

hello world

0.0.1

다음에 의해 생성됨 : Doxygen 1.9.1

1	클래스 색인	1
1.1	클래스 목록	1
2	파일 색인	3
2.1	파일 목록	3
3	클래스 문서화	5
3.1	A 클래스 참조	5
3.1.1	상세한 설명	5
3.1.2	생성자 & 소멸자 문서화	6
3.1.2.1	A() [1/2]	6
3.1.2.2	A() [2/2]	6
3.1.3	멤버 함수 문서화	6
3.1.3.1	print()	7
3.1.4	멤버 데이터 문서화	7
3.1.4.1	a	7
3.1.4.2	b	7
4	파일 문서화	9
4.1	chap1/1-1/build/CMakeFiles/3.22.1/CompilerIdC/CMakeCCompilerId.c 파일 참조	9
4.1.1	매크로 문서화	9
4.1.1.1	__has_include	10
4.1.1.2	ARCHITECTURE_ID	10
4.1.1.3	C_VERSION	10
4.1.1.4	COMPILER_ID	10
4.1.1.5	DEC	10
4.1.1.6	HEX	11
4.1.1.7	PLATFORM_ID	11
4.1.1.8	STRINGIFY	11
4.1.1.9	STRINGIFY_HELPER	11
4.1.2	함수 문서화	11
4.1.2.1	main()	12
4.1.3	변수 문서화	12
4.1.3.1	info_arch	12
4.1.3.2	info_compiler	12
4.1.3.3	info_language_extensions_default	12
4.1.3.4	info_language_standard_default	13
4.1.3.5	info_platform	13
4.2	chap1/1-2/build/CMakeFiles/3.22.1/CompilerIdC/CMakeCCompilerId.c 파일 참조	13
4.2.1	매크로 문서화	13
4.2.1.1	__has_include	14
4.2.1.2	ARCHITECTURE_ID	14
4.2.1.3	C_VERSION	14

4.2.1.4 COMPILER_ID	14
4.2.1.5 DEC	14
4.2.1.6 HEX	15
4.2.1.7 PLATFORM_ID	15
4.2.1.8 STRINGIFY	15
4.2.1.9 STRINGIFY_HELPER	15
4.2.2 함수 문서화	15
4.2.2.1 main()	16
4.2.3 변수 문서화	16
4.2.3.1 info_arch	16
4.2.3.2 info_compiler	16
4.2.3.3 info_language_extensions_default	16
4.2.3.4 info_language_standard_default	17
4.2.3.5 info_platform	17
4.3 chap1/1-1/build/CMakeFiles/3.22.1/CompilerIdCXX/CMakeCXXCompilerId.cpp 파일 참조	17
4.3.1 매크로 문서화	17
4.3.1.1 __has_include	18
4.3.1.2 ARCHITECTURE_ID	18
4.3.1.3 COMPILER_ID	18
4.3.1.4 CXX_STD	18
4.3.1.5 DEC	18
4.3.1.6 HEX	19
4.3.1.7 PLATFORM_ID	19
4.3.1.8 STRINGIFY	19
4.3.1.9 STRINGIFY_HELPER	19
4.3.2 함수 문서화	19
4.3.2.1 main()	20
4.3.3 변수 문서화	20
4.3.3.1 info_arch	20
4.3.3.2 info_compiler	20
4.3.3.3 info_language_extensions_default	20
4.3.3.4 info_language_standard_default	21
4.3.3.5 info_platform	21
4.4 chap1/1-2/build/CMakeFiles/3.22.1/CompilerIdCXX/CMakeCXXCompilerId.cpp 파일 참조	21
4.4.1 매크로 문서화	21
4.4.1.1 __has_include	22
4.4.1.2 ARCHITECTURE_ID	22
4.4.1.3 COMPILER_ID	22
4.4.1.4 CXX_STD	22
4.4.1.5 DEC	22
4.4.1.6 HEX	23
4.4.1.7 PLATFORM_ID	23

4.4.1.8 STRINGIFY	23
4.4.1.9 STRINGIFY_HELPER	23
4.4.2 함수 문서화	23
4.4.2.1 main()	24
4.4.3 변수 문서화	24
4.4.3.1 info_arch	24
4.4.3.2 info_compiler	24
4.4.3.3 info_language_extensions_default	24
4.4.3.4 info_language_standard_default	25
4.4.3.5 info_platform	25
4.5 chap1/1-1/build/CMakeFiles/main.dir/main.cpp.o.d 파일 참조	25
4.6 chap1/1-1/main.cpp 파일 참조	25
4.6.1 상세한 설명	26
4.6.2 함수 문서화	26
4.6.2.1 main()	26
4.6.2.2 print_hello_world()	27
4.7 chap2/2-1/main.cpp 파일 참조	27
4.7.1 함수 문서화	28
4.7.1.1 main()	28
4.8 chap1/1-1/main.hpp 파일 참조	28
4.8.1 상세한 설명	29
4.9 chap1/1-2/booklist.cpp 파일 참조	29
4.9.1 함수 문서화	29
4.9.1.1 main()	29
4.10 chap1/1-2/build/CMakeFiles/booklist.dir/booklist.cpp.o.d 파일 참조	30
Index	31

Chapter 1

클래스 색인

1.1 클래스 목록

다음은 클래스, 구조체, 공용체 그리고 인터페이스들입니다. (간략한 설명만을 보여줍니다) :

A	
A 클래스입니다	5

Chapter 2

파일 색인

2.1 파일 목록

다음은 모든 파일에 대한 목록입니다. (간략한 설명만을 보여줍니다) :

chap1/1-1/ main.cpp	
"Hello, World!"와 입력 매개변수의 값을 출력합니다	25
chap1/1-1/ main.hpp	
"Hello, World!"와 입력 매개변수의 값을 출력합니다	28
chap1/1-1/build/CMakeFiles/3.22.1/CompilerIdC/ CMakeCCompilerId.c	9
chap1/1-1/build/CMakeFiles/3.22.1/CompilerIdCXX/ CMakeCXXCompilerId.cpp	17
chap1/1-1/build/CMakeFiles/main.dir/ main.cpp.o.d	25
chap1/1-2/ booklist.cpp	29
chap1/1-2/build/CMakeFiles/3.22.1/CompilerIdC/ CMakeCCompilerId.c	13
chap1/1-2/build/CMakeFiles/3.22.1/CompilerIdCXX/ CMakeCXXCompilerId.cpp	21
chap1/1-2/build/CMakeFiles/booklist.dir/ booklist.cpp.o.d	30
chap2/2-1/ main.cpp	27

Chapter 3

클래스 문서화

3.1 A 클래스 참조

A 클래스입니다.

A에 대한 협력 다이어그램:

A
+ a + b
+ A() + A() + print()

Public 멤버 함수

- A ()
- A (int a, int b)
- void print ()

Public 속성

- int a
- int b

3.1.1 상세한 설명

A 클래스입니다.

매개변수

<i>a</i>	A 클래스의 멤버 변수입니다.
<i>b</i>	A 클래스의 멤버 변수입니다.
<i>A</i>	A 클래스의 생성자입니다.
<i>print</i>	A 클래스의 멤버 함수입니다.

main.cpp 파일의 19 번째 라인에서 정의되었습니다.

3.1.2 생성자 & 소멸자 문서화

3.1.2.1 A() [1/2]

```
A::A ( ) [inline]
```

main.cpp 파일의 23 번째 라인에서 정의되었습니다.

```
23     {
24         a = 0;
25         b = 0;
26     }
```

3.1.2.2 A() [2/2]

```
A::A (
    int a,
    int b ) [inline]
```

main.cpp 파일의 27 번째 라인에서 정의되었습니다.

```
27     {
28         this->a = a;
29         this->b = b;
30     }
```

3.1.3 멤버 함수 문서화

3.1.3.1 print()

```
void A::print ( ) [inline]
```

main.cpp 파일의 31 번째 라인에서 정의되었습니다.

```
31 {  
32     std::cout << a << " " << b << std::endl;  
33 }
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



3.1.4 멤버 데이터 문서화

3.1.4.1 a

```
int A::a
```

main.cpp 파일의 21 번째 라인에서 정의되었습니다.

3.1.4.2 b

```
int A::b
```

main.cpp 파일의 22 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- [chap1/1-1/main.cpp](#)

Chapter 4

파일 문서화

4.1 chap1/1-1/build/CMakeFiles/3.22.1/CompilerIdC/CMakeCCompilerId.c 파일 참조

매크로

- #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`

함수

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

변수

- 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`

4.1.1 매크로 문서화

4.1.1.1 `__has_include`

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

CMakeCCompilerId.c 파일의 17 번째 라인에서 정의되었습니다.

4.1.1.2 `ARCHITECTURE_ID`

```
#define ARCHITECTURE_ID
```

CMakeCCompilerId.c 파일의 652 번째 라인에서 정의되었습니다.

4.1.1.3 `C_VERSION`

```
#define C_VERSION
```

CMakeCCompilerId.c 파일의 741 번째 라인에서 정의되었습니다.

4.1.1.4 `COMPILER_ID`

```
#define COMPILER_ID ""
```

CMakeCCompilerId.c 파일의 396 번째 라인에서 정의되었습니다.

4.1.1.5 `DEC`

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

값:

```
('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))
```

CMakeCCompilerId.c 파일의 656 번째 라인에서 정의되었습니다.

4.1.1.6 HEX

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

값:

```
('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))
```

CMakeCCompilerId.c 파일의 667 번째 라인에서 정의되었습니다.

4.1.1.7 PLATFORM_ID

```
#define PLATFORM_ID
```

CMakeCCompilerId.c 파일의 524 번째 라인에서 정의되었습니다.

4.1.1.8 STRINGIFY

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

CMakeCCompilerId.c 파일의 417 번째 라인에서 정의되었습니다.

4.1.1.9 STRINGIFY_HELPER

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

CMakeCCompilerId.c 파일의 416 번째 라인에서 정의되었습니다.

4.1.2 함수 문서화

4.1.2.1 main()

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

CMakeCCompilerId.c 파일의 776 번째 라인에서 정의되었습니다.

```
778 {
779     int require = 0;
780     require += info_compiler[argc];
781     require += info_platform[argc];
782     require += info_arch[argc];
783 #ifdef COMPILER_VERSION_MAJOR
784     require += info_version[argc];
785 #endif
786 #ifdef COMPILER_VERSION_INTERNAL
787     require += info_version_internal[argc];
788 #endif
789 #ifdef SIMULATE_ID
790     require += info_simulate[argc];
791 #endif
792 #ifdef SIMULATE_VERSION_MAJOR
793     require += info_simulate_version[argc];
794 #endif
795 #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
796     require += info_cray[argc];
797 #endif
798     require += info_language_standard_default[argc];
799     require += info_language_extensions_default[argc];
800     (void)argv;
801     return require;
802 }
```

4.1.3 변수 문서화

4.1.3.1 info_arch

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

CMakeCCompilerId.c 파일의 733 번째 라인에서 정의되었습니다.

4.1.3.2 info_compiler

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

CMakeCCompilerId.c 파일의 403 번째 라인에서 정의되었습니다.

4.1.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

초기값:

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

CMakeCCompilerId.c 파일의 757 번째 라인에서 정의되었습니다.

4.1.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

초기값:

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

CMakeCCompilerId.c 파일의 754 번째 라인에서 정의되었습니다.

4.1.3.5 info_platform

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

CMakeCCompilerId.c 파일의 732 번째 라인에서 정의되었습니다.

4.2 chap1/1-2/build/CMakeFiles/3.22.1/CompilerIdC/CMakeCCompilerId.c 파일 참조

매크로

- #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`

함수

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

변수

- 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`

4.2.1 매크로 문서화

4.2.1.1 `__has_include`

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

CMakeCCompilerId.c 파일의 17 번째 라인에서 정의되었습니다.

4.2.1.2 `ARCHITECTURE_ID`

```
#define ARCHITECTURE_ID
```

CMakeCCompilerId.c 파일의 652 번째 라인에서 정의되었습니다.

4.2.1.3 `C_VERSION`

```
#define C_VERSION
```

CMakeCCompilerId.c 파일의 741 번째 라인에서 정의되었습니다.

4.2.1.4 `COMPILER_ID`

```
#define COMPILER_ID ""
```

CMakeCCompilerId.c 파일의 396 번째 라인에서 정의되었습니다.

4.2.1.5 `DEC`

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

값:

```
('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))
```

CMakeCCompilerId.c 파일의 656 번째 라인에서 정의되었습니다.

4.2.1.6 HEX

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

값:

```
('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))
```

CMakeCCompilerId.c 파일의 667 번째 라인에서 정의되었습니다.

4.2.1.7 PLATFORM_ID

```
#define PLATFORM_ID
```

CMakeCCompilerId.c 파일의 524 번째 라인에서 정의되었습니다.

4.2.1.8 STRINGIFY

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

CMakeCCompilerId.c 파일의 417 번째 라인에서 정의되었습니다.

4.2.1.9 STRINGIFY_HELPER

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

CMakeCCompilerId.c 파일의 416 번째 라인에서 정의되었습니다.

4.2.2 함수 문서화

4.2.2.1 main()

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

CMakeCCompilerId.c 파일의 776 번째 라인에서 정의되었습니다.

```
778 {
779     int require = 0;
780     require += info_compiler[argc];
781     require += info_platform[argc];
782     require += info_arch[argc];
783 #ifdef COMPILER_VERSION_MAJOR
784     require += info_version[argc];
785 #endif
786 #ifdef COMPILER_VERSION_INTERNAL
787     require += info_version_internal[argc];
788 #endif
789 #ifdef SIMULATE_ID
790     require += info_simulate[argc];
791 #endif
792 #ifdef SIMULATE_VERSION_MAJOR
793     require += info_simulate_version[argc];
794 #endif
795 #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
796     require += info_cray[argc];
797 #endif
798     require += info_language_standard_default[argc];
799     require += info_language_extensions_default[argc];
800     (void)argv;
801     return require;
802 }
```

4.2.3 변수 문서화

4.2.3.1 info_arch

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

CMakeCCompilerId.c 파일의 733 번째 라인에서 정의되었습니다.

4.2.3.2 info_compiler

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

CMakeCCompilerId.c 파일의 403 번째 라인에서 정의되었습니다.

4.2.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

초기값:

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

CMakeCCompilerId.c 파일의 757 번째 라인에서 정의되었습니다.

4.2.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

초기값:

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

CMakeCCompilerId.c 파일의 754 번째 라인에서 정의되었습니다.

4.2.3.5 info_platform

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

CMakeCCompilerId.c 파일의 732 번째 라인에서 정의되었습니다.

4.3 chap1/1-1/build/CMakeFiles/3.22.1/CompilerIdCXX/CMakeCXXCompilerId.cpp 파일 참조

매크로

- #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`

함수

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

변수

- 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`

4.3.1 매크로 문서화

4.3.1.1 `__has_include`

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

CMakeCXXCompilerId.cpp 파일의 11 번째 라인에서 정의되었습니다.

4.3.1.2 `ARCHITECTURE_ID`

```
#define ARCHITECTURE_ID
```

CMakeCXXCompilerId.cpp 파일의 637 번째 라인에서 정의되었습니다.

4.3.1.3 `COMPILER_ID`

```
#define COMPILER_ID ""
```

CMakeCXXCompilerId.cpp 파일의 381 번째 라인에서 정의되었습니다.

4.3.1.4 `CXX_STD`

```
#define CXX_STD __cplusplus
```

CMakeCXXCompilerId.cpp 파일의 735 번째 라인에서 정의되었습니다.

4.3.1.5 `DEC`

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

값:

```
('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))
```

CMakeCXXCompilerId.cpp 파일의 641 번째 라인에서 정의되었습니다.

4.3.1.6 HEX

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

값:

```
('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))
```

CMakeCXXCompilerId.cpp 파일의 652 번째 라인에서 정의되었습니다.

4.3.1.7 PLATFORM_ID

```
#define PLATFORM_ID
```

CMakeCXXCompilerId.cpp 파일의 509 번째 라인에서 정의되었습니다.

4.3.1.8 STRINGIFY

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

CMakeCXXCompilerId.cpp 파일의 402 번째 라인에서 정의되었습니다.

4.3.1.9 STRINGIFY_HELPER

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

CMakeCXXCompilerId.cpp 파일의 401 번째 라인에서 정의되었습니다.

4.3.2 함수 문서화

4.3.2.1 main()

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

CMakeCXXCompilerId.cpp 파일의 767 번째 라인에서 정의되었습니다.

```
768 {
769     int require = 0;
770     require += info_compiler[argc];
771     require += info_platform[argc];
772     #ifdef COMPILER_VERSION_MAJOR
773     require += info_version[argc];
774     #endif
775     #ifdef COMPILER_VERSION_INTERNAL
776     require += info_version_internal[argc];
777     #endif
778     #ifdef SIMULATE_ID
779     require += info_simulate[argc];
780     #endif
781     #ifdef SIMULATE_VERSION_MAJOR
782     require += info_simulate_version[argc];
783     #endif
784     #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
785     require += info_cray[argc];
786     #endif
787     require += info_language_standard_default[argc];
788     require += info_language_extensions_default[argc];
789     (void)argv;
790     return require;
791 }
```

4.3.3 변수 문서화

4.3.3.1 info_arch

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

CMakeCXXCompilerId.cpp 파일의 718 번째 라인에서 정의되었습니다.

4.3.3.2 info_compiler

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

CMakeCXXCompilerId.cpp 파일의 388 번째 라인에서 정의되었습니다.

4.3.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

초기값:

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

CMakeCXXCompilerId.cpp 파일의 754 번째 라인에서 정의되었습니다.

4.3.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

초기값:

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

CMakeCXXCompilerId.cpp 파일의 738 번째 라인에서 정의되었습니다.

4.3.3.5 info_platform

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

CMakeCXXCompilerId.cpp 파일의 717 번째 라인에서 정의되었습니다.

4.4 chap1/1-2/build/CMakeFiles/3.22.1/CompilerIdCXX/CMakeCXXCompilerId.cpp 파일 참조

매크로

- #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

함수

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

변수

- 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`

4.4.1 매크로 문서화

4.4.1.1 `__has_include`

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

CMakeCXXCompilerId.cpp 파일의 11 번째 라인에서 정의되었습니다.

4.4.1.2 `ARCHITECTURE_ID`

```
#define ARCHITECTURE_ID
```

CMakeCXXCompilerId.cpp 파일의 637 번째 라인에서 정의되었습니다.

4.4.1.3 `COMPILER_ID`

```
#define COMPILER_ID ""
```

CMakeCXXCompilerId.cpp 파일의 381 번째 라인에서 정의되었습니다.

4.4.1.4 `CXX_STD`

```
#define CXX_STD __cplusplus
```

CMakeCXXCompilerId.cpp 파일의 735 번째 라인에서 정의되었습니다.

4.4.1.5 `DEC`

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

값:

```
('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))
```

CMakeCXXCompilerId.cpp 파일의 641 번째 라인에서 정의되었습니다.

4.4.1.6 HEX

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

값:

```
('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))
```

CMakeCXXCompilerId.cpp 파일의 652 번째 라인에서 정의되었습니다.

4.4.1.7 PLATFORM_ID

```
#define PLATFORM_ID
```

CMakeCXXCompilerId.cpp 파일의 509 번째 라인에서 정의되었습니다.

4.4.1.8 STRINGIFY

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

CMakeCXXCompilerId.cpp 파일의 402 번째 라인에서 정의되었습니다.

4.4.1.9 STRINGIFY_HELPER

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

CMakeCXXCompilerId.cpp 파일의 401 번째 라인에서 정의되었습니다.

4.4.2 함수 문서화

4.4.2.1 main()

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

CMakeCXXCompilerId.cpp 파일의 767 번째 라인에서 정의되었습니다.

```
768 {
769     int require = 0;
770     require += info_compiler[argc];
771     require += info_platform[argc];
772     #ifdef COMPILER_VERSION_MAJOR
773     require += info_version[argc];
774     #endif
775     #ifdef COMPILER_VERSION_INTERNAL
776     require += info_version_internal[argc];
777     #endif
778     #ifdef SIMULATE_ID
779     require += info_simulate[argc];
780     #endif
781     #ifdef SIMULATE_VERSION_MAJOR
782     require += info_simulate_version[argc];
783     #endif
784     #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
785     require += info_cray[argc];
786     #endif
787     require += info_language_standard_default[argc];
788     require += info_language_extensions_default[argc];
789     (void)argv;
790     return require;
791 }
```

4.4.3 변수 문서화

4.4.3.1 info_arch

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

CMakeCXXCompilerId.cpp 파일의 718 번째 라인에서 정의되었습니다.

4.4.3.2 info_compiler

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

CMakeCXXCompilerId.cpp 파일의 388 번째 라인에서 정의되었습니다.

4.4.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

초기값:

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

CMakeCXXCompilerId.cpp 파일의 754 번째 라인에서 정의되었습니다.

4.4.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

초기값:

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

CMakeCXXCompilerId.cpp 파일의 738 번째 라인에서 정의되었습니다.

4.4.3.5 info_platform

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

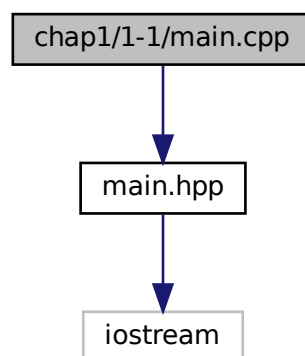
CMakeCXXCompilerId.cpp 파일의 717 번째 라인에서 정의되었습니다.

4.5 chap1/1-1/build/CMakeFiles/main.dir/main.cpp.o.d 파일 참조

4.6 chap1/1-1/main.cpp 파일 참조

"Hello, World!"와 입력 매개변수의 값을 출력합니다.

```
#include "main.hpp"  
main.cpp에 대한 include 의존 그래프
```



클래스

- class [A](#)
[A](#) 클래스입니다.

함수

- void `print_hello_world` (int a)
"Hello, World!"와 입력 매개변수의 값을 출력합니다.
- int `main` ()
`main` 함수입니다.

4.6.1 상세한 설명

"Hello, World!"와 입력 매개변수의 값을 출력합니다.

작성자

최수길

날짜

2024-02-27

4.6.2 함수 문서화

4.6.2.1 `main()`

```
int main ( )
```

`main` 함수입니다.

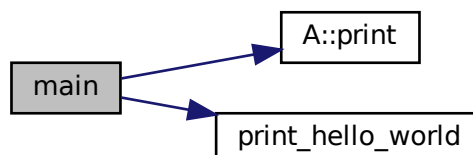
반환값

정상 종료가 되면 0을 반환합니다.

`main.cpp` 파일의 48 번째 라인에서 정의되었습니다.

```
48 {
49     int a = 0;
50     A b(1, 2);
51     b.print();
52     for (int i = 0; i < 10; i++) {
53         print_hello_world(a);
54         a += 1;
55     }
56     return 0;
57 }
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



4.6.2.2 print_hello_world()

```
void print_hello_world (
    int a )
```

"Hello, World!"와 입력 매개변수의 값을 출력합니다.

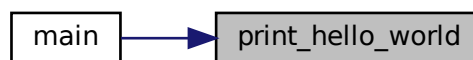
매개변수

<i>a</i>	메시지와 함께 출력될 정수 값입니다.
----------	----------------------

main.cpp 파일의 41 번째 라인에서 정의되었습니다.

```
41 {
42     std::cout << "Hello, World!" << a << std::endl;
43 }
```

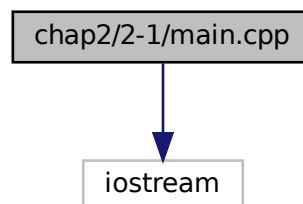
이 함수를 호출하는 함수들에 대한 그래프입니다.:



4.7 chap2/2-1/main.cpp 파일 참조

```
#include <iostream>
```

main.cpp에 대한 include 의존 그래프



함수

- int `main` ()

4.7.1 함수 문서화

4.7.1.1 main()

```
int main ( )
```

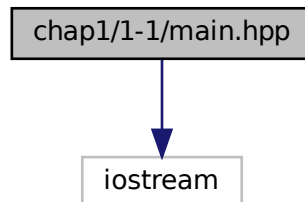
main.cpp 파일의 3 번째 라인에서 정의되었습니다.

```
3 {
4     std::cout << "Hello, World!" << std::endl;
5     std::cout << "첫 번째 맛보기 C++ 프로그램입니다." << std::endl;
6     return 0;
7 }
```

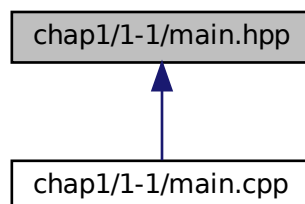
4.8 chap1/1-1/main.hpp 파일 참조

"Hello, World!"와 입력 매개변수의 값을 출력합니다.

```
#include <iostream>
main.hpp에 대한 include 의존 그래프
```



이 그래프는 이 파일을 직/간접적으로 include 하는 파일들을 보여줍니다.:



4.8.1 상세한 설명

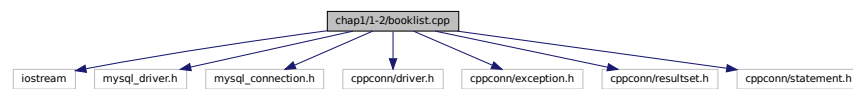
"Hello, World!"와 입력 매개변수의 값을 출력합니다.

파일입니다.

4.9 chap1/1-2/booklist.cpp 파일 참조

```
#include <iostream>
#include <mysql_driver.h>
#include <mysql_connection.h>
#include <cppconn/driver.h>
#include <cppconn/exception.h>
#include <cppconn/resultset.h>
#include <cppconn/statement.h>
```

booklist.cpp에 대한 include 의존 그래프



함수

- int `main` ()

4.9.1 함수 문서화

4.9.1.1 main()

```
int main ( )
```

booklist.cpp 파일의 12 번째 라인에서 정의되었습니다.

```
12 {
13     sql::mysql::MySQL_Driver *driver;
14     sql::Connection *con;
15
16     string host = "localhost:3306";
17     string user = "root";
18     string password = "0000";
19
20     try {
21         driver = sql::mysql::get_mysql_driver_instance();
22         con = driver->connect(host, user, password);
23
24         con->setSchema("madangdb");
25
26         sql::Statement *stmt = con->createStatement();
27         sql::ResultSet *res = stmt->executeQuery("SELECT * FROM Book");
28
29         while (res->next()) {
30             cout << res->getInt("bookid") << "\t";
31             cout << res->getString("bookname") << "\t";
```

```
32         cout << res->getString("publisher") << "\t";
33         cout << res->getInt("price") << endl;
34     }
35
36     delete res;
37     delete stmt;
38     delete con;
39 } catch (sql::SQLException &e) {
40     cout << "# ERR: SQLException in " << __FILE__;
41     cout << "(" << __FUNCTION__ << ") on line " << __LINE__ << endl;
42     cout << "# ERR: " << e.what();
43     cout << " (MySQL error code: " << e.getErrorCode();
44     cout << ", SQLState: " << e.getSQLState() << ")" << endl;
45 }
46
47 return 0;
48 }
```

4.10 chap1/1-2/build/CMakeFiles/booklist.dir/booklist.cpp.o.d 파일 참조

Index

- __has_include
 - CMakeCCompilerId.c, [9](#), [13](#)
 - CMakeCXXCompilerId.cpp, [17](#), [21](#)
- A, [5](#)
 - A, [6](#)
 - a, [7](#)
 - b, [7](#)
 - print, [6](#)
- a
 - A, [7](#)
- ARCHITECTURE_ID
 - CMakeCCompilerId.c, [10](#), [14](#)
 - CMakeCXXCompilerId.cpp, [18](#), [22](#)
- b
 - A, [7](#)
- booklist.cpp
 - main, [29](#)
- C_VERSION
 - CMakeCCompilerId.c, [10](#), [14](#)
- chap1/1-1/build/CMakeFiles/3.22.1/CompilerIdC/CMakeCCompilerId.c,
 - [9](#)
- chap1/1-1/build/CMakeFiles/3.22.1/CompilerIdCXX/CMakeCXXCompilerId.cpp,
 - [17](#)
- chap1/1-1/build/CMakeFiles/main.dir/main.cpp.o.d, [25](#)
- chap1/1-1/main.cpp, [25](#)
- chap1/1-1/main.hpp, [28](#)
- chap1/1-2/booklist.cpp, [29](#)
- chap1/1-2/build/CMakeFiles/3.22.1/CompilerIdC/CMakeCCompilerId.c,
 - [13](#)
- chap1/1-2/build/CMakeFiles/3.22.1/CompilerIdCXX/CMakeCXXCompilerId.cpp,
 - [21](#)
- chap1/1-2/build/CMakeFiles/booklist.dir/booklist.cpp.o.d,
 - [30](#)
- chap2/2-1/main.cpp, [27](#)
- CMakeCCompilerId.c
 - __has_include, [9](#), [13](#)
 - ARCHITECTURE_ID, [10](#), [14](#)
 - C_VERSION, [10](#), [14](#)
 - COMPILER_ID, [10](#), [14](#)
 - DEC, [10](#), [14](#)
 - HEX, [10](#), [14](#)
 - info_arch, [12](#), [16](#)
 - info_compiler, [12](#), [16](#)
 - info_language_extensions_default, [12](#), [16](#)
 - info_language_standard_default, [12](#), [16](#)
 - info_platform, [13](#), [17](#)
 - main, [11](#), [15](#)
 - PLATFORM_ID, [11](#), [15](#)
 - STRINGIFY, [11](#), [15](#)
 - STRINGIFY_HELPER, [11](#), [15](#)
- CMakeCXXCompilerId.cpp
 - __has_include, [17](#), [21](#)
 - ARCHITECTURE_ID, [18](#), [22](#)
 - COMPILER_ID, [18](#), [22](#)
 - CXX_STD, [18](#), [22](#)
 - DEC, [18](#), [22](#)
 - HEX, [18](#), [22](#)
 - info_arch, [20](#), [24](#)
 - info_compiler, [20](#), [24](#)
 - info_language_extensions_default, [20](#), [24](#)
 - info_language_standard_default, [20](#), [24](#)
 - info_platform, [21](#), [25](#)
 - main, [19](#), [23](#)
 - PLATFORM_ID, [19](#), [23](#)
 - STRINGIFY, [19](#), [23](#)
 - STRINGIFY_HELPER, [19](#), [23](#)
- COMPILER_ID
 - CMakeCCompilerId.c, [10](#), [14](#)
 - CMakeCXXCompilerId.cpp, [18](#), [22](#)
- CXX_STD
 - CMakeCXXCompilerId.cpp, [18](#), [22](#)
- DEC
 - CMakeCCompilerId.c, [10](#), [14](#)
 - CMakeCXXCompilerId.cpp, [18](#), [22](#)
- HEX
 - CMakeCCompilerId.c, [10](#), [14](#)
 - CMakeCXXCompilerId.cpp, [18](#), [22](#)
- info_arch
 - CMakeCCompilerId.c, [12](#), [16](#)
 - CMakeCXXCompilerId.cpp, [20](#), [24](#)
- info_compiler
 - CMakeCCompilerId.c, [12](#), [16](#)
 - CMakeCXXCompilerId.cpp, [20](#), [24](#)
- info_language_extensions_default
 - CMakeCCompilerId.c, [12](#), [16](#)
 - CMakeCXXCompilerId.cpp, [20](#), [24](#)
- info_language_standard_default
 - CMakeCCompilerId.c, [12](#), [16](#)
 - CMakeCXXCompilerId.cpp, [20](#), [24](#)
- info_platform
 - CMakeCCompilerId.c, [13](#), [17](#)
 - CMakeCXXCompilerId.cpp, [21](#), [25](#)
- main

- booklist.cpp, [29](#)
- CMakeCCompilerId.c, [11](#), [15](#)
- CMakeCXXCompilerId.cpp, [19](#), [23](#)
- main.cpp, [26](#), [28](#)
- main.cpp
 - main, [26](#), [28](#)
 - print_hello_world, [26](#)
- PLATFORM_ID
 - CMakeCCompilerId.c, [11](#), [15](#)
 - CMakeCXXCompilerId.cpp, [19](#), [23](#)
- print
 - A, [6](#)
- print_hello_world
 - main.cpp, [26](#)
- STRINGIFY
 - CMakeCCompilerId.c, [11](#), [15](#)
 - CMakeCXXCompilerId.cpp, [19](#), [23](#)
- STRINGIFY_HELPER
 - CMakeCCompilerId.c, [11](#), [15](#)
 - CMakeCXXCompilerId.cpp, [19](#), [23](#)