

[Домой](#) / [Программирование](#) / [Общее программирование](#) / [Доксиген](#)

Доксиген

Статья автора: [Geoffrey Hunter](#)

Дата публикации: 30 апреля 2013

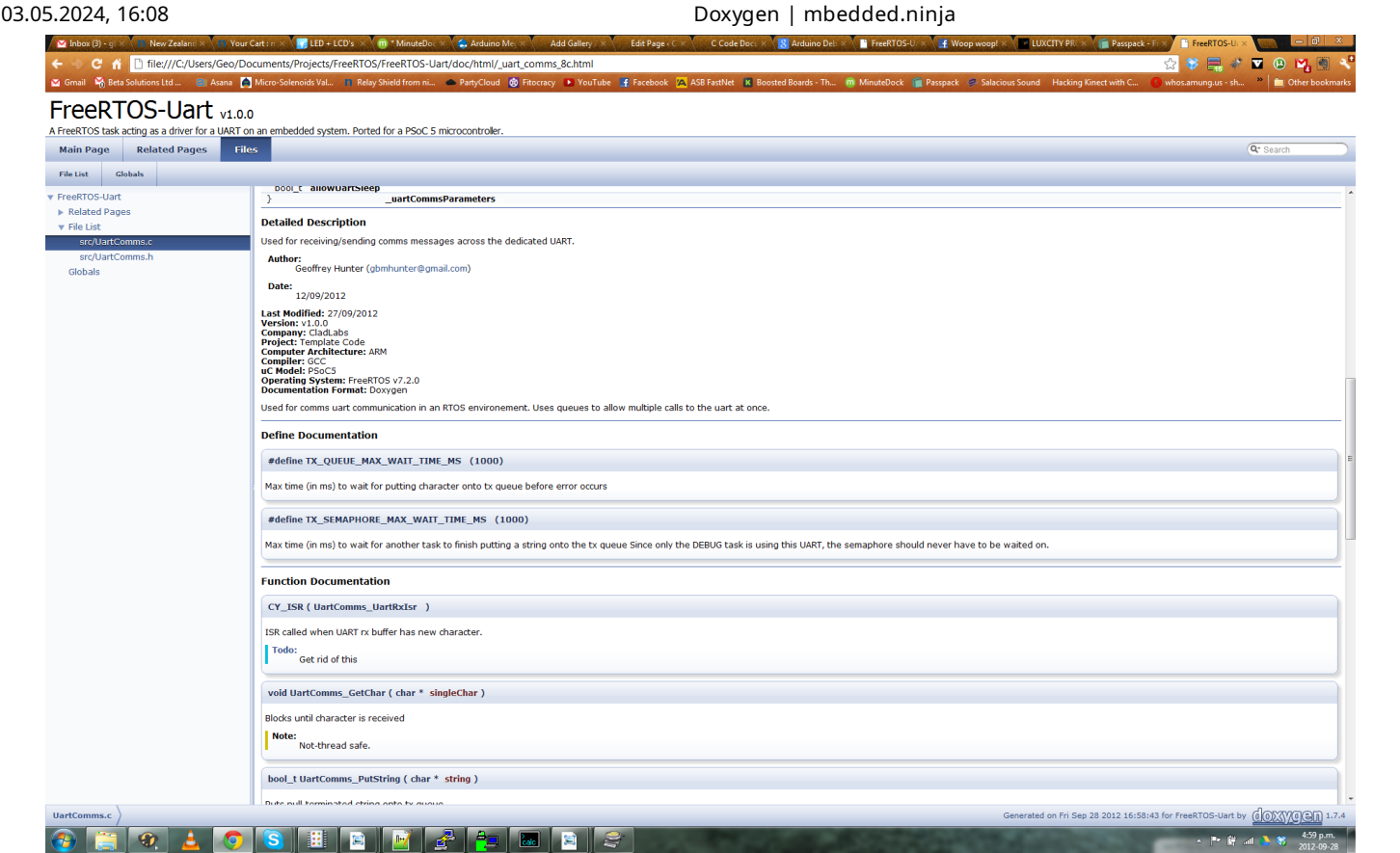
Последнее изменение: 30 апреля 2013

1. Обзор

Doxygen - это программа с открытым исходным кодом для документирования кода. Обычно используется для создания документации из исходных файлов, к которым добавлены специальные идентификаторы, помогающие создавать полезную документацию. Doxygen поддерживает множество языков (включая C, C++, C#, Java, Python, VHDL, PHP и другие ...), однако здесь рассматривается только документация по C / C++. По состоянию на 2016 год считается, что это самый популярный метод документирования кода на C / C++.



По сути, это программа, которая просматривает ваш исходный код и извлекает информацию о функциях, переменных, перечислениях, определениях и почти обо всем остальном, включая специальные комментарии и идентификаторы, которые вы размещаете в коде, и компилирует ее в красивый документ html, pdf или latex. С помощью правильных команд вы можете создать настраиваемую главную страницу, связывающую похожие функции, автоматические списки ошибок / задач, вставлять уравнения latex и многое другое!



Скриншот html-документации, которую Doxygen генерирует с помощью правильно прокомментированного кода.

2. Краткий справочник по Doxygen

Краткий справочник по наиболее популярным ключевым словам Doxygen для документирования исходного кода:

Команда	Описание
@brief	Содержит краткое описание объекта (переменная / функция / перечисление / структура / определение), которое будет отображаться рядом с именем объекта в индексе в верхней части страниц, создаваемых Doxygen.
@подроб	Более подробное описание объекта, которое появится под индексом в верхней части страницы.

Команда	Описание
н	
@параметры	Используется для описания параметров (аргументов) в функции.
@возвращает	Используется для описания возвращаемого значения из функции (если таковая имеется). Имеет ту же функциональность, что и @return .
@sa	Используется для обозначения других объектов (сокращение от "смотри также").
@публичный	Определяет объект как общедоступный. Полезно в программировании на C, поскольку видимость обычно не является неявной.
@личное	Определите объект как закрытый. Полезно в программировании на C, поскольку видимость обычно не является неявной.
@todo	Я добавляю это везде, где хочу вернуться и что-то изменить позже. Doxygen автоматически находит эти заметки и объединяет их на странице "Что нужно сделать" в документации!
@отладка	Я добавляю это везде, где я добавил код, предназначенный специально для отладки.

3. Как использовать Doxygen

Чтобы использовать Doxygen, вы сначала размещаете соответствующие комментарии к документации в исходных файлах. Комментарии должны быть сделаны с помощью специального оператора комментариев, который распознает Doxygen. Обычно комментарии к документации

размещаются непосредственно перед объектом (например, функцией, переменной), который вы хотите задокументировать. Затем, чтобы сгенерировать документацию, запустите Doxygen из командной строки в Linux или Doxywizard в Windows, и это создаст некоторую документацию в формате HTML или Latex (PDF) в хорошем формате.

```
1    // Normal C comment. Doxygen will ignore this
2    /// Comment Doxygen will recognise that tells you something
3    /// about DocumentedFunction()
4    void DocumentedFunction() {
5        // function code...
6    }
```

4. Пример заголовка файла

Обычно я использую следующие заголовки файлов при документировании с помощью Doxygen.

```
1    ///
2    /// @file    DoxygenExample.h
3    /// @author    Geoffrey Hunter (gbmhunter@gmail.com)
4    /// @edited    n/a
5    /// @date    12/09/2012
6    /// @brief    Doxygen example file.
7    /// @details
8    /// <b>Last Modified: </b> 07/11/2012\n
9    /// <b>Version: </b> v1.0.0\n
10   /// <b>Company: </b> CladLabs\n
11   /// <b>Project: </b> Doxygen Examples\n
12   /// <b>Language: </b> C++\n
13   /// <b>Computer Architecture: </b> ARM\n
14   /// <b>Compiler: </b> GCC\n
15   /// <b>uC Model: </b> PSoC5\n
16   /// <b>Operating System: </b> FreeRTOS v7.2.0\n
17   /// <b>Documentation Format: </b> Doxygen\n
18   /// <b>License: </b> GPLv3\n
19   ///
```

ПРИМЕР: Полный рабочий пример Doxygen на C++ можно найти по адресу <https://github.com/gbmhunter/CppTemplate>. Конфигурационный файл Doxygen создает документацию для этого проекта шаблона C++, а выходные данные HTML-документации можно просмотреть на страницах GitHub по адресу <https://gbmhunter.github.io/CppTemplate/index.html>.

5. Главная страница

При первом использовании Doxygen вы открываете только что созданную документацию и обнаруживаете, что первая страница (она же главная страница или целевая страница) пуста. Это потому, что Doxygen разработал его таким образом, и чтобы поместить в него что-то еще, вы должны специально сообщить Doxygen об этом. Это можно сделать с помощью `@mainpage` команды. В следующем примере добавлена главная страница с двумя разделами.

```
1  /*!  
2  
3      @mainpage  
4  
5          This is the main page  
6  
7      @section section1 Section 1  
8  
9          This is section 1 on the main page  
10  
11     @section section2 Section 2  
12  
13         This is section 2 on the main page  
14  
15     */
```

📖 ПРИМЕЧАНИЕ

Этот код для создания главной страницы не обязательно должен быть каким-то особенным, он может быть в одном из файлов, которые

Doxygen анализирует при создании документа.

6. Doxygen “Экранирует” блоки комментариев

Doxygen поддерживает ряд “экранирований”, способов сигнализации о том, что комментарий к коду должен быть проанализирован движком Doxygen.

```
1    /**
2        * A JavaDoc style Doxygen comment block
3        */
4
5    /*!
6        * A Qt style Doxygen comment block
7        */
8
9    ///
10   /// Another Doxygen comment block
11   ///
12
13   //!
14   //! And another Doxygen comment block!
15   !!
```

Все приведенные выше блоки комментариев обрабатываются Doxygen идентично.

Также предусмотрено размещение документации после объектов, таких как переменные и параметры функции. Это можно сделать, добавив `<` в конец управляющей последовательности Doxygen, например:

```
1    int myVar; /*!< This is my variable! */
2
3    int myVar; /**< This is my variable! */
```

```
4
5     int myVar; //!< This is my variable!
6
7     int myVar; ///< This is my variable!
```

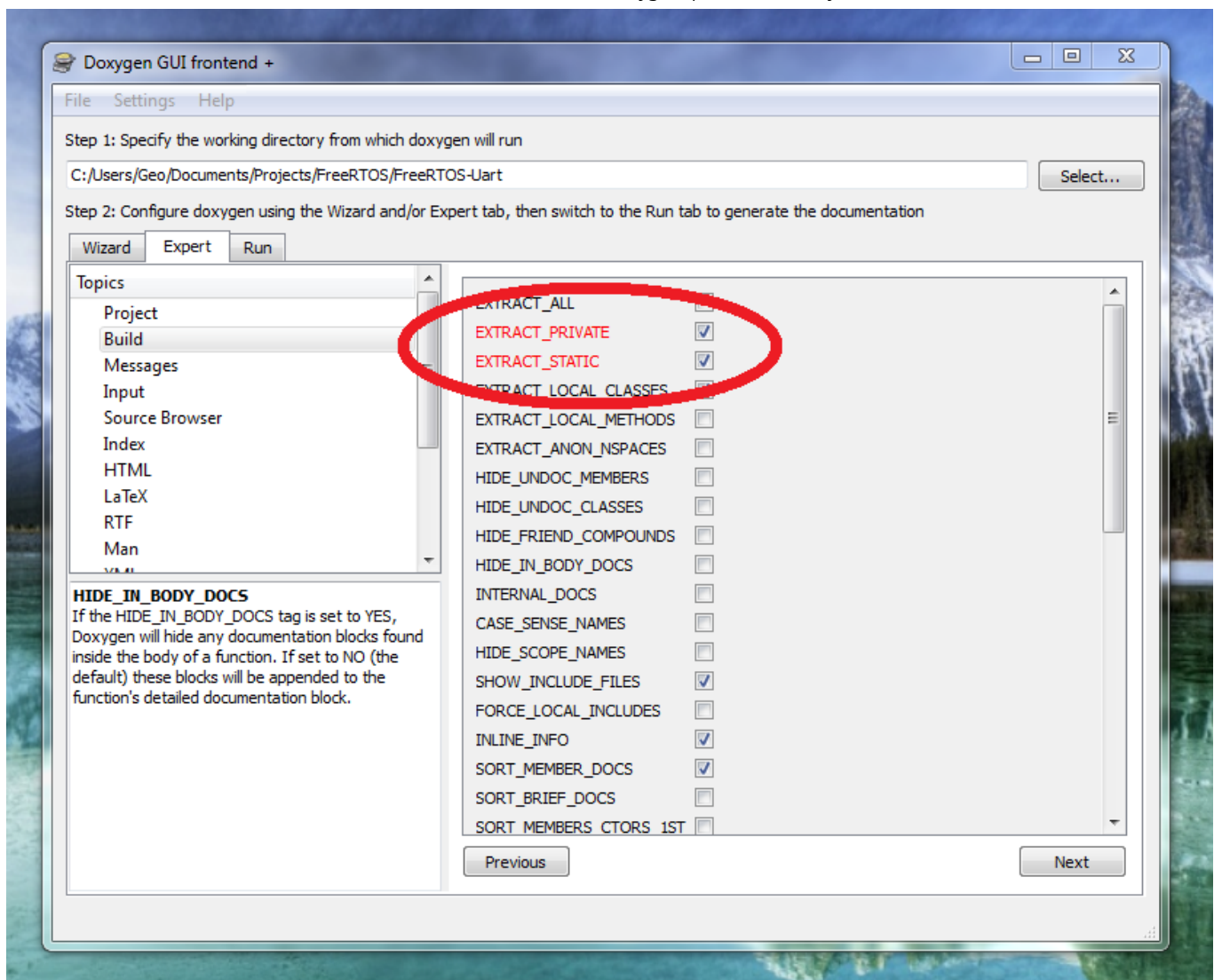
7. Группы

Группы хороши для ... объединения вещей вместе.

```
1     // Groups
2     //!< \defgroup TestGroup
3     //!< @{
4
5     // Anything in here will belong to TestGroup
6
7     void FuncInTestGroup() {
8         // Code
9     }
10
11     //!< @}
```

8. Настройки Doxywizard

Doxywizard - это графический интерфейс Windows для использования Doxygen. Настройка `EXTRACT_PRIVATE = 1` и `EXTRACT_STATIC = 1` приводит к тому, что doxygen включает в документацию все объекты с комментариями, а не только те, которые он считает общедоступными или доступными из других файлов (статические). Я считаю, что это гораздо полезнее, чем исключать их, поскольку без присутствия этих объектов в документации читатель может задаться вопросом, как вообще работает ваш код (C не имеет приятной структуры открытого интерфейса, которую имеет объектно-ориентированное программирование). На следующем рисунке показаны два параметра, выбранные при использовании графического интерфейса Doxygen (Doxywizard).



Выбираем 'EXTRACT_PRIVATE' и "ИЗВЛЕЧЬ СТАТИЧЕСКИЙ файл" в графическом интерфейсе Doxygen.

Обязательно сохраните файл конфигурации (doxyfile, без расширения) где-нибудь вместе с исходным кодом, чтобы вы могли запустить его снова, и doxygen запомнит настройки.

Авторы



Джеффри Хантер

Чувак что-то мастерит.

[Twitter](#)

[GitHub](#)

[Поддержи меня в Ko-fi](#)



Эта работа лицензирована по [международной лицензии Creative Commons Attribution 4.0](https://creativecommons.org/licenses/by/4.0/).

Контент по теме

- [Шпаргалка CMake](#)
- [pybind11](#)
- [Код Visual Studio](#)
- [Профилирование GCC](#)
- [Python извлекает привязки из C / C ++](#)

Теги

- [Доксиген](#)
- [код](#)
- [комментарии](#)
- [C](#)
- [C++](#)
- [Документация](#)
- [API](#)
- [Докси-волшебник](#)

0 Comments

 Login

G

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name



Share

Best Newest Oldest

Be the first to comment.

[Subscribe](#) [Privacy](#) [Do Not Sell My Data](#)

Авторские права © 2023 mbedded.ninja. Создан с помощью Hugo.