

Sample-project

Generated by Doxygen 1.9.3

1 Introduction	1
1.1 Structure of the directories	1
1.2 Description for each directory or file	1
1.3 Tutorial	2
2 Namespace Index	3
2.1 Namespace List	3
3 Class Index	5
3.1 Class List	5
4 Namespace Documentation	7
4.1 Rex Namespace Reference	7
5 Class Documentation	9
5.1 Rex::Person Class Reference	9
5.1.1 Constructor & Destructor Documentation	9
5.1.1.1 Person() [1/2]	9
5.1.1.2 Person() [2/2]	9
5.1.1.3 ~Person()	10
5.1.2 Member Function Documentation	10
5.1.2.1 SetAge()	10
5.1.2.2 SetName()	10
5.1.2.3 ShowInfo()	10
Index	13

Chapter 1

Introduction

This tutorial is used to build a C++ project with the following software.

- MinGW-w64 is used to compile source codes.
- Visual Studio Code is used as editor.
- CMake is used to build the project.
- Git is used for version control. (How to use Git won't be described here.)
- Doxygen is used to generate documents for the code in html and tex format.
- Miktex is used to generate PDF format documents from tex files generated by Doxygen.

1.1 Structure of the directories

The structure of the directories contained in the project folder is as following.

```
Sample-project
|-- .git/
|-- bin/
|   |-- Sample-project.exe
|-- build/
|-- doc/
|   |-- html/
|   |-- latex/
|   |-- Doxyfile
|-- include/
|   |-- Person.h
|-- source/
|   |-- CMakeLists.txt
|   |-- main.cpp
|   |-- Person.cpp
|-- .gitignore
|-- CMakeLists.txt
|-- README.md
```

1.2 Description for each directory or file

- **Sample-project/**: contains needed files for this project
 - ****.git/****: contains files for Git
 - **bin/**: contains executable files

- * **Sample-project.exe**: the executable file of the project
- **build/**: contains files for building the project
- **doc/**: contains documents
 - * **html/**: documents in html format
 - * **latex/**: documents in tex format from which pdf file can be generated
 - * **Doxyfile**: configuration file generated and used by Doxygen
- **include/**: contains header files
 - * **Person.h**: header for the class Person
- **source/**: contains source files
 - * **CMakeLists.txt**: configuration file for source files
 - * **main.cpp**: main function
 - * **Person.cpp**: implementation for the class Person
- ****.gitignore**: records which directories and files being ignored by Git
- **CMakeLists.txt**: configuration file of the project
- **README.md**: Information about the project

1.3 Tutorial

Step-by-step instructions for constructing a new project.

1. Install all software mentioned in Introduction.
2. Create a folder, named Project for an example, and the following folders inside Project.
 - bin/, doc/, include/, source/
3. Create

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Rex	7
-------------------------------	-------------------

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Rex::Person	9
---------------------------------------	---

Chapter 4

Namespace Documentation

4.1 Rex Namespace Reference

Classes

- class [Person](#)

Chapter 5

Class Documentation

5.1 Rex::Person Class Reference

```
#include <Person.h>
```

Public Member Functions

- [Person](#) ()
Construct a new [Person](#) object.
- [Person](#) (const std::string &Name, const unsigned int &Age)
Construct a new [Person](#) object.
- [~Person](#) ()
Destroy the [Person](#) object.
- void [ShowInfo](#) () const
Show the name and age of the person.
- void [SetName](#) (const std::string &Name)
Set the Name_ object.
- void [SetAge](#) (const unsigned int &Age)
Set the Age object.

5.1.1 Constructor & Destructor Documentation

5.1.1.1 [Person](#)() [1/2]

```
Rex::Person::Person ( )
```

Construct a new [Person](#) object.

5.1.1.2 [Person](#)() [2/2]

```
Rex::Person::Person (
    const std::string & Name,
    const unsigned int & Age )
```

Construct a new [Person](#) object.

Parameters

in	<i>Name</i>	Name of the person
in	<i>Age</i>	Age of the person

5.1.1.3 ~Person()

```
Rex::Person::~~Person ( )
```

Destroy the [Person](#) object.

5.1.2 Member Function Documentation**5.1.2.1 SetAge()**

```
void Rex::Person::SetAge (
    const unsigned int & Age )
```

Set the Age object.

Parameters

in	<i>Age</i>	Age
----	------------	-----

5.1.2.2 SetName()

```
void Rex::Person::SetName (
    const std::string & Name )
```

Set the Name_ object.

Parameters

in	<i>Name</i>	Name
----	-------------	------

5.1.2.3 ShowInfo()

```
void Rex::Person::ShowInfo ( ) const
```

Show the name and age of the person.

Index

~Person
Rex::Person, [10](#)

Person
Rex::Person, [9](#)

Rex, [7](#)
Rex::Person, [9](#)
~Person, [10](#)
Person, [9](#)
SetAge, [10](#)
SetName, [10](#)
ShowInfo, [10](#)

SetAge
Rex::Person, [10](#)

SetName
Rex::Person, [10](#)

ShowInfo
Rex::Person, [10](#)