

Doxygen で C/C++コードを文書化する

TN0004-00 / 2023 年 11 月 05 日 / こがねさん(著)

<https://www.kumikomist.com/>

■目次

1. はじめに.....	2	5.2. Doxygen の設定.....	12
2. Doxygen とは.....	2	6. Doxygen コメントの記入例.....	13
3. インストール.....	3	6.1. よく使うコマンド.....	13
3.1. 必要なもの.....	3	6.2. ファイルのコメント例.....	13
4. Doxygen の推奨設定.....	4	6.3. 関数のコメント例.....	14
4.1. Wizard 設定.....	4	6.4. 変数のコメント例.....	14
4.2. Expert 設定.....	7	6.5. その他のコメント.....	14
5. CSS 設定.....	10	7. Doxygen の実行.....	15
5.1. CSS.....	10	7.1. Doxygen の実行結果.....	16

■文書内の記号

	取り扱いにおける禁止事項（してはイケないこと）を示しています。
	取扱における指示事項（必ずしなければイケないこと）を示しています。

	取り扱いにおける注記事項を示しています。
	取り扱いにおけるポイントを示しています。

1. はじめに

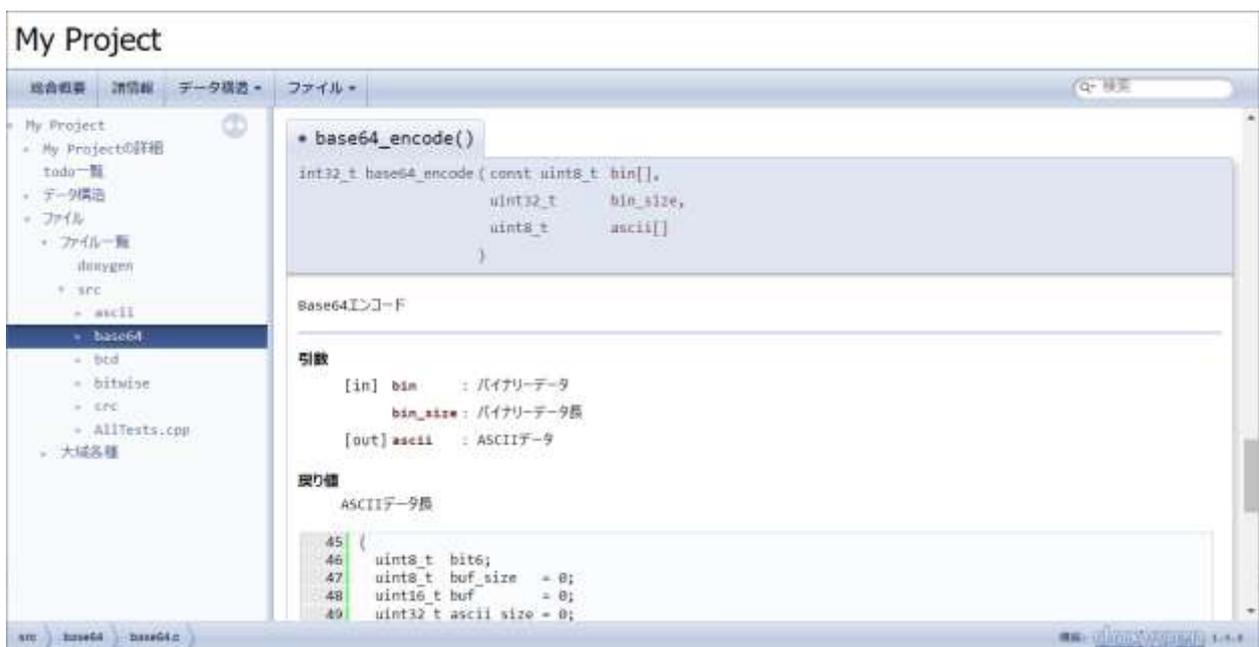
Doxygen は、C/C++や Java、Python などソースコードをドキュメント化するツールです。ソースコードに規定された形式でコメントを記述することで、HTML や LaTeX、RTF ファイルを生成してくれます。これによりソースコードとドキュメントの一貫性を保つことが容易になります。

2. Doxygen とは

たとえばソースコードに下記のようなコメントを残すことで、

```
/** -----
 * @brief Base64 エンコード
 *
 * @param [in] bin : バイナリーデータ
 * @param bin_size : バイナリーデータ長
 * @param [out] ascii : ASCII データ
 * @return ASCII データ長
 */
int32_t base64_encode(const uint8_t bin[], uint32_t bin_size, uint8_t ascii[])
{
    ...
}
```

下図のような HTML ファイルを作成できるのが Doxygen です。ファイル一覧、変数一覧、関数一覧などを確認できます。



実際のところ、Doxygen にはそれほどソースコードの維持・管理に役立つような効果はありません。しかし Doxygen のルールに倣うことで、皆が統一したルールにしたがってコメントを記述できるようになる効果は期待できます。

ソースコードだけでなく、コメントの書き方も人それぞれです。それを誰かの記述一本に揃えようとする揉めるかもしれません。しかしこういった第三者機関やオープンソースなど、ある程度知名度のあるルールにしたがうとなれば揉めにくいかもしれません。

3. インストール

3.1. 必要なもの

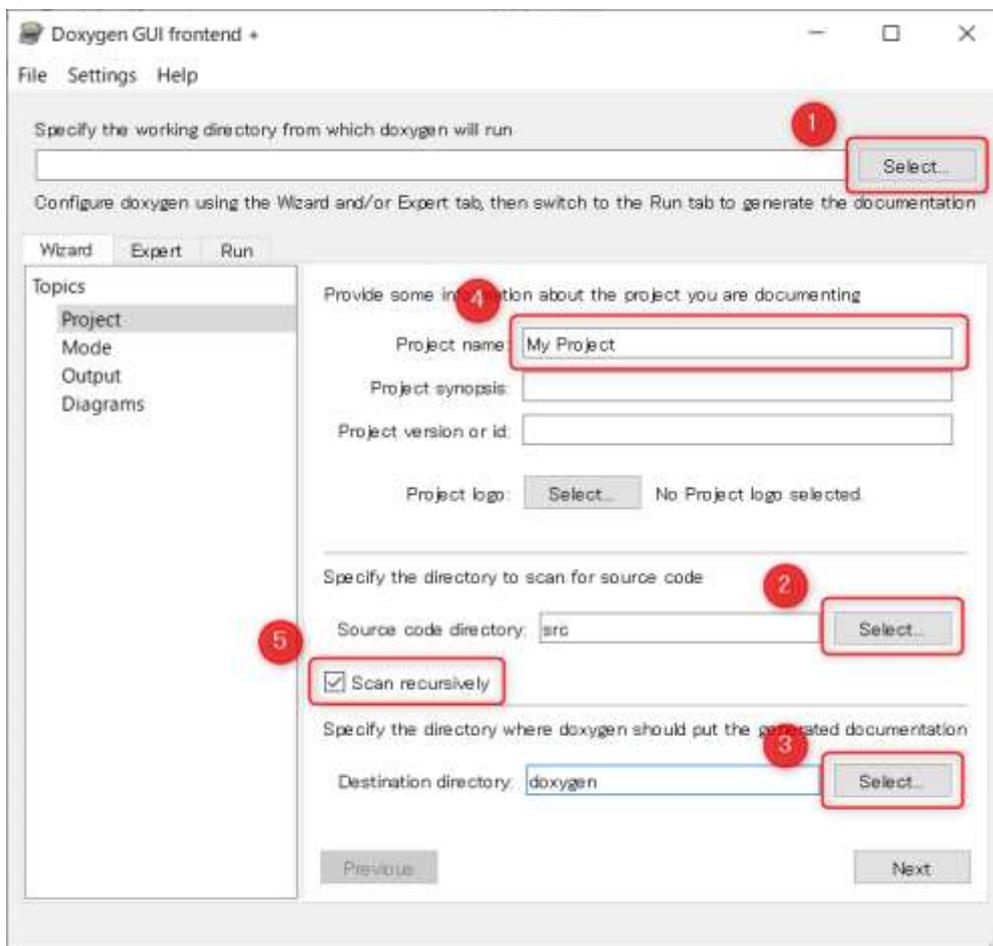
- Doxygen : <https://www.doxygen.nl/download.html>
- Graphviz : <https://graphviz.gitlab.io/download/>

4. Doxygen の推奨設定

Doxygen は英語版しかなく、また設定項目が多くてよくわからないため、おすすめの設定を紹介します。

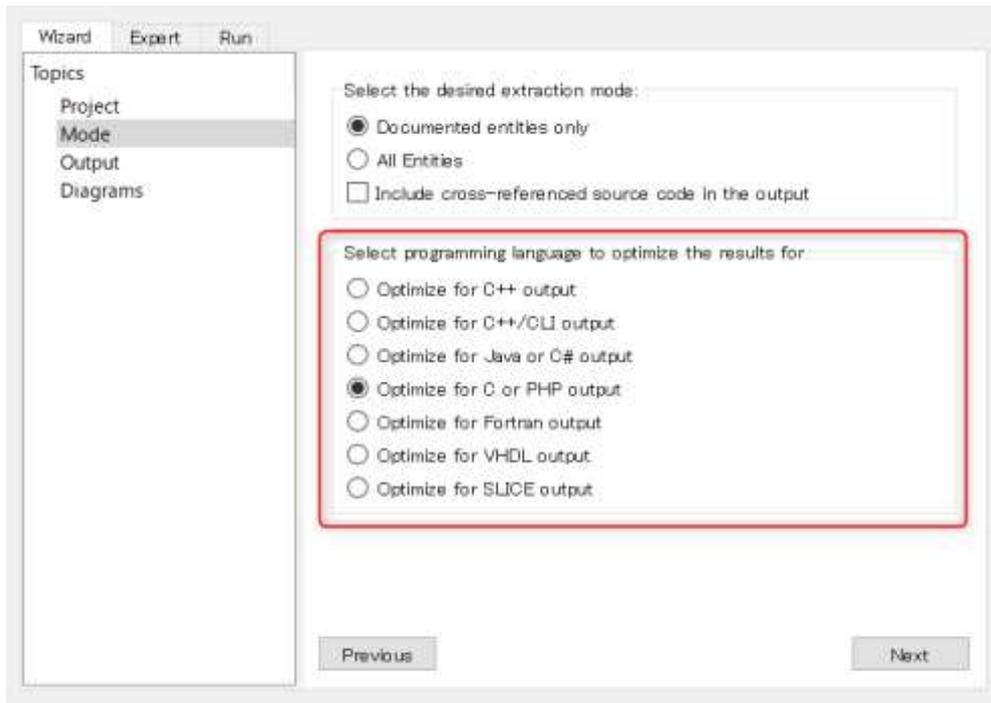
4.1. Wizard 設定

- (1) 最初はプロジェクトに関する設定です。
 - ①～③はフォルダー設定です。
 - ①project
 - └ ②src：ソースファイルのあるフォルダー
 - └ ③doxygen：生成したファイルを保存するフォルダー^(※1)
 - ④プロジェクト名を入力します。
 - ⑤サブフォルダーのファイルもドキュメントの生成対象にします。



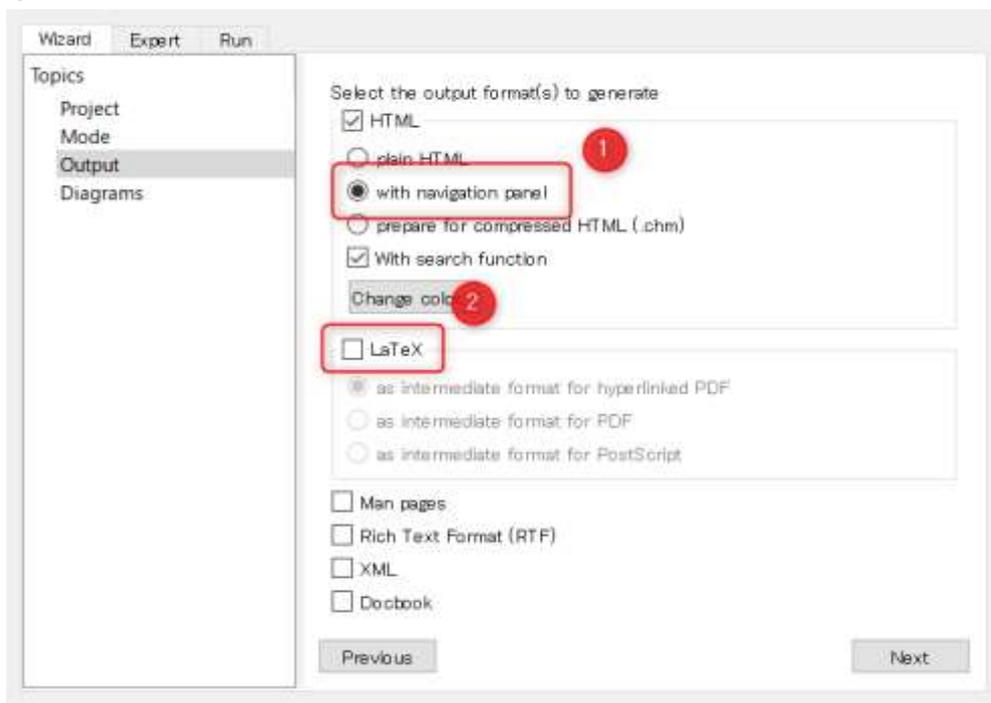
- パスに全角文字があると、ドキュメント生成が上手くいきません。

(2) ドキュメント化する対象のプログラミング言語を選択します。



(3) ①ドキュメントにサイドメニューを追加します。

②LaTeX 出力は不要なのでチェックを外します。



- (4) ①②関係図（図 1 参照）を描画する場合はチェックを付けます。
※Graphviz をインストールしている必要があります。

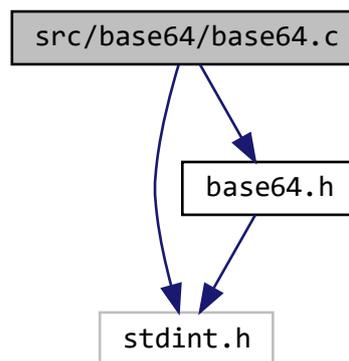
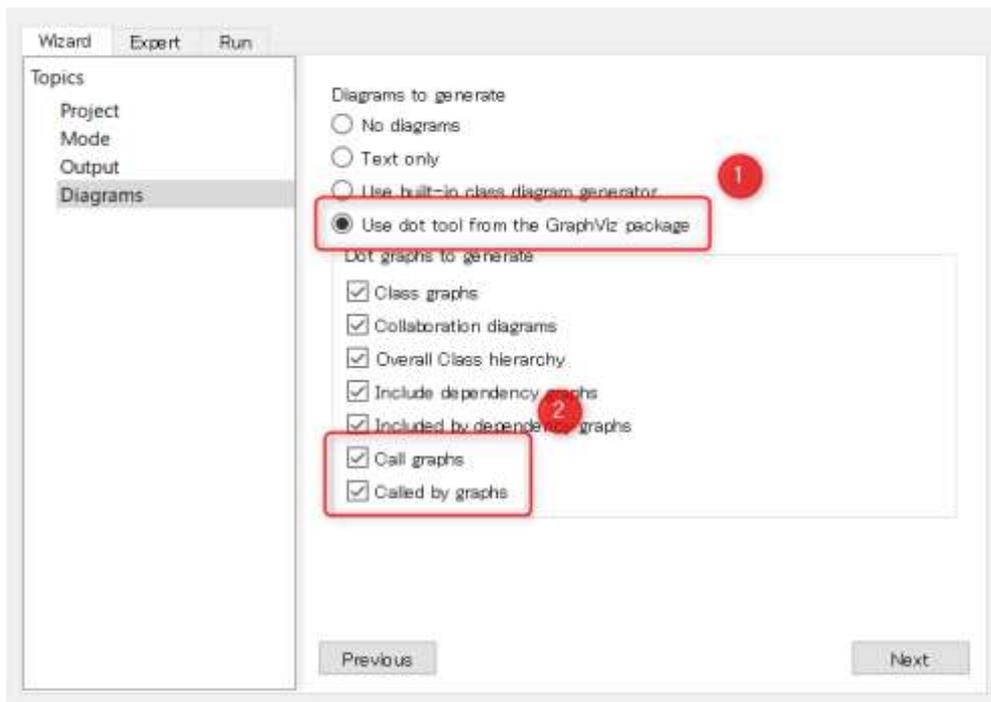
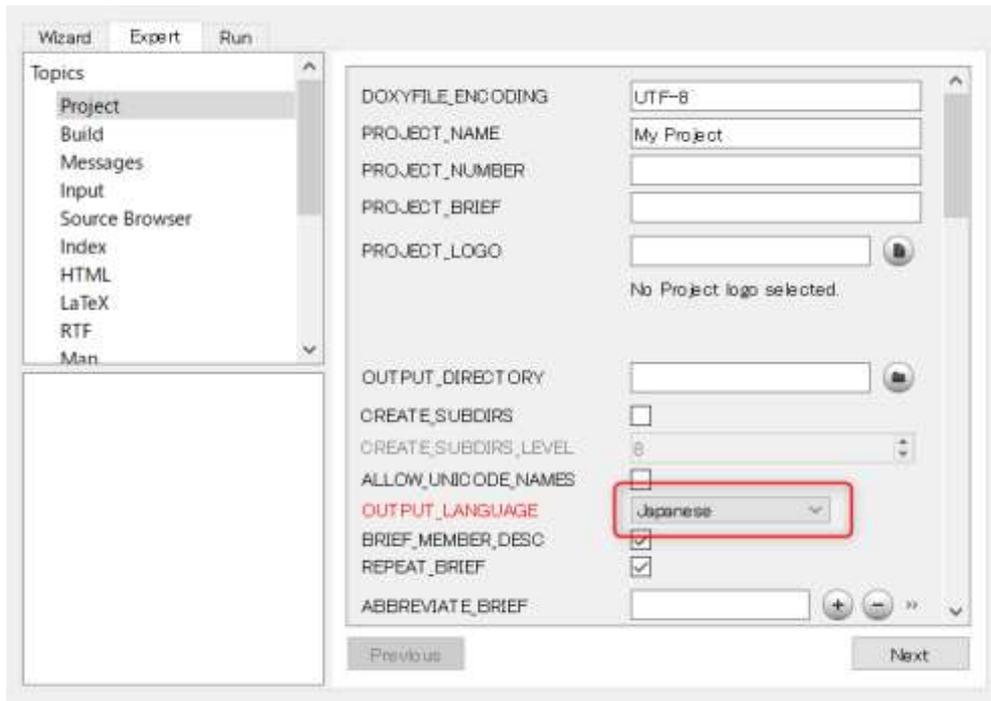


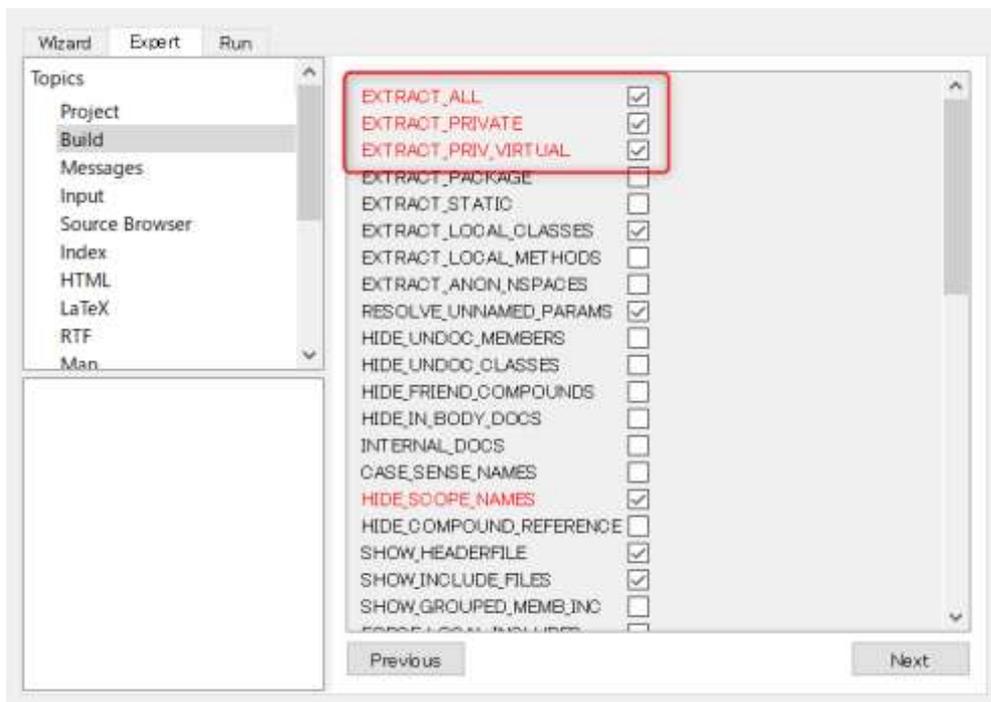
図 1 関係図サンプル

4.2. Expert 設定

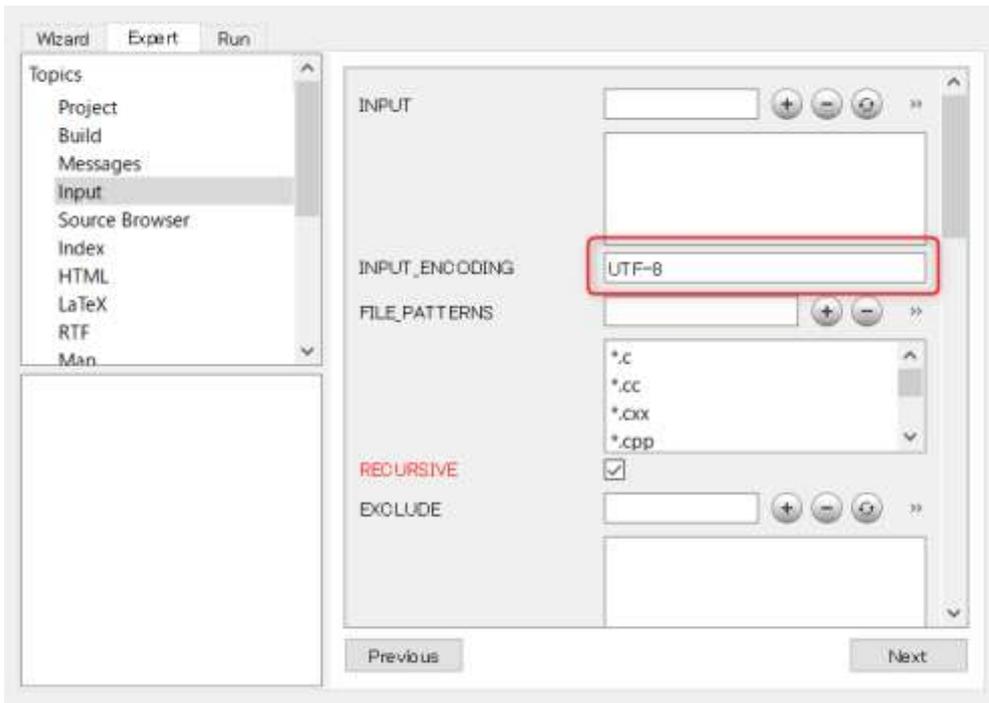
- (1) 生成後ドキュメントの表示言語を選択します。



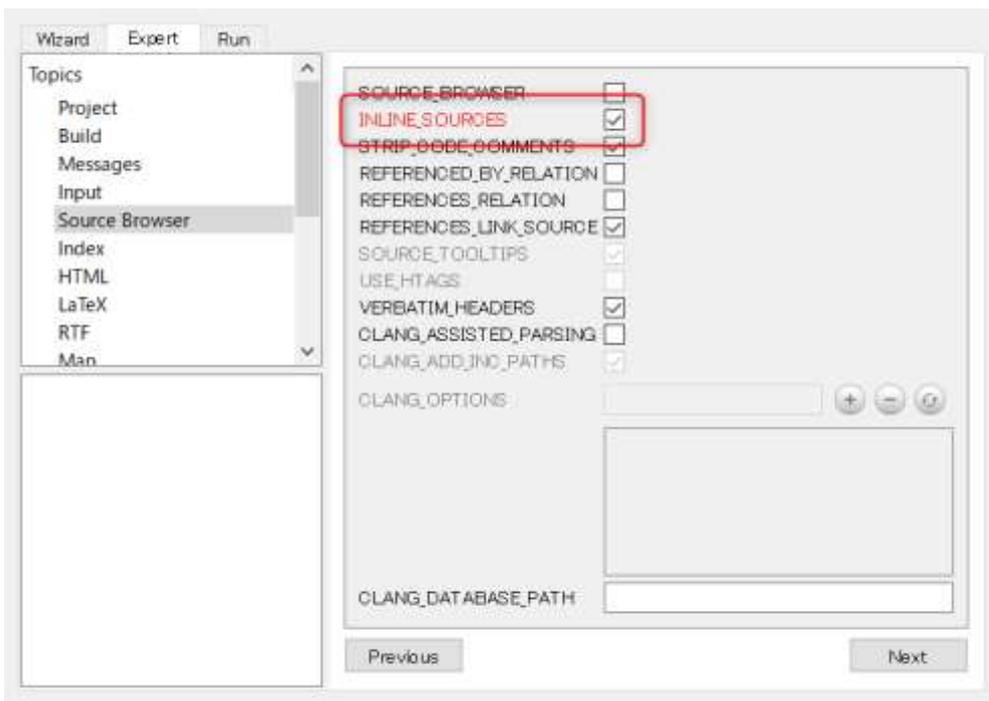
- (2) すべての変数を生成対象にします。



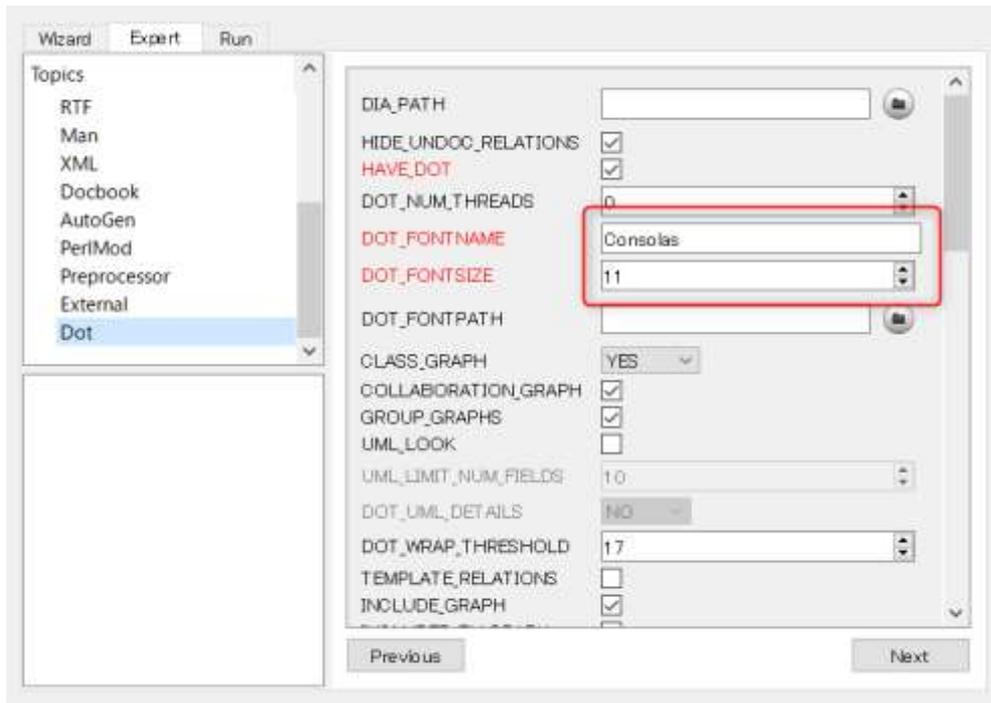
- (3) ソースコードのエンコードを設定します。Shift-JIS なら「CP932」と入力してください。



- (4) ソースコード付きでドキュメントを生成します。

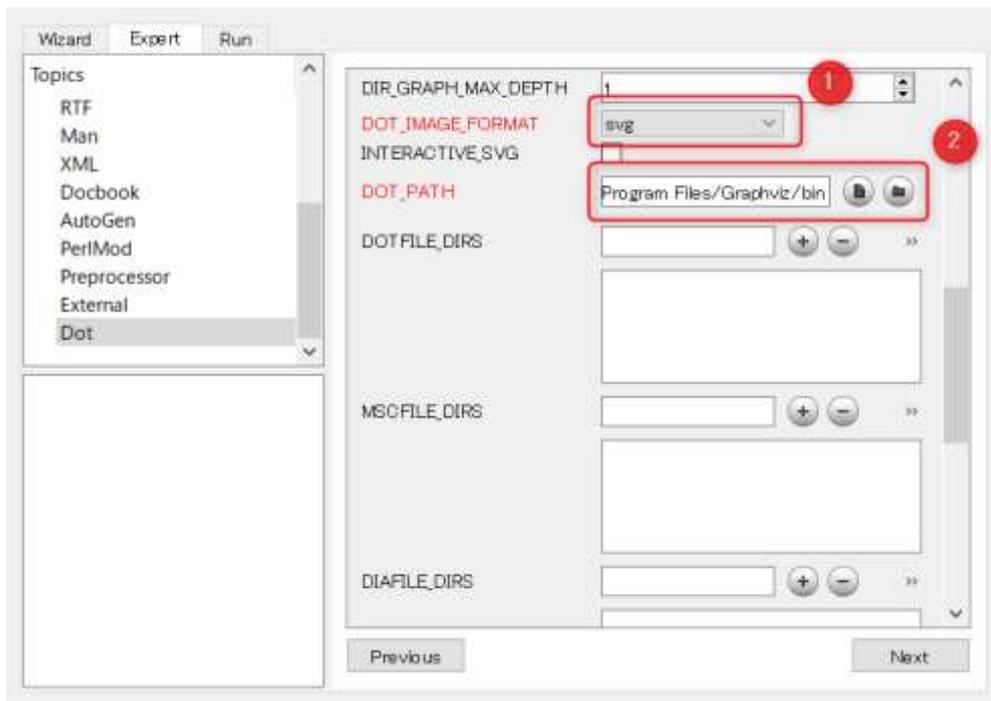


(5) 関係図のフォントとフォントサイズを設定します。



(6) ①関係図のフォーマットを設定します。SVG がおすすめです。

②関係図を描画する際は、Graphviz へのパスを設定します。



5. CSS 設定

Doxygen の生成する HTML は少し見づらいため、自分用に CSS をカスタムしています。その設定内容を紹介します。

5.1. CSS

本例は「doxygen.css」を直接触るのではなく、追加で CSS ファイルを作成して登録します。元々のデザインが悪いわけではないので、主には下記の点を少し修正したぐらいです。

- 大きすぎる文字を小さくする。
- 小さすぎる文字を大きくする。
- 英数字のフォントを Consolas にする。

[mystyle.css]

```
body, table, div, p, dl {
  font-family: Consolas, 'Meiryo UI', monospace;
  font-size: 98%;
}

/* プロジェクト名 */
#projectname {
  font-size: 200%;
}

/* 左のツリー構造のナビ */
#nav-tree .label {
  font-family: Consolas, 'Meiryo UI', monospace;
  font-size: 100%;
}

/* ページタイトル */
.title {
  font-family: Consolas, 'Meiryo UI', monospace;
  font-size: 160%;
  font-weight: normal;
}

h1 {
  border-bottom: 1px solid #879ECB;
  color: #354C7B;
  font-size: 140%;
  font-weight: normal;
  margin-top: 40px;
  padding-bottom: 4px;
}

h2.memtitle {
  font-family: Consolas, 'Meiryo UI', monospace;
  font-size: 120%;
  line-height: 100%;
}
```

```
/* 目次 */
div.toc h3 {
  font-family: Consolas, 'Meiryo UI', monospace;
  font-size: 100%;
  line-height: 100%;
}

div.toc li {
  font-family: Consolas, 'Meiryo UI', monospace;
  font-size: 100%;
  line-height: 100%;
}

/* 本文 */
.textblock {
  line-height: 150%;
}

table.directory {
  font-family: Consolas, 'Meiryo UI', monospace;
}

div.summary {
  font-size: 100%;
}

/* 関数プロトコル */
.memproto {
  padding: 2px;
}

/* ソースコード */
div.line {
  font-family: Consolas, 'Meiryo UI', monospace;
  font-size: 100%;
  line-height: 120%;
}

/* 引数・戻り値 */
.params, .retval, .exception, .tparams {
  font-family: Consolas, 'Meiryo UI', monospace;
}

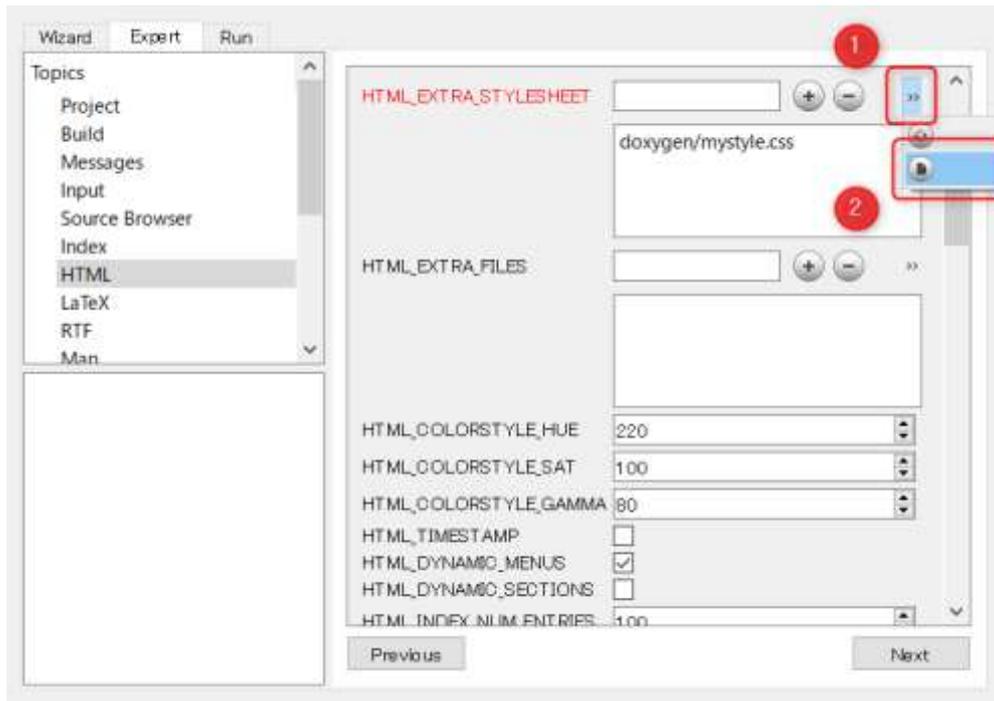
/* include */
code {
  font-family: Consolas, 'Meiryo UI', monospace;
}

/* extern */
span.mlabel {
  font-size: 90%;
}

/* 表示階層 */
.directory .levels {
  font-size: 100%;
}
```

5.2. Doxygen の設定

作成した CSS を、HTML_EXTRA_STYLESHEET に登録すると反映されます。



6. Doxygen コメントの記入例

6.1. よく使うコマンド

Doxygen には数多くのコマンドがあります。しかしこれらすべてのコマンドを覚える必要はありません。コメントを記入するために時間を取られては本末転倒です。ここでは最低限、覚えておくと便利なコマンドを紹介します。

表 1 Doxygen のよく使うコメント

コマンド	内容
@brief {概要}	関数や変数などの概要を記入します。基本的には 1 文で表せられる程度に抑えると良いでしょう。
@details {詳細}	概要で記載しきれない補足説明を記載します。コメントを途中で改行したい場合は、「@n」コマンドを挿入してください。
@file {ファイル名}	コメントを記載しているファイルの名前を入力します。
@n	本コマンドを入れた地点でコメントを改行します。HTML で「 」を入れるのと同じです。
@param [入出力] {引数名} {引数の説明}	関数の引数に関する情報を記入します。[入出力]枠は省略することもできますが、[in]で入力、[in,out]で入出力、[out]で出力引数であることを明示的に表します。
@return {戻り値の説明}	関数の戻り値がどういった内容のものかを表します。戻り値で関数の成否を表すような場合は、「@retval」コマンドを使用します。
@retval {戻り値} {戻り値の詳細}	関数の戻り値で成否を表すような場合に使用します。



- より詳細なコメントを調べたいときは、下記のページが参考になります。
<https://cercopes-z.com/Doxygen/list-command-dxy.html>

6.2. ファイルのコメント例

```
/**
 * @file file.c
 * @brief ファイルの説明
 * @details 詳細説明を書くとき
 */
```

6.3. 関数のコメント例

```
/**
 * @brief 文字列のコピー
 * @details s1 に s2 の文字列を n 文字コピーする @n
 * s2 の長さが n より少ない場合、残りを '¥0' で埋める
 *
 * @param [out] s1 : 複写先の文字列型配列
 * @param [in] s2 : 複写する文字列
 * @param n : 複写文字数
 * @return s1 (複写後の文字列)
 */
char *strncpy(char *s1, const char *s2, size_t n) {
    ...
}

/**
 * @brief 文字列の比較
 * @details '¥0'以降の比較は行いません
 *
 * @param [in] s1 : 比較文字列 1
 * @param [in] s2 : 比較文字列 2
 * @param n : 比較文字数
 * @return 比較結果
 * @retval 0 : 一致
 * @retval 正の値 : s1 > s2
 * @retval 負の値 : s1 < s2
 */
int strncmp(const char *s1, const char *s2, size_t n) {
    ...
}
```

6.4. 変数のコメント例

独立した行にコメントを記入する際と、文末にコメント記入際は記載方法が変わるので注意してください。

```
//! ファイル名
char g_file_name[FILE_NAME_MAX];

char g_file_name[FILE_NAME_MAX];    //!< ファイル名

//! @brief ファイル名
//! @details 半角英数字のみ
char g_file_name[FILE_NAME_MAX];
```

6.5. その他のコメント

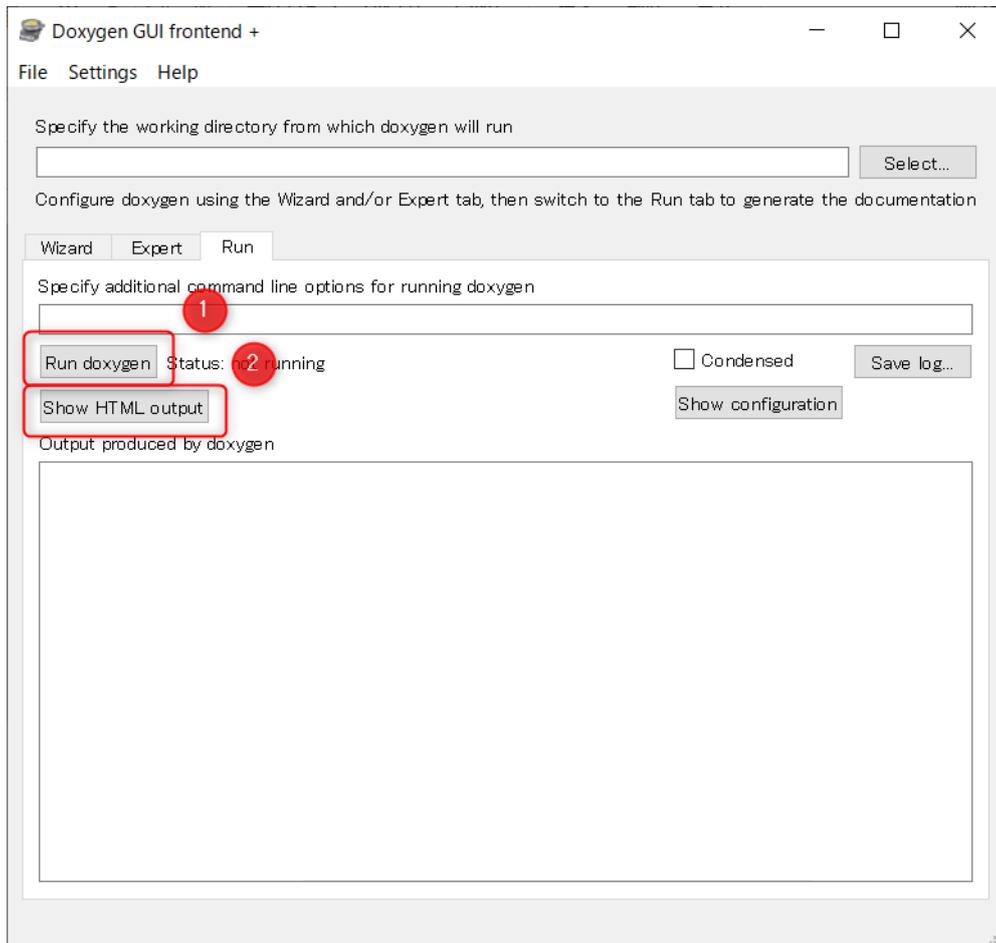
「@todo」や「@bug」コマンドを使用すると、HTML化したときにリスト表示されます。

```
//! @todo 以降の処理はあとで書く

//! @bug ○○のときの挙動がおかしい
```

7. Doxygen の実行

下図①のボタンをクリックすることで、Doxygen のドキュメントを生成します。そのあと②のボタンをクリックすると、生成したドキュメントを表示します。



- 生成した Doxygen ファイルの本体は「`$DOXYGEN%html%index.html`」で、上図②をクリックしたときのページでもあります。

7.1. Doxygen の実行結果

Doxygen の生成ファイルを開くと、図 2 のような画面が表示されます。データ構造やファイルリンクをクリックすることで、ドキュメント化した内容を確認できます。

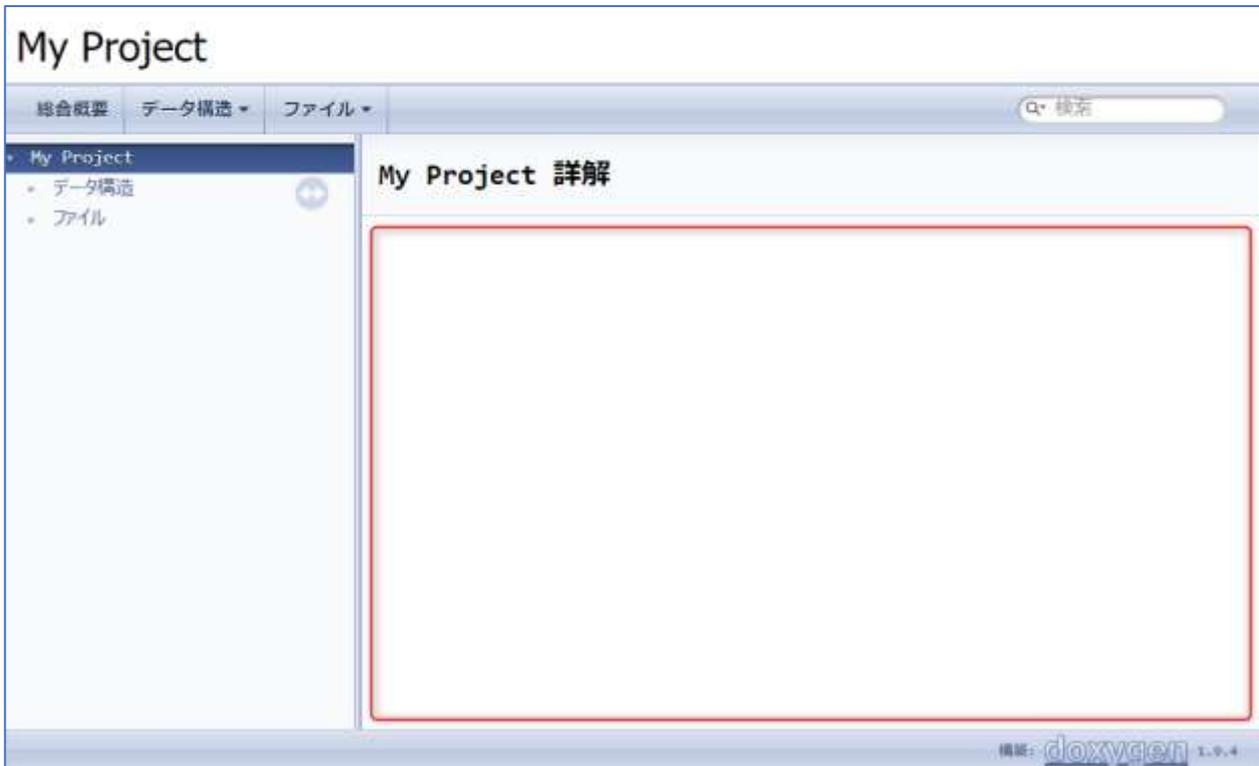


図 2 index.html

ここで図 2 の赤枠部が空欄になっていますが、専用の dox ファイルを作成することで簡単な説明書きなどを掲載することができます。この dox ファイルにも Doxygen のコマンドが適用されます。「6. Doxygen コメントの記入例」を参考に作成してください。

[mainpage.dox]

```
/**
@mainpage My Project の詳細

[TOC]

@section はじめに
Doxygen のメインページに自由に文章を並べることができます。 @n
文章は Doxygen のコメントルールにしたがう必要があります。 @n
Doxygen のコメントについては[こちら](https://cercopes-z.com/Doxygen/list-command-dxy.html)が参考になります。

@section 使用上の注意
@li リスト 1
@li リスト 2
@li リスト 3

@attention
注意点を特別強調して記載することもできます。

@todo
```

やり残したことがあれば、ToDoとして記載しておきましょう。

*/

作成した dox ファイルを図 3 にある INPUT に登録して実行することで、図 4 のように表示することができます。

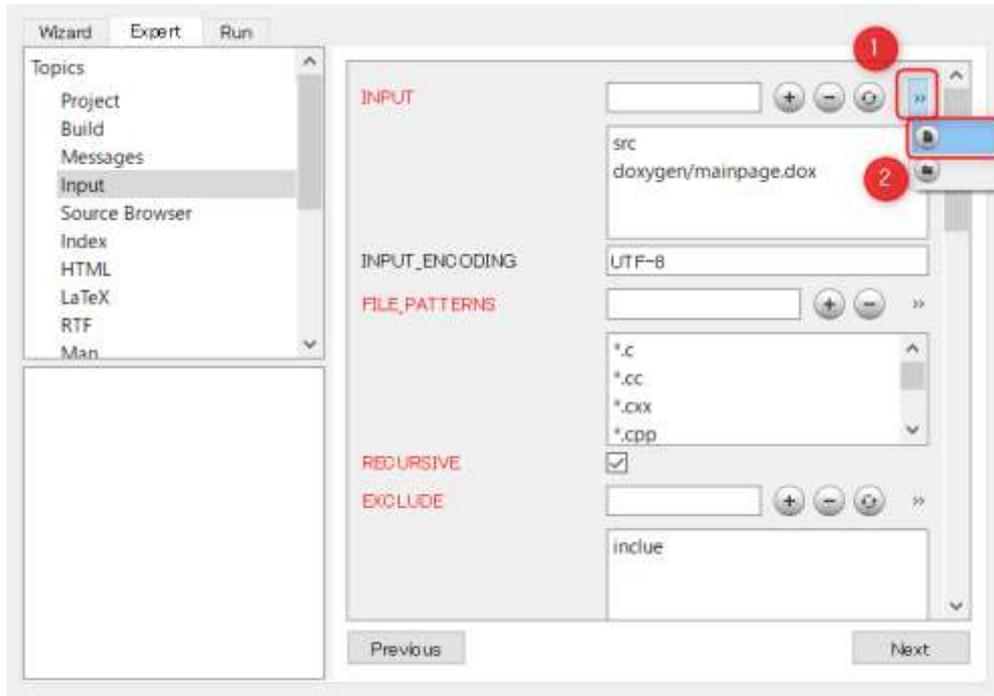


図 3



図 4