VSCode [illegible] *Windows C++* [illegible]

[illegible]

[illegible]

[illegible] VSCode [illegible] Visual Studio [illegible] C++ [illegible] Clang [illegible] MSYS2 + mingw-w64 [illegible]

[illegible] *MSYS2*

[illegible] GitHub Release [illegible] MSYS2 [illegible] GitHub [illegible]

[illegible] next [illegible]

[illegible] Windows Terminal[illegible] JSON [illegible] JSON [illegible] MSYS2 [illegible]

```
"profiles": {
  "list":
  [
    // ...
    {
      "guid": "{17da3cac-b318-431e-8a3e-7fcdefe6d114}",
      "name": "Clang64 / MSYS2",
      "commandline": "C:/msys64/msys2_shell.cmd -defterm -here -no-start -clang64",
      "startingDirectory": "C:/msys64/home/%USERNAME%",
      "icon": "C:/msys64/Clang64.ico",
    },
    {
      "guid": "{71160544-14d8-4194-af25-d05feeac7233}",
      "name": "MSYS / MSYS2",
      "commandline": "C:/msys64/msys2_shell.cmd -defterm -here -no-start -msys",
      "startingDirectory": "C:/msys64/home/%USERNAME%",
      "icon": "C:/msys64/msys2.ico",
    },
    // ...
  ]
}
```

[illegible] Windows Terminal [illegible] Clang64 / MSYS2 [illegible]

```
sed -i "s#https\?://mirror.msys2.org/#https://mirrors.tuna.tsinghua.edu.cn/msys2/#g" /etc/pacman.d/mirrorlist*
```

[illegible] MSYS2 [illegible]

```
pacman -Syu
```

☒☒☒☒☒

☒☒ *Clang* ☒☒☒

Warning

clang ☒☒☒☒☒ clang64 ☒☒☒☒☒☒☒☒☒ clangd ☒☒☒☒☒☒☒

☒ Clang64 / MSYS2 ☒☒☒☒☒☒

```
pacman -S mingw-w64-clang-x86_64-toolchain
```

☒☒☒☒ clang ☒☒☒☒

☒☒ *Xmake*

- **Windows ☒☒☒**

  ☒ Release ☒☒☒☒☒☒☒ xmake-master.win64.exe☒☒☒☒☒☒

- **winget**

  ☒☒☒☒☒

  ```
  winget install xmake
  ```

  ☒☒☒☒ Xmake☒

- **msys2**

  ☒ Clang64 / MSYS2 ☒☒☒☒☒☒

  ```
  pacman -Sy mingw-w64-x86_64-xmake
  ```

  ☒☒☒☒ Xmake☒

☒☒ *VSCode* ☒☒

☒ VSCode ☒☒☒☒☒☒☒☒ Clangd☒CodeLLDB☒Xmake ☒☒☒☒

CodeLLDB ☒☒☒ VSIX ☒☒☒☒☒☒☒

☒☒

VSCode ☒☒

☒ VSCode ☒ setting.json ☒☒☒☒☒☒☒☒

```
// settings.json
// Xmake
"xmake.debugConfigType": "codelldb", // □□ codelldb □□□□ cpptools □□□□
"xmake.runMode": "buildRun", // □□□□□ build
"xmake.buildLevel": "verbose", // □□□□□□□□□□□,□□□warnings□□,□□□□□□□□□□□□□□□,verbose□□□□□□□□□□□□□□,debug□□□□ xmake -vD □□□□□□□□□□□□□□□
"xmake.customDebugConfig": {
    "console": "integratedTerminal" // XMake□□□□□□□□□□□ debug console,□□□□□externalTerminal
},
```

```
// Clangd
"clangd.path": "C:/msys64/clang64/bin/clangd.exe", // Clangd □□□□
"clangd.arguments": [
  "--all-scopes-completion", // □□□□(□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□),□□□
main()□□□□cout,□□□□`#include <iostream>`,□□□□`std::cout`□□□,□□"--header-
insertion=iwyu",□□□□□□□□□□□□
  "--background-index", // □□□□□□□□□□□
  "--clang-tidy", // □□ Clang-Tidy □□□□□□□□□
  "--clang-tidy-checks=performance-*, bugprone-*, misc-*, google-*, modernize-*,
readability-*, portability-*",
  "--compile-commands-dir=${workspaceFolder}/.vscode", // □□□□□
(compile_commands.json □□)□□□□□
  "--completion-parse=auto", // □ clangd □□□□□□□□□□□□□
  "--completion-style=detailed", // □□□□□□□(□□□□□□□□□□□□□);□□□□□□ detailed
  "--enable-config", // □□□□□□(YAML□□)
  "--fallback-style=LLVM", // □□□□□□□: □□□□ .clang-format □□□□□,□□□□ LLVM,
Google, Chromium, Mozilla, Webkit, Microsoft, GNU
  "--function-arg-placeholders=true", // □□□□□□□□□□□□□□□□□□□□□ Tab □□□□□□□□□□□□□
□□□
  "--header-insertion-decorators", // □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  "--header-insertion=never", // □□□□□□□□□□□□□
  "--log=verbose", // □ Clangd □□□□□□□□
  "--pch-storage=memory", // pch □□□□□(Memory □ Disk,□□□□□□□□□□□□□□□□)
  "--pretty", // □□□ JSON □□□□□
  "--ranking-model=decision_forest", // □□□□□□□□hueristics (□□□),
decision_forest (□□□)
  "-j=12", // □□□□□□□□□
  "--query-driver=C:/msys64/clang64/bin/clang*" // clang □□□□□□□□□□□□□□ clangd □
□□□□□□□
],
"clangd.checkUpdates": true, // □□□□ clangd □□
"clangd.onConfigChanged": "restart", // □□□□□□□□□□□
"clangd.detectExtensionConflicts": true, // □□□□□□ clangd □□□□□□□□□□
"editor.suggest.snippetsPreventQuickSuggestions": false, // clangd□snippets□□□□□
□□□□□□□□□□□□□□□Intellisense□
// LLDB
"lldb.commandCompletions": true, // LLDB □□□□□□
"lldb.dereferencePointers": true, // LLDB □□□□□□□□□
"lldb.evaluateForHovers": true, // LLDB □□□□□□□□□□□□□□
"lldb.launch.expressions": "native", // LLDB □□□□□□□□□□
"lldb.showDisassembly": "never", // LLDB □□□□□□□
"lldb.verboseLogging": true,
```

⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ C++ ⊠⊠⊠⊠⊠⊠

⊠⊠⊠⊠⊠

⊠⊠⊠⊠⊠⊠⊠⊠⊠ .clangd⊠.clang-tidy ⊠ .clang-format ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

- .clangd ⊠⊠⊠ ⊠⊠⊠ readability-identifier-naming ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

```
# .clangd
Diagnostics:
  ClangTidy:
    Add: [readability-identifier*]
    CheckOptions:
      # □□□□□□
      # □□□□□□□□□ https://clang.llvm.org/extra/clang-tidy/checks/readability/identifier-naming.html
      readability-identifier-naming.VariableCase: lower_case
      readability-identifier-naming.GlobalVariableCase: lower_case
      readability-identifier-naming.GlobalVariablePrefix: g_

      readability-identifier-naming.MemberPrefix: m_
      readability-identifier-naming.MemberCase: lower_case
      readability-identifier-naming.ClassMemberPrefix: m_
      readability-identifier-naming.ClassMemberCase: lower_case

      readability-identifier-naming.EnumCase: UPPER_CASE
      readability-identifier-naming.EnumConstantCase: UPPER_CASE
      readability-identifier-naming.MacroDefinitionCase: UPPER_CASE

      readability-identifier-naming.ClassCase: CamelCase
      readability-identifier-naming.StructCase: Camelcase

      readability-identifier-naming.FunctionCase: camelCase
      readability-identifier-naming.ClassMethodCase: camelCase
      readability-identifier-naming.MethodCase: camelCase

      readability-identifier-naming.ConstantCase: UPPER_CASE
      readability-identifier-naming.ConstantParameterCase: camel_Snake_Back

CompileFlags:
  Add:
    - '-pedantic'
    - '-Wall'
    - '-Wextra'
    - '-Wcast-align'
    - '-Wdouble-promotion'
    - '-Wformat=2'
    - '-Wimplicit-fallthrough'
    - '-Wmisleading-indentation'
    - '-Wnon-virtual-dtor'
    - '-Wnull-dereference'
    - '-Wold-style-cast'
    - '-Woverloaded-virtual'
    - '-Wpedantic'
    - '-Wshadow'
    - '-Wunused'
    - '-pthread'
    - '-fuse-ld=lld'
    - '-fsanitize=address'
    - '-fsanitize=undefined'
    - '-stdlib=libc++'
    - '-std=c++20' # □□□□□□□□
```

- .clang-tidy ⊠⊠⊠

```
# .clangd-tidy
---
Checks: "bugprone-*,\
  google-*,\
  misc-*,\
  modernize-*,\
  performance-*,\
  readability-*,\
  portability-*,\
  "
HeaderFilterRegex: 'Source/cm[^/]*\.(h|hxx|cxx)$'
CheckOptions:
  - key: modernize-use-default-member-init.UseAssignment
    value: '1'
  - key: modernize-use-equals-default.IgnoreMacros
    value: '0'
  - key: modernize-use-auto.MinTypeNameLength
    value: '80'
```

- .clang-format □□□ □□□□ BasedOnStyle □□□□□□□□□□□□□□□□□□

```
# .clang-format
---
## □□: None, Cpp, Java, JavaScript, ObjC, Proto, TableGen, TextProto
Language: Cpp
## □□□□
## BasedOnStyle:  LLVM
## □□□□□(public□private□)□□□
AccessModifierOffset: -4
## □□□(□□□□□□□□□□□□□□)□□□□: Align, DontAlign, AlwaysBreak(□□□□□□□□□)
AlignAfterOpenBracket: Align
## □□□□□
AlignArrayOfStructures: None
## □□□□□□□
AlignConsecutiveMacros: None
## □□□□□□□□□□□□
AlignConsecutiveAssignments: None
## □□□□□□□
AlignConsecutiveBitFields: None
## □□□□□□□□□□□□□□□□
AlignConsecutiveDeclarations: None
## □□□□□□□(□□□□□□□)□□□□
AlignEscapedNewlines: Left
## □□□□□□□□□□□□□□□□
AlignOperands: Align
## □□□□□□□□□□
AlignTrailingComments: true
## □□□□□□□□□□□□□□□□□
AllowAllParametersOfDeclarationOnNextLine: true
## □□□□□□□□□□□
AllowShortEnumsOnASingleLine: true
## □□□□□□□□□□
AllowShortBlocksOnASingleLine: Never
## □□□□case□□□□□□□
AllowShortCaseLabelsOnASingleLine: false
## □□□□□□□□□□□
AllowShortFunctionsOnASingleLine: All
## □□□□□□□□□□□□□
```

```
AllowShortLambdasOnASingleLine: All
## □□□□if□□□□□□□□
AllowShortIfStatementsOnASingleLine: Never
## □□□□□□□□□□□□
AllowShortLoopsOnASingleLine: false
## □□□□□□□□□□
AlwaysBreakAfterReturnType: None
## □□□□□string□□□□□□
AlwaysBreakBeforeMultilineStrings: false
## □□□template□□□□□
AlwaysBreakTemplateDeclarations: MultiLine
AttributeMacros:
  - __capability
## false□□□□□□□□□□□□□□□□□□□□□
BinPackArguments: true
## false□□□□□□□□□□□□□□□□□□□□□
BinPackParameters: true
## □□□□□□□□□BreakBeforeBraces□□□Custom□□□□
BraceWrapping:
  AfterCaseLabel: false
  AfterClass: false
  AfterControlStatement: Never
  AfterEnum: false
  AfterFunction: false
  AfterNamespace: false
  AfterObjCDeclaration: false
  AfterStruct: false
  AfterUnion: false
  AfterExternBlock: false
  BeforeCatch: false
  BeforeElse: false
  BeforeLambdaBody: false
  BeforeWhile: false
  IndentBraces: false
  SplitEmptyFunction: true
  SplitEmptyRecord: true
  SplitEmptyNamespace: true
## □□□□□□□□□
BreakBeforeBinaryOperators: None
## □concept□□□
BreakBeforeConceptDeclarations: true
## □□□□□□□: Attach(□□□□□□□□□□□□□□□)
BreakBeforeBraces: Attach
## □□□□□□
BreakInheritanceList: AfterComma
## □□□□□□□□□
BreakBeforeTernaryOperators: true
## □□□□□□□□□□□□□□
BreakConstructorInitializers: BeforeComma
## □java□□□□□□□□
BreakAfterJavaFieldAnnotations: false
## □□□□□□□□0□□□□□□
ColumnLimit: 96
## □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
CommentPragmas: '^ IWYU pragma:'
```

```
## □□□□□□□□□□□□
CompactNamespaces: false
## □□□□□□□□□□□□□□□
ConstructorInitializerIndentWidth: 4
## □□□□□□□□□
ContinuationIndentWidth: 4
## □□C++11□□□□□□□□□□{□□}□□□□
Cpp11BracedListStyle: true
## □□□□□□□□□□
DeriveLineEnding: true
## □□□□□□□□□□□□□□□□
DerivePointerAlignment: false
## □□□□□
DisableFormat: false
## □□□□□□□□□□□□□
EmptyLineAfterAccessModifier: Never
## □□□□□□□□□□□□□□□□□□□□□□
EmptyLineBeforeAccessModifier: LogicalBlock
## □□□□□□□□□□□□□□□□□□□□□□□□(Experimental)
ExperimentalAutoDetectBinPacking: false
## □□□□namespace□□
FixNamespaceComments: true
## □□□□□□foreach□□□□□□□□□□□
ForEachMacros:
  - foreach
  - Q_FOREACH
  - BOOST_FOREACH
IfMacros:
  - KJ_IF_MAYBE
## □□#include□□□□□□□□□□□□
IncludeBlocks: Merge
## □□□□□□□□□□□□□□□#include□□□□□□
IncludeCategories:
  - Regex: '^"(llvm|llvm-c|clang|clang-c)/'
    Priority: 2
    SortPriority: 0
    CaseSensitive: false
  - Regex: '^(<|"(gtest|gmock|isl|json)/)'
    Priority: 3
    SortPriority: 0
    CaseSensitive: false
  - Regex: '.*'
    Priority: 1
    SortPriority: 0
    CaseSensitive: false
IncludeIsMainRegex: '(Test)?$'
IncludeIsMainSourceRegex: ''
## □□□□□□□
IndentAccessModifiers: false
## □□case□□
IndentCaseLabels: false
## case □□□□□□□□□ case □□□□□□□□□
IndentCaseBlocks: false
## □□goto□□□
IndentGotoLabels: false
```

```
## □□□□□□□□
IndentPPDirectives: None
## □□□□□□□□□
IndentExternBlock: AfterExternBlock
## □□□□□□requires□□
IndentRequires: false
## □□□□
IndentWidth: 4
## □□□□□□□□□□□□□□□□□□□□□□□□□
IndentWrappedFunctionNames: false
## □□□□□□
InsertTrailingCommas: None
## □□JavaScript□□□□□
JavaScriptQuotes: Leave
## □□ JavaScript □□/□□□□
JavaScriptWrapImports: true
## □□□□□□□□□□
KeepEmptyLinesAtTheStartOfBlocks: true
## □□□ lambda □□□□ lambda □□
LambdaBodyIndentation: Signature
## □□□□□□□□□□□□□
MacroBlockBegin: ''
## □□□□□□□□□□□□□
MacroBlockEnd: ''
## □□□□□□□□□
MaxEmptyLinesToKeep: 1
## □□□□□□□
NamespaceIndentation: Inner
ObjCBinPackProtocolList: Auto
## □□ObjC□□□□□□
ObjCBlockIndentWidth: 4
ObjCBreakBeforeNestedBlockParam: true
## □ObjC□@property□□□□□□□
ObjCSpaceAfterProperty: false
## □ObjC□protocol□□□□□□□□□
ObjCSpaceBeforeProtocolList: true
PenaltyBreakAssignment: 2
PenaltyBreakBeforeFirstCallParameter: 19
PenaltyBreakComment: 300
PenaltyBreakFirstLessLess: 120
PenaltyBreakString: 1000
PenaltyBreakTemplateDeclaration: 10
PenaltyExcessCharacter: 1000000
PenaltyReturnTypeOnItsOwnLine: 60
PenaltyIndentedWhitespace: 0
## □□□□□: Left, Right, Middle
PointerAlignment: Left
## □□□□□□□□□□□
PPIndentWidth: -1
## □□□□□
ReferenceAlignment: Pointer
## □□□□□□□□
ReflowComments: true
## □□□□□□□□□□□□□□
ShortNamespaceLines: 1
```

```
## □□□□#include
SortIncludes: false
## java□□□□□□□□□□□□□
SortJavaStaticImport: Before
## □using□□□□
SortUsingDeclarations: true
## □C□□□□□□□□□□□
SpaceAfterCStyleCast: false
## □!□□□□□
SpaceAfterLogicalNot: false
## □Template□□□□□□□□
SpaceAfterTemplateKeyword: true
## □□□□□□□□□□□□
SpaceBeforeAssignmentOperators: true
## □□case□□□□□□□□
SpaceBeforeCaseColon: false
## □□C++11□□□□□□□□□□□
SpaceBeforeCpp11BracedList: false
## □□□□□□□□□□□□□□□□□
SpaceBeforeCtorInitializerColon: true
## □□□□□□□□□□
SpaceBeforeInheritanceColon: true
## □□□□□□□□□□□□: Never, ControlStatements, Always
SpaceBeforeParens: ControlStatements
## □□□□□□□□□□□□□□□□□□□ PointerAlignment
SpaceAroundPointerQualifiers: Default
## □□□□□□for□□□□□□□□□□
SpaceBeforeRangeBasedForLoopColon: true
## {}□□□□□□□
SpaceInEmptyBlock: false
## □□□□□□□□□□□
SpaceInEmptyParentheses: false
## □□□□□□□□□□□□□(□□□□//)
SpacesBeforeTrailingComments: 1
## □□□□□<□□>□□□□□
SpacesInAngles: Never
## □□if/for/switch/while□□□□□□□□
SpacesInConditionalStatement: false
## □□□(ObjC□JavaScript□□□□□□□)□□□□□□□□
SpacesInContainerLiterals: true
## □C□□□□□□□□□□□□□□
SpacesInCStyleCastParentheses: false
## □□□□□□□□□□□□□□□□□□□□□□□□□□□-1□□□□□□□□□□□□□□□
SpacesInLineCommentPrefix:
  Minimum: 1
  Maximum: -1
## □□□□□(□□)□□□□□
SpacesInParentheses: false
## □□□□□[□□]□□□□□□□□lamda□□□□□□□□□□□□□□□□□
SpacesInSquareBrackets: false
## □□[□□□□□
SpaceBeforeSquareBrackets: false
## □□:□□□□□□□
BitFieldColonSpacing: Both
## □□
```

```yaml
Standard: Auto
## [illegible]
StatementAttributeLikeMacros:
  - Q_EMIT
## [illegible]
StatementMacros:
  - Q_UNUSED
  - QT_REQUIRE_VERSION
## tab[illegible]
TabWidth: 4
## [illegible]\n[illegible]
UseCRLF: false
## [illegible]tab[illegible]ForIndentation——[illegible]
UseTab: Never
## [illegible]
WhitespaceSensitiveMacros:
  - STRINGIZE
  - PP_STRINGIZE
  - BOOST_PP_STRINGIZE
  - NS_SWIFT_NAME
  - CF_SWIFT_NAME
```

Xmake [illegible]

[illegible] `hello_world.cpp` [illegible] Hello world [illegible]

```cpp
// hello_world.cpp
#include <iostream>

int main() {
    std::cout << "Hello, world!" << std::endl;
    return 0;
}
```

[illegible] `xmake.lua` [illegible] Xmake [illegible]

```lua
-- xmake.lua
add_defines("ROOT")
add_rules("mode.debug", "mode.release")
set_toolchains("clang")
set_languages("cxx17","c11")
set_optimize("fastest")
set_warnings("all")

-- [illegible]
target("Hello world!")
    set_kind("binary")
    add_files("hello_world.cpp")
target_end()
```

[illegible] VSCode [illegible] Xmake [illegible] ▸ [illegible]Xmake [illegible] Clang [illegible] hello world [illegible]

[illegible] C++ [illegible] Xmake [illegible]

☒☒☒☒

☒☒☒☒☒☒☒☒☒☒☒☒☒

- [☒☒☒☒]Visual Studio Code ☒☒ C/C++ ☒☒☒☒☒☒☒☒☒(VSCode + Clangd + XMake) - ☒☒
- ☒☒☒☒☒ VSCode+clangd+lldb+cmake ☒☒ C/C++☒☒☒☒☒☒ - ☒☒
- ☒☒ VSCode + XMake + LLVM ☒☒☒☒ C++☒llvm-mingw + clangd + modules☒
- MSYS2 ☒ CMake ☒☒☒☒☒☒ | ☒☒☒☒☒☒