**版本修订表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 状态 | 日期 | 修改人 | 说明 |
| V1.0 | 新建 | 2016.10.31 | 易嘉祯 | 编写一期设计方案 |

## 命名规范

除文档中提到的类名以外类名都以LQ开头(类名和函数名都是最好一看就明白具体作用):

例如： class LQCommonTools // 即CommonTools

{

….

}

类中的成员：

指针类型： SocketID\* m\_pSocketID;

Bool\* m\_pbStopThread;

Bool类型： Bool m\_bStopThread;

Typedef (void\*tfFunctionName)(char\* inContent, char\* outContent);

(Void\* fFunctionName)(char\* inContent, char\* outContent);

函数指针前加f(即function)，函数指针类型前加tf(即typedef function);

静态类型：static int sNum; static int\* spNum;

定义新的数据类型，结构体，枚举，常量和宏时都以LQ\_开头：

例如：

1.定义新数据类型

typedef string LQ\_IP; // IP

typedef unsigned int LQ\_Port; // Port

2.结构体

typedef struct

{

LQ\_IP ip;

LQ\_Port port;

}LQ\_AddressBase;

2.宏定义和常量定义

#define LQ\_MAX\_IP\_LENGTH 32(等价于const Uint32 LQ\_MAX\_IP\_LENGTH = 32)

3.枚举(内部需用lq开头)

(1). 已命名的枚举

/\* LQ\_Status用于表示状态类型 \*/

enum LQ\_Status

{

lqStatOK = 0,

lqStatInteralError = 1,

lqStatError = 2,

lqStatNumParams = 3 // 用于表示总共枚举种类个数（若第一个枚举变量值为0，且依次递增，则把该枚举变量加上，否则可以不写该枚举），命名规则：枚举类型名(lqStat)+NumParams

};

(2). 未命名的枚举

enum

{

lqStatOK = 0,

lqStatInteralError = 1,

lqStatError = 2,

lqStatNumParams = 3

};

全局变量前一律加g\_(最好不要用全局变量):

int g\_threadCount = 0; // 全局变量存储线程个数

int\* g\_pThreadCount = NULL;

所有公共用的结构体，枚举，常量，宏，自定义数据类型，都放在consdef.h文件中(所有.h文件都将包含该文件)

## 注释规范

## 编码规范

## 版本控制

1. 版本命名规范

完全的版本号定义，分四项：：<主版本号>.<子版本号>.<阶段版本号>.<日期版本号>，如 1.1.1.051021。

2. 版本号修改规则

\* 主版本号(1)：当功能模块有较大的变动，比如增加多个模块或者整体架构发生变化。此版本号由项目决定是否修改。

\* 子版本号(1)：当功能有一定的增加或变化，比如增加了对权限控制、增加自定义视图等功能。此版本号由项目决定是否修改。

\* 阶段版本号(1)：一般是 Bug 修复或是一些小的变动，要经常发布修订版，时间间隔不限，修复一个严重的bug即可发布一个修订版。此版本号由项目经理决定是否修改。

\* 日期版本号(051021):用于记录修改项目的当前日期，每天对项目的修改都需要更改日期版本号。此版本号由开发人员决定是否修改。