

Introducción a doxygen

Parte III: Uso del comando doxygen



Practique los ejemplos en Gitpod



Open in Gitpod

Disponible en



¡Únete al grupo en Telegram!



Objetivos de esta introducción



Ideas para explicar Doxygen

- ▶ Utilizar la documentación de Doxygen.
- ▶ Explicar el uso del manual y la configuración.

```
gitpod ~/dune-basics $ tldr doxygen
```

doxygen

La documentación del sistema para varios lenguajes se puede encontrar en:

<http://www.doxygen.nl>.

- Para generar un archivo o plantilla de configuración utilice el comando '[Doxyfile](#)':
`doxygen -g`
- Para generar un archivo o plantilla de configuración con un nombre específico '[MiDoxygen](#)':
`doxygen -g MiDoxygen`
- Genera una plantilla de un archivo de configuración:
`doxygen -g ruta/a/archivo_de_configuracion`
- Genera documentación usando un archivo de configuración existente:
`doxygen ruta/a/archivo_de_configuracion`

En caso de no tener instalado doxygen en Arch Linux o derivados el programa utilice el comando:

```
gitpod ~/dune-basics $ yay -Sy doxygen --noconfirm
```

```
gitpod ~/dune-basics $ doxygen -g MiDoxigen
```

Configuration file '[MiDoxigen](#)' created.

Now edit the configuration file [and](#) enter

```
doxygen MiDoxigen
```

to generate the documentation [for](#) your project

El archivo MiDoxygen es un archivo de texto plano con alrededor de 2600 líneas, que contiene la configuración, su estructura es del tipo clave valor, con líneas comentadas por #, a continuación explicaremos algunas de ellas para hacer una configuración básicas, en caso de necesitar más información dirijase a la página [3].

El PROJECT_NAME puede ser una palabra simple o una secuencia de palabras
entre comillas dobles (a menos que usted utilice Doxywizard) esto es para
identificar el proyecto para el que se va a generar la documentación.
Este nombre es usado en el título de la mayoría de las páginas generadas y en pequeños otros lugares.
El valor por defecto es: My Project, y se le puede dar el nombre de su proyecto.

PROJECT_NAME = "Dune-project-cpp-review"

Si se utiliza el programa doxywizard puede configurar los parámetros del archivo doxyfile.

The screenshot shows the Doxywizard application window with the following elements:

- Menu Bar:** File, Settings, Help
- Working Directory:** A text field with a "Select..." button.
- Instructions:** "Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation"
- Tabs:** Wizard (selected), Expert, Run
- Left Panel (Topics):**
 - Project (selected)
 - Mode
 - Output
 - Diagrams
- Main Content Area:**
 - Section:** Provide some information about the project you are documenting
 - Fields:**
 - Project name: My Project
 - Project synopsis: (empty)
 - Project version or id: (empty)
 - Project logo: Select... No Project logo selected.
 - Section:** Specify the directory to scan for source code
 - Fields:**
 - Source code directory: /home/john/Github/example
 - Scan recursively: ☐
 - Section:** Specify the directory where doxygen should put the generated documentation
 - Fields:**
 - Destination directory: (empty)
- Buttons:** Previous, Next

Usando doxygen con duneproject

Una vez que ya tiene su proyecto dune configurado, automáticamente tendrá el directorio `doc/doxygen` que con el comando `make doxygen_dune-basics` le generará el directorio `build/doc/doxygen/html` con la página de documentación.

```
├─ cmake
│   └─ modules
│       ├── CMakeLists.txt
│       └─ DuneTestMacros.cmake
├─ CMakeLists.txt
├─ config.h.cmake
├─ doc
│   ├── CMakeLists.txt
│   └─ doxygen
│       ├── CMakeLists.txt
│       └─ Doxylocal
├─ dune
│   ├── CMakeLists.txt
│   └─ test
│       ├── CMakeLists.txt
│       └─ test.hh
├─ dune.module
├─ dune-test.pc.in
├─ README
├─ src
│   ├── CMakeLists.txt
│   └─ dune-test.cc
```

Aeiou.

cmake

CMakeLists.txt

doc

dune

dune.module

dune-test.pc.in

README

src

El archivo doc/doxygen/Doxylocal es generado luego de utilizar el comando duneproject

```
# Este archivo contiene cambios locales en la configuración de doxygen
# por favor use '+' para adicionar archivos/directorios a las listas

# El tag INPUT puede ser usado para especificar los archivos y/o directorios que contienen
# los archivos fuentes documentados. Puede entrar archivos como "miarchivo.cpp" o
# directorios como "/usr/src/myproject". Separe archivos o directorios con espacio.

INPUT                += @top_srcdir@dune/ \
                       @srcdir@/bienvenida.txt
# Ve a p.ej dune-grid para ejemplos de la página principal y módulos
# INPUT              += @srcdir@/mainpage \
#                       @srcdir@/modules

# El tag EXCLUDE puede ser usado para especificar archivos y/o directorios que deban
# ser excluidos desde los archivos fuente INPUT. De esta forma puedes fácilmente excluir un
# subdirectorio desde un árbol de directorio cuya raíz es especificado con el tag INPUT.

# EXCLUDE             += @top_srcdir@dune/basics/test

# El tag EXAMPLE_PATH puede ser usado para especificar uno o más archivos o
# directorios que contenga fragmentos de código ejemplo que son incluidas (vea
# el comando \include).

# EXAMPLE_PATH        += @top_srcdir@/src

# El tag IMAGE_PATH puede ser usado para especificar uno o más archivos o
# directorios que contenga imagen que son incluidas en la documentación (vea
# el comando \image).

# IMAGE_PATH          += @top_srcdir@dune/basics/pics
```

Con Doxygen creamos un bloque de comentario en el cuerpo de una función empezando con `/**` y ciertos parámetros señalados con el símbolo `@` como se ve en el siguiente código

Cuadro: Comparación de la línea de comando del gestión de software (fuente: wiki.archlinux.org)

Acción	Arch	Red Hat/Fedora	Debian/Ubuntu
Instala paquetes	<code>pacman -S</code>	<code>dnf install</code>	<code>apt install</code>

Listado 1: Programa saludo.hh.

```
#ifndef SALUDO

#include <iostream>
#include <string>

/**
 * @brief Funciona que te saluda
 *
 * @param nombre
 */

void saludo(std::string nombre)
{
    std::cout << "¡Hola "
               << nombre
               << "!\n"
               << "Bienvenido al tutorial de Doxygen"
               << "\n";
}

#endif // SALUDO
```

Referencias

► Libros



Oliver Sander. *DUNE — The Distributed and Unified Numerics Environment*. First. Lecture Notes in Computational Science and Engineering 140. Springer International Publishing, 2020. ISBN: 978-3-030-59701-6. DOI: 10.1007/978-3-319-03038-8.

► Artículos



Peter Bastian y col. “The Dune framework: Basic concepts and recent developments”. En: *Computers & Mathematics with Applications* 81.1 (1 de ene. de 2021). Development and Application of Open-source Software for Problems with Numerical PDEs, págs. 75-112. ISSN: 0898-1221. DOI: <https://doi.org/10.1016/j.camwa.2020.06.007>. URL: <https://www.sciencedirect.com/science/article/pii/S089812212030256X>.

► Sitios web



Doxygen. *Doxygen Manual*. 11 de abr. de 2021. URL: <https://www.doxygen.nl/manual> (visitado 31-05-2021).