# protobuf的使用说明

1. 下载与安装

1.1 下载protobuf-2.5.0.tar.gz

1.2 安装

使用tar -zxf protobuf-2.5.0.tar.gz命令解压后得到是 protobuf-2.5.0的源码，

cd protobuf-2.5.0 进入目录

假如 你希望编译成功后输出的目录 为 /home/work /protobuf/ 则输入如下两条命令：

./configure --prefix=/home/work /protobuf/

make && make install

编译成功后将export PATH= /home/work /protobuf/bin:$PATH加入到环境变量中

最后输入 protoc --version命令，如显示libprotoc 2.5.0则安装成功

1. 使用

2.1 在.proto文件中定义消息格式

|  |
| --- |
| syntax = "proto2";  package tutorial;  message Person {  required string name = 1;  required int32 id = 2;  optional string email = 3;  enum PhoneType {  MOBILE = 0;  HOME = 1;  WORK = 2;  }  message PhoneNumber {  required string number = 1;  optional PhoneType type = 2 [default = HOME];  }  repeated PhoneNumber phones = 4;  }  message AddressBook {  repeated Person people = 1;  } |

**.proto文件以包声明开始，防止名字冲突。**

**简单类型：bool, int32, float, double, string.**

**其它类型：如上述的Person, PhoneNumber**

**类型可以嵌套。**

**“=1”， “=2”标识唯一“tag”.tag数1-15需要至少一个字节。**

**required: 必须设置它的值**

**optional: 可以设置，也可以不设置它的值**

**repeated: 可以认为是动态分配的数组**

2.2 生成代码

执行命令：

protoc -I=$SRC\_DIR --cpp\_out=$DST\_DIR $SRC\_DIR/addressbook.proto

--cpp\_out 表示生成 c++代码

SRC\_DIR 源文件目录，可以不指定

DST\_DIR 目标文件目录，生成的代码会放在这个指定目录

生成后的文件为：

addressbook.pb.h,

addressbook.pb.cc

为方便编译，可以将.cc 改为.cpp

2.3 接口使用

生成的代码有以下接口：

|  |
| --- |
| inline bool has\_name() const;  inline void clear\_name();  inline const ::std::string& name() const;  inline void set\_name(const ::std::string& value);  inline void set\_name(const char\* value);  inline ::std::string\* mutable\_name();  // id  inline bool has\_id() const;  inline void clear\_id();  inline int32\_t id() const;  inline void set\_id(int32\_t value);  // email  inline bool has\_email() const;  inline void clear\_email();  inline const ::std::string& email() const;  inline void set\_email(const ::std::string& value);  inline void set\_email(const char\* value);  inline ::std::string\* mutable\_email();  // phone  inline int phone\_size() const;  inline void clear\_phone();  inline const ::google::protobuf::RepeatedPtrField< ::tutorial::Person\_PhoneNumber >& phone() const;  inline ::google::protobuf::RepeatedPtrField< ::tutorial::Person\_PhoneNumber >\* mutable\_phone();  inline const ::tutorial::Person\_PhoneNumber& phone(int index) const;  inline ::tutorial::Person\_PhoneNumber\* mutable\_phone(int index);  inline ::tutorial::Person\_PhoneNumber\* add\_phone(); |

2.4 解析与序列化

bool SerializeToString(string\* output) const: 序列化消息，将存储字节的以string方式输出。注意字节是二进制，而非文本；

bool ParseFromString(const string& data): 解析给定的string

bool SerializeToOstream(ostream\* output) const: 写消息给定的c++ ostream中

bool ParseFromIstream(istream\* input): 从给定的c++ istream中解析出消息