

My title

Bo Ma

2019 年 10 月 21 日

1 一般步骤

1. 生成配置文件 `Doxyfile: doxygen (-s) -g`
2. 建立目录结构, 比如`Doxyfile`文件 `\doc`文件夹(输出)\`src`文件夹(放代码) ->三者在同级
3. 根据情况设置配置文件中的参数 (关键)
4. 运行 `doxygen Doxyfile` 生成大概结构
5. 针对每个代码文件写 `doxygen` 注释
6. 继续运行 `doxygen Doxyfile`

2 配置文件 Doxyfile 设置

```
DOXYFILE_ENCODING      = UTF-8
PROJECT_NAME            = "Project Name"
PROJECT_NUMBER          = 1.0
PROJECT_BRIEF           = "This is a brief descript."
EXTRACT_ALL             = YES
OUTPUT_DIRECTORY        = "./doc"
OUTPUT_LANGUAGE          = Chinese
FULL_PATH_NAMES         = NO
WARN_LOGFILE             = "./doc/build.log"
INPUT                   = "./src"
FILE_PATTERNS           =
SOURCE_BROWSER           = NO
```

```

GENERATE_LATEX      = NO
CALL_GRAPH          = YES
CALLER_GRAPH        = YES
UML_LOOK            = YES
RECURSIVE           = YES

```

2.0.1 文件注释

```

/**
 * Copyright (c) 2017, All rights reserved.
 *
 * @file $file$
 * @brief $brief$
 * Details.
 *
 * @author $author$, $email$
 * @date $yy$-$mm$-$dd$
 * @version $maj$.$min$
 *
 *****/

```

```

\subsubsection{命名空间}%
\label{ssub:ming_ming_kong_jian_}

```

```

\begin{verbatim}
/**
 * @brief $brief$
 * Details.
 **/

```

2.0.2 结构体/枚举

```

/**
 * @brief $brief$
 * Details.

```

```
**/
```

2.0.3 类注释

```
/**  
 * @brief $brief$  
 * Details.  
 **/
```

2.0.4 函数注释

```
/**  
 * @brief $brief$-$test$  
 * Details.  
 * @param $param1$ : $param1_detail$  
 * @param $param2$ : $param2_detail$  
 * @return $return_detail$  
 *      -<em>false</em> fail  
 *      -<em>true</em>  succeed  
 * @retval $return_note$  
 * @deprecated $deprecated$  
 * @see $see$  
 * @pre $pre$  
 **/
```

3 关键字收集

- @author 作者
- @brief 摘要
- @version 版本号
- @date 日期

- `@file` 文件名，可以默认为空，DoxyGen 会自己加
- `@class` 类名
- `@param` 函数参数
- `@return` 函数返回值描述
- `@exception` 函数抛异常描述
- `@warning` 函数使用中需要注意的地方
- `@remarks` 备注
- `@see` see also 字段
- `@note` brief 下空一行后的内容表示详细描述，但也可以不空行用 `note` 表示
- `@par` 开始一个段落，段落名称描述由你自己指定，比如可以写一段示例代码
- `@code` 引用代码段
- `@endcode` 引用代码段结束
- `@pre` 函数前置条件，比如对输入参数的要求
- `@post` 函数后置条件，比如对系统状态的影响或返回参数的结果预期
-
- `@am[in|out]` 参数名及其解释
- `@exception` 用来说明异常类及抛出条件
- `@return` 对函数返回值做解释
- `@note` 表示注解，暴露给源码阅读者的文档
- `@remark` 表示评论，暴露给客户程序员的文档
- `@since` 表示从那个版本起开始有了这个函数
- `@deprecated` 引起不推荐使用的警告
- `@see` 表示交叉参考

`@author` 作者