# 客户端Excel导出lua说明

## Excel格式

即支持【xlsx】也支持【xls】，但是为了和导出C#配置兼容，所以Excel格式依然使用【xls】

## 配置结构说明

### 配置文件目录



### 创建配置文件

简单粗暴的方法，直接复制一个已经存在的配置文件

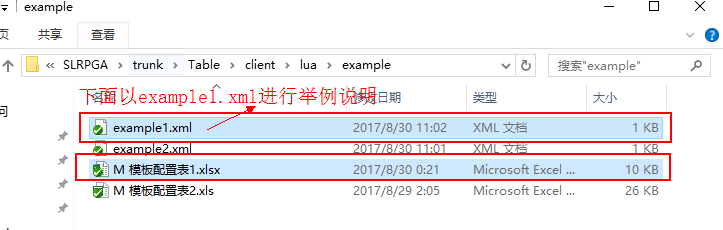
### 编辑配置文件

配置文件使用【xml】格式

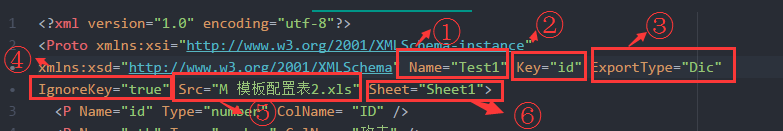
推荐使用【Atom】【Subline】【Notepad++】，甚至是【文本编辑器】进行编辑

### 配置文件属性

以模板结构说明，模板路径如下截图：



#### 1），结构属性



##### ，Name属性

A，必填；

B，表示通过这个结构生成的lua文件的名称

##### ，Key属性

A，必填；

B，用以区分不同数据条目；

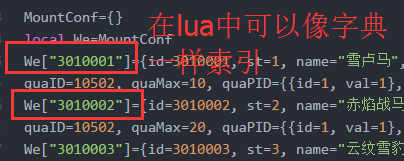
C，其数据类型可以是Number(数字) 或者 字符串；

D，值应该是子元素中某个的Name属性的值

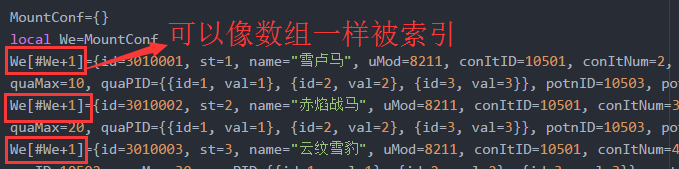
##### ，ExportType属性

A，选填；值有Dic(字典)和Arr(数组)；默认值为Arr

B，Dic时，导出的配置如下：



C，Arr，导出的配置如下



使用Arr属性时，在从Excel导出时，根据ID进行了升序排序；

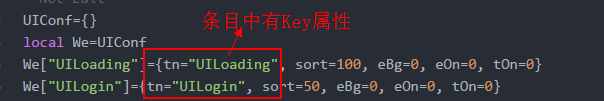
在Lua中可以使用BinaryTool（二分搜索工具）快速搜索

使用Arr时，键值类型只能是Number(数字类型)

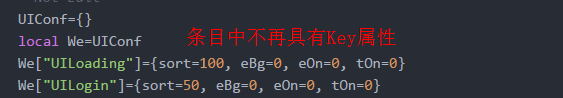
##### ， IgnoreKey属性

A，可选的；

B，默认值为false时；



值为true时：



##### ，Src属性

A，必填；

B，Excel文件路径，

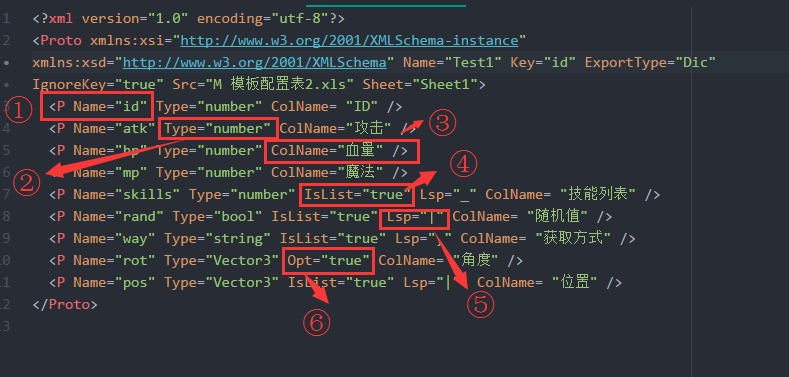
C，直接填写文件名称属性即可；

##### ，Sheet属性

A，必填；

B，Excel文件中表单名称；

#### 2），元素属性



##### ①， Name属性

A，必填；

B，导出lua文件中的数据条目属性名称

C，尽量使用缩写，以减少字符占用

##### ②， Type属性

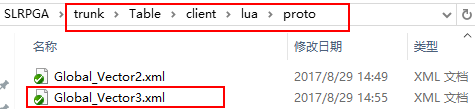
A，必填；

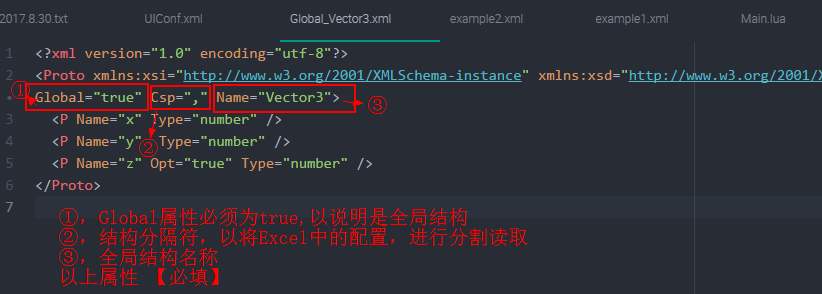
B，基础数据类型有三种

1. Number ，数字类型，只要是数字填这个就行，而不用理会数字是多少位
2. Bool，布尔类型，建议使用Number代替，在Excel配置表中使用0代替false，使用1代替true
3. String，字符类型
4. 类型名称不区分大小写

C，全局类型

1. 如果类型的值，不是以上三种，则会把此类型当作自定义类型，在当前配置目录中搜索否有全局类型，
2. 不允许在配置文件中定义其它类型，
3. 全局类型只能在配置文件中单独定义，以如下文件进行说明：





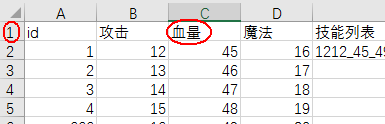
1. 全局类型的文件名称加上Global\_前缀以示说明

##### ③， ColName属性

* + 1. 必填；
    2. 对应Excel中的第一行的列名，如：



对应：



##### ④， IsList属性

1. 可选的；

B，如果设置为true，则代表这个属性是一个数组/列表，并且 【Lsp】属性必须填写

##### ⑤， 结构元素 Lsp属性

A，数组/列表分隔符

B，如果【IsList】设置为true，则这个属性必须填写，

其值只能是如下字符的其中一个：

{ , 逗号} { | 逗号} { ; 分号} { : 冒号} { \_ 下划线}

C，同样的【全局类型】中的分隔符也只能是其中一个

D，如果发现分割符数量不够时，可以先检查是否合理的设计配置

如果确实数量不够，在扩展

##### ⑥， Opt属性

A，可选的；

B，如果值为true，则在导出配置时，没有配置就不导出，这里程序员需要注意，在脚本中这个对应属性就是nil；

C，目的是减少lua配置文件占用磁盘空间的大小

##### ，Default属性

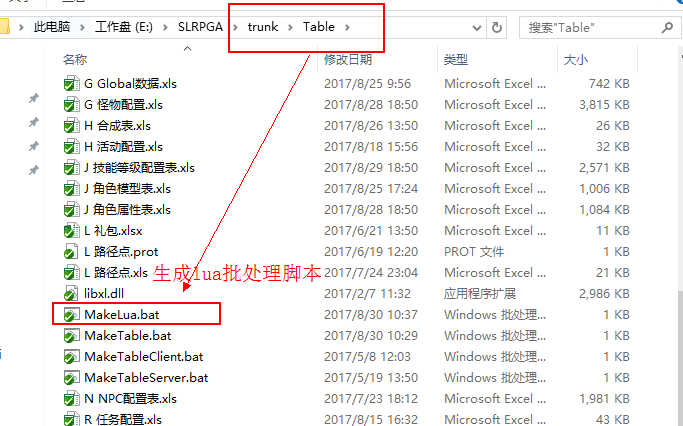
* + 1. 可选的
    2. 其值导出时未填写的默认值

## 三，配置生成

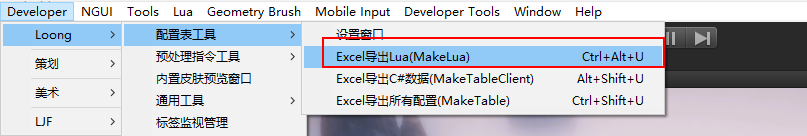
确保配置结构无误；确保Excel已经配置；在这样的前提下，可以使用如下方式生成lua配置：

### 直接双击批处理文件

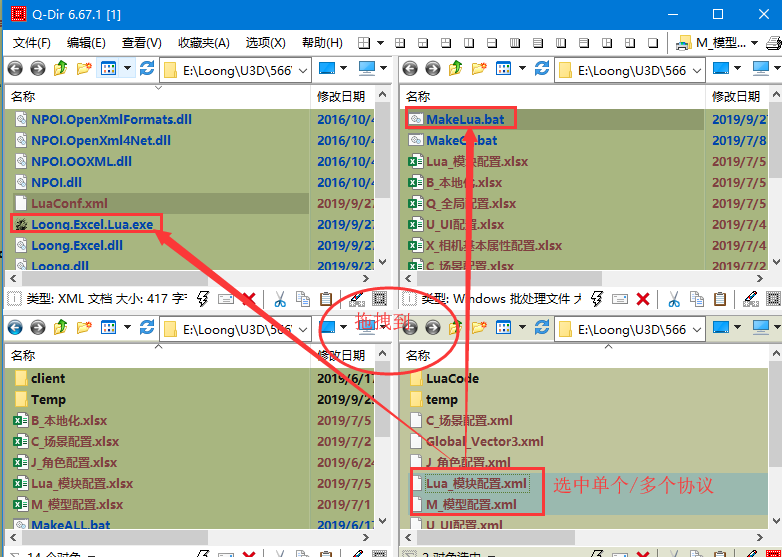
此时会生成所有配置



### 在工程内使用菜单或者快捷键



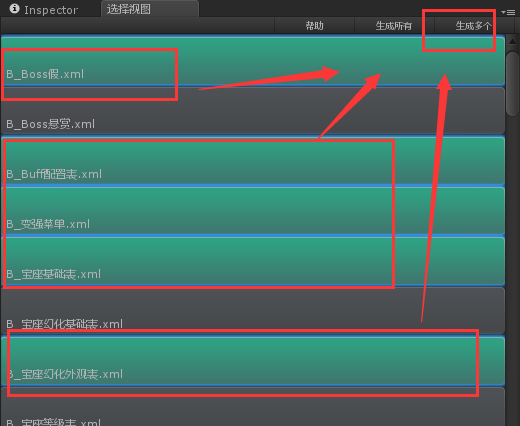
### 选中单个/多个协议文件拖拽到BAT或EXE文件批量生成



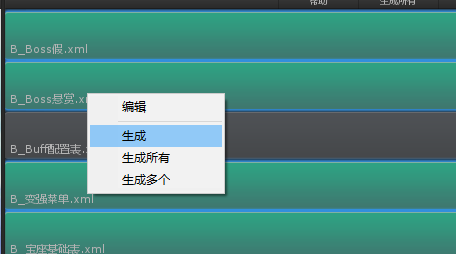
### 在工程内使用可视化工具生成

菜单【Developer/Loong/配置表工具/LUA】

* 1. ，多选生成



* 1. ，单独生成



### 也可以使用全部导出

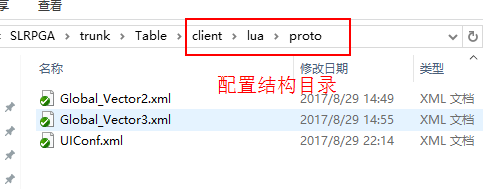
即在2图的MakeTable，除了导出lua配置，还会导出C#的二进制.tbl文件

## 四，配置提交到SVN

生成配置后，要在工程内跑下，看下程序运行状态下，所需求的配置是否更改并且无误；

确保无误后：

### 提交配置结构



### 提交lua生产文件

