

Doxygen記述ルール

C/C++言語のソースを記述例について説明。

Doxygen文法マニュアル

「[Doxygen マニュアル](#)」の特に下記ページを参照。

- [特殊コマンド](#)
- [コードのドキュメント付け](#)
- [Markdownサポート](#)

Doxygen設定項目

設定ファイル「Doxyfile」は、doxygen -g コマンド実行でデフォルト値ファイルが生成できる。特に下記項目について留意して設定する。

必須	設定項目	設定値	備考
◎	PROJECT_NAME	"サンプル Project"	プロジェクト名またはプログラム名を記述する
◎	OUTPUT_LANGUAGE	Japanese	日本語出力する
◎	OPTIMIZE_OUTPUT_FOR_C	YES	C/C++言語ではYESに設定
	PROJECT_NUMBER	1.1	バージョン情報が必要であれば記述する
△	RECURSIVE	NO or YES	YESにするとDoxygen実行ディレクトリ以下を再帰的に探してドキュメントを生成する
	GENERATE_LATEX	NO	LaTeX出力をしない（時間がかかるので、通常は抑制し、必要時にYESにする）
	HAVE_DOT	NO	graphvizを使用しない（時間がかかるので、通常は抑制し、必要時にYESにする）

コメント規約ルール例

以下の `ファイル` `グローバル関数` `グローバル変数` `来歴` のコメントについては最低限記述する。doxygen コマンド実行でドキュメントを生成する。

ファイル・コメント

ファイルの先頭に、そのファイルの概要説明を記述。

```
/**
 * @file    sample.c
 * @brief   GCC Test sample program
```

```
* @author T.Yokobayashi de JR4QPV
* @date   2020/04/16
*/
```

- 必要があれば、@details に詳細説明を記載。

グローバル関数・コメント

関数の直前に記述。グローバル関数は ***** 行を上下に付加して強調する。

```
/**
*****
*
* @brief   Sample main program
* @param   argc  引数の個数
* @param   argv  引数文字列
* @retval  0   正常
* @retval  1   引数エラー
* @details
*****
*/
int main(int argc, char *argv[])
{
    int i;
```

- 戻り値はretvalで数値を指定せず「@return =0:正常,=-1:異常」などのように簡単に記述してもよい。
- @attentionの注意事項は無ければ省略する。
- 引数と戻り値がない時は「none」と記述する。
@param none
@return none

ファイル内関数・コメント

関数の直前に記述。ファイル内スコープの static関数 は、Doxygenではドキュメント出力されないようだが、ソースをみれば関数の使い方が判るように以下のように記載しておく。

```
/**
* @brief   サブルーチン関数
* @param   mode  モード指定(0 or 1)
* @return   終了コード (=0:正常, ≠0:エラー)
* @details
*   - modeが不正な時は「-1」が戻る
*/
static int sub_function(int mode)
{
    int rc;
```

グローバル変数・コメント

グローバル変数の簡易説明を下記のように記述。

```
T_MYTBL _mytbl[16]; //!< 管理テーブル
```

- //!< は ///< と書いてもよい。
- /*!< Node管理テーブルの実態 */ のよう記述すると、別セクションに独立に説明が出力される。

来歴・コメント

ファイルの末尾に、そのファイルの来歴説明を記述。

```
/**
 * -----
 *
 * @file   main.c
 * History
 * -----
 * - 2018/11/04 New created. (By T.Yokobayashi)
 * - 2020/04/16 Write a Doxygen comment.
 */
```

- @fileの記述がキーとなるので、先頭に記述したファイル名と必ず一致するようにする。
- 最低限以下の情報は記載する。細かい変更はGitで管理できる。
「新規作成日・作成者」, 「大きな機能変更」, 「大きなバグ修正」

その他のコメント

覚え書き

下記記述で、関数の使用法などの留意点などを残す。

```
@note メモメモ
```

残作業

下記記述で、残作業をドキュメントに残す。

```
///@todo 残作業メモ
```

バグ

下記記述で、バグ情報をドキュメントに残す。

```
///@bug バグ情報メモ
```

補足

- 説明文の記述には[Markdown](#)記法が使える。なので、「##」で見出し、「-」文字で箇条書きになる。
- その他にも色々なコマンドがある。

参考

1. [Doxygenコメント規約例](#)

Rules

From:

<https://yokoweb.net/dokuwiki/> - 新石器Wiki

Permanent link:

<https://yokoweb.net/dokuwiki/programing/doxygen/doxygen-coding-rules-c>

Last update: **2026/07/28 13:16**

