;//道具列表
}
```

3. 使用方法

> 编辑协议内容，执行`make.bat`并提交SVN

4. 注意事项及F&Q

> - 通用的协议放在: `PCommon` 中 使用 `import "PCommon.proto";` 进行引用
> - 模块相关的协议必须设置包名,包命与文件名相同。例 : `package PLogin;`
> - 所有协议文件首行都必须申明使用的proto版本信息 : `syntax = "proto2";
> - proto2中需要明文申明该值必传 `require` 关键词
> - 为了优化服务器对协议解析的效率及资源占用，必须在所有文件中包含如下定义: `option optimize_for = LITE_RUNTIME;`
> - 好的设计可以让开发效率大幅提高，所以一定要复用`message`!!! 复用`message`!!! 复用`message`!!! (重要事情说三遍)

5. vscode编辑

> - 基于当前文档目录打开vscode
> - 在插件中搜索 `vscode-proto3` 并安装
> - 在工程编辑中输入 `ctrl + shift + b`可以自动运行 make.bat文件实现编译，方便快捷操作