

# 代码注释规范

---

NSLAB, 2011/09/01

## 说明

本规范符合 Doxygen 一般要求, 仅供 NSLAB 组内成员参考使用。部分信息来源网络, 不对内容、版权以及引起的任何问题负任何责任。内容主要包括文件头部的注释要求、函数的注释要求、类型注释以及代码中的注释。其他更复杂情况建议大家参考 Doxygen 的规范进行。所举的例子为了方便叙述, 中英混杂, 但从代码的移植性上来说**建议大家均采用英文注释和说明**。

## 文件注释

### 头文件

头文件头部注释规范, 注释必须列出版权说明、版本号、生成日期、作者、内容、功能、与其它文件的关系、修改日志等, 复杂的头文件的注释中还应包含函数功能简要说明。一般格式为

@关键字 具体内容。

例如

```
/**
Copyright (C), 2010-2011.
@file header.h
@brief This file is a Try, it has something powerful.
@author Venus Team
@email venus@nslab.org
@version 1.0
@date 2011-09-01
@update 2011-09-01
@warning This is only a test.
@todo Something in plan.
@history: revision history...
*/
```

### 源文件

源文件头部注释规范, 列出: 版权说明、版本号、生成日期、作者、模块目的/功能、主要函数及其功能、修改日志等。示例: 下面这段源文件的头注释比较标准, 当然, 并不局限于此格式, 但上述信息建议要包含在内。

```
/**
Copyright (C), 2010-2011.
@file body.c
```

```
@brief This file is a Try, it has something powerful.
@author Venus Team
@email venus@nslab.org
@version 1.0
@date 2011-09-01
@update 2011-09-01
@todo Something in plan.
@warning This is only a test.
@list: func1, func2, func3...
@history: revision history...
*/
```

## 函数注释

函数头部注释放在函数代码开始之前，主要列出：函数的目的/功能、输入参数、输出参数、返回值、调用关系（函数、表）等。示例：下面这段函数的注释比较标准，当然，并不局限于此格式，但上述信息建议要包含在内：

```
/**
 * @fn func_name
 * @brief describe what does the function do.
 * @param param1 This is a param.
 * @param param2 This is another param.
 * @return return_value
 */
```

## 类型注释

### 类

```
/**
 * @class class_name
 * @brief briefly introduce the class.
 * Here detailly describe the class.
 */
```

### 结构体

```
/**
 * @struct struct name
 * @brief briefly introduce the struct.
 * Here detailly describe the struct.
 */
```

## 内部变量

所有的内部变量注释规范为

///< 注释，例如

```
///< a public variable. Details.
///< another public variable. Details.
```

## 语句注释

语句的注释，主要包括段落注释和临时注释两种，段落注释建议为

```
/* This block mainly does sth. */  
Code bla bla..
```

临时注释跟在代码行的后面，可以为

```
Code bla bla. /// why should I do this?
```