

Sample-project

Generated by Doxygen 1.9.3



<b>1 Introduction</b>	<b>1</b>
1.1 Structure of the directories	1
1.2 Description for each directory or file	1
1.3 Tutorial	2
<b>2 Namespace Index</b>	<b>3</b>
2.1 Namespace List	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Namespace Documentation</b>	<b>9</b>
5.1 Rex Namespace Reference	9
<b>6 Class Documentation</b>	<b>11</b>
6.1 Rex::Person Class Reference	11
6.1.1 Constructor & Destructor Documentation	11
6.1.1.1 Person() [1/2]	11
6.1.1.2 Person() [2/2]	11
6.1.1.3 ~Person()	12
6.1.2 Member Function Documentation	12
6.1.2.1 SetAge()	12
6.1.2.2 SetName()	12
6.1.2.3 ShowInfo()	12
<b>7 File Documentation</b>	<b>15</b>
7.1 build/CMakeFiles/3.22.3/CompilerIdC/CMakeCCompilerId.c File Reference	15
7.1.1 Macro Definition Documentation	15
7.1.1.1 __has_include	16
7.1.1.2 ARCHITECTURE_ID	16
7.1.1.3 C_VERSION	16
7.1.1.4 COMPILER_ID	16
7.1.1.5 DEC	16
7.1.1.6 HEX	16
7.1.1.7 PLATFORM_ID	17
7.1.1.8 STRINGIFY	17
7.1.1.9 STRINGIFY_HELPER	17
7.1.2 Function Documentation	17
7.1.2.1 main()	17
7.1.3 Variable Documentation	17
7.1.3.1 info_arch	17

7.1.3.2 info_compiler	17
7.1.3.3 info_language_extensions_default	18
7.1.3.4 info_language_standard_default	18
7.1.3.5 info_platform	18
7.2 build/CMakeFiles/3.22.3/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference	18
7.2.1 Macro Definition Documentation	19
7.2.1.1 __has_include	19
7.2.1.2 ARCHITECTURE_ID	19
7.2.1.3 COMPILER_ID	19
7.2.1.4 CXX_STD	19
7.2.1.5 DEC	19
7.2.1.6 HEX	20
7.2.1.7 PLATFORM_ID	20
7.2.1.8 STRINGIFY	20
7.2.1.9 STRINGIFY_HELPER	20
7.2.2 Function Documentation	20
7.2.2.1 main()	20
7.2.3 Variable Documentation	20
7.2.3.1 info_arch	21
7.2.3.2 info_compiler	21
7.2.3.3 info_language_extensions_default	21
7.2.3.4 info_language_standard_default	21
7.2.3.5 info_platform	21
7.3 build/source/CMakeFiles/Sample-project.dir/main.cpp.obj.d File Reference	21
7.4 build/source/CMakeFiles/Sample-project.dir/Person.cpp.obj.d File Reference	21
7.5 include/Person.h File Reference	21
7.6 Person.h	22
7.7 README.md File Reference	22
7.8 source/main.cpp File Reference	22
7.8.1 Function Documentation	23
7.8.1.1 main()	23
7.9 source/Person.cpp File Reference	23
<b>Index</b>	<b>25</b>

# 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:

<a href="#">Rex</a> . . . . .	9
-------------------------------	---





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Rex::Person</a> . . . . .	11
---------------------------------------	----



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

build/CMakeFiles/3.22.3/CompilerIdC/ <a href="#">CMakeCCompilerId.c</a> . . . . .	15
build/CMakeFiles/3.22.3/CompilerIdCXX/ <a href="#">CMakeCXXCompilerId.cpp</a> . . . . .	18
build/source/CMakeFiles/Sample-project.dir/ <a href="#">main.cpp.obj.d</a> . . . . .	21
build/source/CMakeFiles/Sample-project.dir/ <a href="#">Person.cpp.obj.d</a> . . . . .	21
include/ <a href="#">Person.h</a> . . . . .	21
source/ <a href="#">main.cpp</a> . . . . .	22
source/ <a href="#">Person.cpp</a> . . . . .	23



## Chapter 5

# Namespace Documentation

### 5.1 Rex Namespace Reference

#### Classes

- class [Person](#)



## Chapter 6

# Class Documentation

### 6.1 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.*

#### 6.1.1 Constructor & Destructor Documentation

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

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

Construct a new [Person](#) object.

##### 6.1.1.2 [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

**6.1.1.3 ~Person()**

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

Destroy the [Person](#) object.

**6.1.2 Member Function Documentation****6.1.2.1 SetAge()**

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

Set the Age object.

**Parameters**

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

**6.1.2.2 SetName()**

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

Set the Name\_ object.

**Parameters**

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

**6.1.2.3 ShowInfo()**

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



Show the name and age of the person.

The documentation for this class was generated from the following files:

- include/[Person.h](#)
- source/[Person.cpp](#)



## Chapter 7

# File Documentation

### 7.1 build/CMakeFiles/3.22.3/CompilerIdC/CMakeCCompilerId.c File Reference

#### Macros

- `#define __has_include(x) 0`
- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define C_VERSION`

#### Functions

- `int main (int argc, char *argv[])`

#### Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_standard_default`
- `const char * info_language_extensions_default`

#### 7.1.1 Macro Definition Documentation

#### 7.1.1.1 `__has_include`

```
#define __has_include(
    x ) 0
```

#### 7.1.1.2 `ARCHITECTURE_ID`

```
#define ARCHITECTURE_ID
```

#### 7.1.1.3 `C_VERSION`

```
#define C_VERSION
```

#### 7.1.1.4 `COMPILER_ID`

```
#define COMPILER_ID ""
```

#### 7.1.1.5 `DEC`

```
#define DEC(
    n )
```

##### Value:

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + (n) % 10)
```

#### 7.1.1.6 `HEX`

```
#define HEX(
    n )
```

##### Value:

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```

### 7.1.1.7 PLATFORM\_ID

```
#define PLATFORM_ID
```

### 7.1.1.8 STRINGIFY

```
#define STRINGIFY(  
    X ) STRINGIFY_HELPER(X)
```

### 7.1.1.9 STRINGIFY\_HELPER

```
#define STRINGIFY_HELPER(  
    X ) #X
```

## 7.1.2 Function Documentation

### 7.1.2.1 main()

```
int main (  
    int argc,  
    char * argv[] )
```

## 7.1.3 Variable Documentation

### 7.1.3.1 info\_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

### 7.1.3.2 info\_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

### 7.1.3.3 info\_language\_extensions\_default

```
const char* info_language_extensions_default
```

#### Initial value:

```
= "INFO" ":" "extensions_default["  
  "OFF"  
"]"
```

### 7.1.3.4 info\_language\_standard\_default

```
const char* info_language_standard_default
```

#### Initial value:

```
=  
  "INFO" ":" "standard_default[" C_VERSION "]"
```

### 7.1.3.5 info\_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

## 7.2 build/CMakeFiles/3.22.3/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

### Macros

- `#define __has_include(x) 0`
- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define CXX_STD __cplusplus`

### Functions

- `int main (int argc, char *argv[ ])`

### Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_standard_default`
- `const char * info_language_extensions_default`

## 7.2.1 Macro Definition Documentation

### 7.2.1.1 `__has_include`

```
#define __has_include(  
    x ) 0
```

### 7.2.1.2 `ARCHITECTURE_ID`

```
#define ARCHITECTURE_ID
```

### 7.2.1.3 `COMPILER_ID`

```
#define COMPILER_ID ""
```

### 7.2.1.4 `CXX_STD`

```
#define CXX_STD __cplusplus
```

### 7.2.1.5 `DEC`

```
#define DEC(  
    n )
```

#### Value:

```
('0' + ((n) / 10000000) % 10), \  
( '0' + ((n) / 1000000) % 10), \  
( '0' + ((n) / 100000) % 10), \  
( '0' + ((n) / 10000) % 10), \  
( '0' + ((n) / 1000) % 10), \  
( '0' + ((n) / 100) % 10), \  
( '0' + ((n) / 10) % 10), \  
( '0' + ((n) % 10))
```

### 7.2.1.6 HEX

```
#define HEX(  
    n )
```

#### Value:

```
('0' + ((n)>>28 & 0xF)), \
('0' + ((n)>>24 & 0xF)), \
('0' + ((n)>>20 & 0xF)), \
('0' + ((n)>>16 & 0xF)), \
('0' + ((n)>>12 & 0xF)), \
('0' + ((n)>>8  & 0xF)), \
('0' + ((n)>>4  & 0xF)), \
('0' + ((n)    & 0xF))
```

### 7.2.1.7 PLATFORM\_ID

```
#define PLATFORM_ID
```

### 7.2.1.8 STRINGIFY

```
#define STRINGIFY(  
    X ) STRINGIFY\_HELPER(X)
```

### 7.2.1.9 STRINGIFY\_HELPER

```
#define STRINGIFY_HELPER(  
    X ) #X
```

## 7.2.2 Function Documentation

### 7.2.2.1 main()

```
int main (  
    int argc,  
    char * argv[] )
```

## 7.2.3 Variable Documentation



### 7.2.3.1 info\_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

### 7.2.3.2 info\_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

### 7.2.3.3 info\_language\_extensions\_default

```
const char* info_language_extensions_default
```

**Initial value:**

```
= "INFO" ":" "extensions_default["  
  "OFF"  
"]"
```

### 7.2.3.4 info\_language\_standard\_default

```
const char* info_language_standard_default
```

**Initial value:**

```
= "INFO" ":" "standard_default["  
  "98"  
"]"
```

### 7.2.3.5 info\_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

## 7.3 build/source/CMakeFiles/Sample-project.dir/main.cpp.obj.d File Reference

## 7.4 build/source/CMakeFiles/Sample-project.dir/Person.cpp.obj.d File Reference

## 7.5 include/Person.h File Reference

```
#include <string>
```

## Classes

- class [Rex::Person](#)

## Namespaces

- namespace [Rex](#)

## 7.6 Person.h

[Go to the documentation of this file.](#)

```
1 #ifndef Person_h_
2 #define Person_h_
3
4 #include<string>
5
6 namespace Rex
7 {
8
9 class Person
10 {
11     private:
12
13         std::string Name_ {"Unknown"};
14
15         unsigned int Age_ {0};
16
17     public:
18
19         Person();
20
21         Person( const std::string& Name, const unsigned int& Age );
22
23         ~Person();
24
25         void ShowInfo() const;
26
27         void SetName( const std::string& Name );
28
29         void SetAge( const unsigned int& Age );
30 }; // end of the class Person
31
32 } // end of the namespace Rex
33
34 #endif // end of the definition Person_h_
```

## 7.7 README.md File Reference

## 7.8 source/main.cpp File Reference

```
#include "Person.h"
```

## Functions

- int [main](#) (int argc, char \*argv[ ])

## 7.8.1 Function Documentation

### 7.8.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

## 7.9 source/Person.cpp File Reference

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

### Namespaces

- namespace [Rex](#)



# Index

- `__has_include`
    - `CMakeCCompilerId.c`, [15](#)
    - `CMakeCXXCompilerId.cpp`, [19](#)
  - `~Person`
    - `Rex::Person`, [12](#)
- `ARCHITECTURE_ID`
  - `CMakeCCompilerId.c`, [16](#)
  - `CMakeCXXCompilerId.cpp`, [19](#)
- `build/CMakeFiles/3.22.3/CompilerIdC/CMakeCCompilerId.c`, [15](#)
- `build/CMakeFiles/3.22.3/CompilerIdCXX/CMakeCXXCompilerId.cpp`, [18](#)
- `build/source/CMakeFiles/Sample-project.dir/main.cpp.obj.d`, [21](#)
- `build/source/CMakeFiles/Sample-project.dir/Person.cpp.obj.d`, [21](#)
- `C_VERSION`
  - `CMakeCCompilerId.c`, [16](#)
- `CMakeCCompilerId.c`
  - `__has_include`, [15](#)
  - `ARCHITECTURE_ID`, [16](#)
  - `C_VERSION`, [16](#)
  - `COMPILER_ID`, [16](#)
  - `DEC`, [16](#)
  - `HEX`, [16](#)
  - `info_arch`, [17](#)
  - `info_compiler`, [17](#)
  - `info_language_extensions_default`, [17](#)
  - `info_language_standard_default`, [18](#)
  - `info_platform`, [18](#)
  - `main`, [17](#)
  - `PLATFORM_ID`, [16](#)
  - `STRINGIFY`, [17](#)
  - `STRINGIFY_HELPER`, [17](#)
- `CMakeCXXCompilerId.cpp`
  - `__has_include`, [19](#)
  - `ARCHITECTURE_ID`, [19](#)
  - `COMPILER_ID`, [19](#)
  - `CXX_STD`, [19](#)
  - `DEC`, [19](#)
  - `HEX`, [19](#)
  - `info_arch`, [20](#)
  - `info_compiler`, [21](#)
  - `info_language_extensions_default`, [21](#)
  - `info_language_standard_default`, [21](#)
  - `info_platform`, [21](#)
  - `main`, [20](#)
- `PLATFORM_ID`, [20](#)
- `STRINGIFY`, [20](#)
- `STRINGIFY_HELPER`, [20](#)
- `COMPILER_ID`
  - `CMakeCCompilerId.c`, [16](#)
  - `CMakeCXXCompilerId.cpp`, [19](#)
- `CXX_STD`
  - `CMakeCXXCompilerId.cpp`, [19](#)
- `DEC`
  - `CMakeCCompilerId.c`, [16](#)
  - `CMakeCXXCompilerId.cpp`, [19](#)
- `HEX`
  - `CMakeCCompilerId.c`, [16](#)
  - `CMakeCXXCompilerId.cpp`, [19](#)
- `include/Person.h`, [21](#), [22](#)
- `info_arch`
  - `CMakeCCompilerId.c`, [17](#)
  - `CMakeCXXCompilerId.cpp`, [20](#)
- `info_compiler`
  - `CMakeCCompilerId.c`, [17](#)
  - `CMakeCXXCompilerId.cpp`, [21](#)
- `info_language_extensions_default`
  - `CMakeCCompilerId.c`, [17](#)
  - `CMakeCXXCompilerId.cpp`, [21](#)
- `info_language_standard_default`
  - `CMakeCCompilerId.c`, [18](#)
  - `CMakeCXXCompilerId.cpp`, [21](#)
- `info_platform`
  - `CMakeCCompilerId.c`, [18](#)
  - `CMakeCXXCompilerId.cpp`, [21](#)
- `main`
  - `CMakeCCompilerId.c`, [17](#)
  - `CMakeCXXCompilerId.cpp`, [20](#)
  - `main.cpp`, [23](#)
- `main.cpp`
  - `main`, [23](#)
- `Person`
  - `Rex::Person`, [11](#)
- `PLATFORM_ID`
  - `CMakeCCompilerId.c`, [16](#)
  - `CMakeCXXCompilerId.cpp`, [20](#)
- `README.md`, [22](#)
- `Rex`, [9](#)
- `Rex::Person`, [11](#)
- `~Person`, [12](#)

- Person, [11](#)
- SetAge, [12](#)
- SetName, [12](#)
- ShowInfo, [12](#)
- SetAge
  - Rex::Person, [12](#)
- SetName
  - Rex::Person, [12](#)
- ShowInfo
  - Rex::Person, [12](#)
- source/main.cpp, [22](#)
- source/Person.cpp, [23](#)
- STRINGIFY
  - CMakeCCompilerId.c, [17](#)
  - CMakeCXXCompilerId.cpp, [20](#)
- STRINGIFY\_HELPER
  - CMakeCCompilerId.c, [17](#)
  - CMakeCXXCompilerId.cpp, [20](#)