ProtoBuffer介绍

特点：

1、平台无关、语言无关。

2、二进制、数据自描述。

3、提供了完整详细的操作API。

4、高性能 比xml要快20-100倍

5、尺寸小 比xml要小3-10倍 高可扩展性

6、数据自描述、前后兼容

一、下载protobuf（客户端引擎中使用的是2.x版本）：

官网：https://developers.google.com/protocol-buffers/

二、安装：

unzip protobuf-2.x.x.zip

cd protobuf-2.x.x

./configure

make

make check

sudo make install

sudo make ldconfig

三、使用：

1. 定义模型文件（.proto）

**import** "myproject/CommonMessages.proto"

enum UserStatus

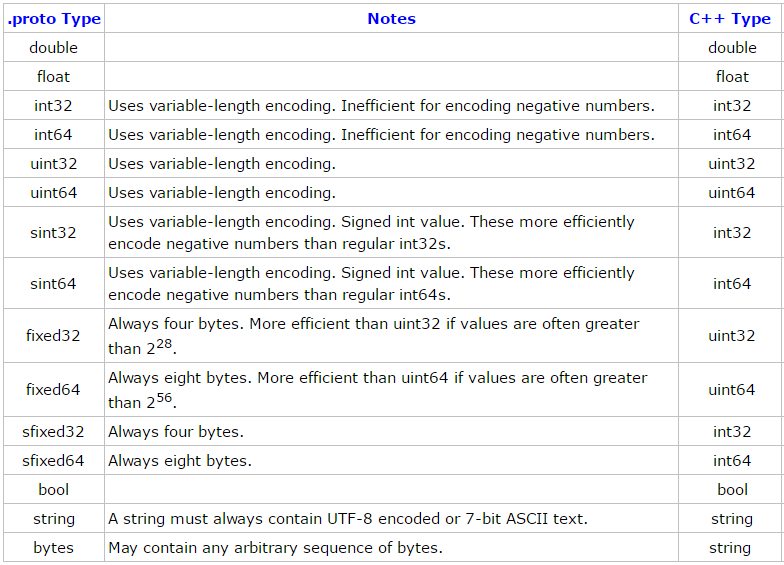
{  
  OFFLINE = 0;  
  ONLINE = 1;  
}  
message UserInfo

{  
 **required** int64 acctID = 1;  
 **optional** string name = 2;  
 **required** UserStatus status = 3;  
}  
message LogonRespMessage

{  
 **required** LoginResult logonResult = 1;  
 **repeated** UserInfo userInfo = 2;  
}

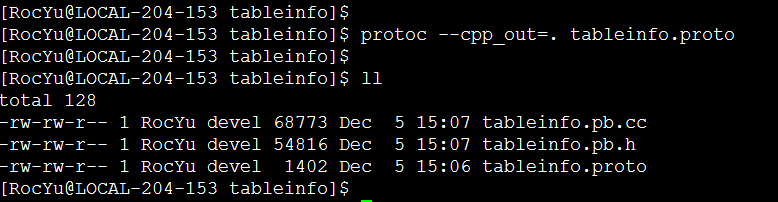
说明：

1. message是消息定义的关键字，等同于C++中的struct/class，enum与C++中等同。
2. LogonReqMessage为消息的名字，等同于结构体名或类名。
3. acctID和name分别表示消息字段名，等同于C++中的成员变量名。
4. 标签数字**1**和**2**则表示不同的字段在序列化后的二进制数据中的布局位置，在同一message中不能重复。标签值为1到15的字段在编码时可以得到优化，既标签值和类型信息仅占有一个byte，标签范围是16到2047的将占有两个bytes，总共支持字段数为2的29次方减一。
5. 每个消息中必须至少留有一个required类型的字段。
6. 可以使用import引入其他文件中定义的消息。
7. 可以在proto文件中定义包名，转换成C++时替换为命名空间。
8. 类型对照表



1. 编译生成C++文件

**protoc** --proto\_path=IMPORT\_PATH --cpp\_out=DST\_DIR path/to/file.proto



1. 应用到项目中
2. 将生成的tableinfo.pb.cc和tableinfo.pb.h拷贝到项目源码目录。
3. 修改makefile，引用.cc文件，并在编译选项中加入-lprotobuf