

# Living comfortably at HEAD with Bazel

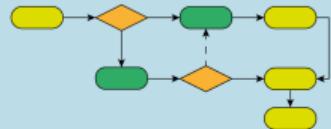
Šimon Tóth

November 19, 2022

<https://github.com/HappyCerberus/meetingcpp22-bazel>



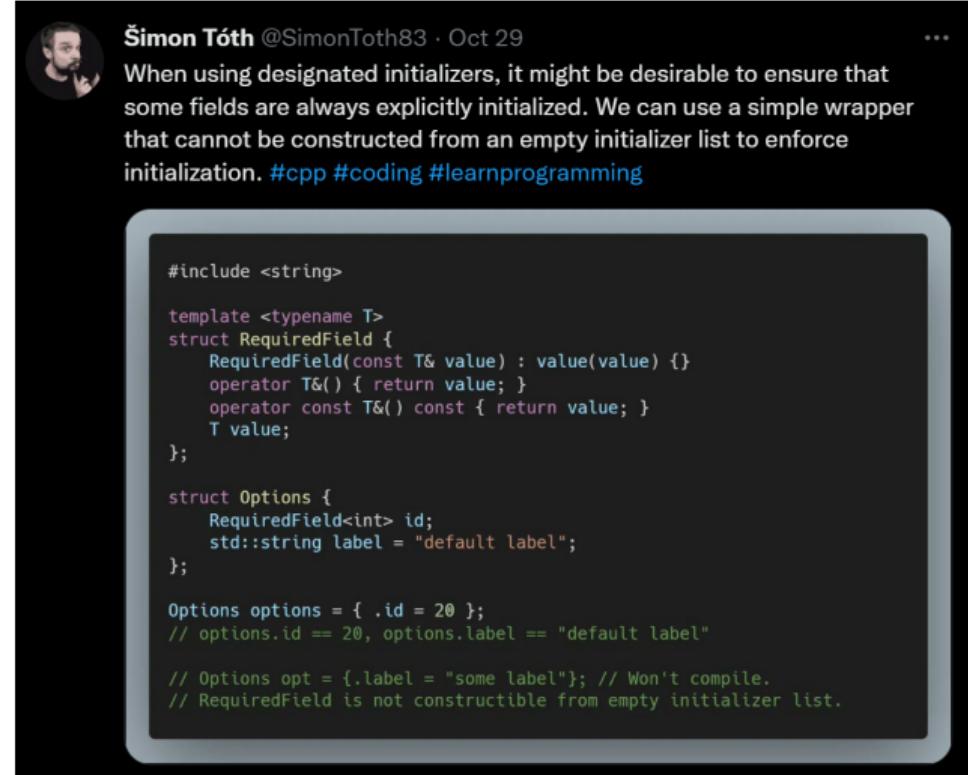
A Complete Guide to  
Standard C++ Algorithms



RNDr. Šimon Tóth

- ▶ Free on GitHub:  
[HappyCerberus/book-cpp-algorithms](https://github.com/HappyCerberus/book-cpp-algorithms)
- ▶ Donate to EFF on LeanPub:  
[leanpub.com/cpp-algorithms-guide](https://leanpub.com/cpp-algorithms-guide)

- ▶ [twitter.com/SimonToth83](https://twitter.com/SimonToth83)
- ▶ [linkedin.com/in/simontoth](https://linkedin.com/in/simontoth)



Simon Tóth @SimonToth83 · Oct 29

When using designated initializers, it might be desirable to ensure that some fields are always explicitly initialized. We can use a simple wrapper that cannot be constructed from an empty initializer list to enforce initialization. #cpp #coding #learnprogramming

```
#include <string>

template <typename T>
struct RequiredField {
    RequiredField(const T& value) : value(value) {}
    operator T&() { return value; }
    operator const T&() const { return value; }
    T value;
};

struct Options {
    RequiredField<int> id;
    std::string label = "default label";
};

Options options = { .id = 20 };
// options.id == 20, options.label == "default label"

// Options opt = {.label = "some label"}; // Won't compile.
// RequiredField is not constructible from empty initializer list.
```

# Monorepo vs Polyrepo

mono- one, single

mono- one, single

poly- many

@HEAD built from current sources

@HEAD built from current sources  
binary versioned binaries

## Comparison

@HEAD monorepo

binary polyrepo

# Comparison

@HEAD monorepo

- ▶ 😞 scalability

binary polyrepo

- ▶ 😢 ABI stability

# Comparison

## @HEAD monorepo

- ▶ 😞 scalability
- ▶ 😊 global operations are possible

## binary polyrepo

- ▶ 😢 ABI stability
- ▶ 😱 global operations do not exist

# Comparison

## @HEAD monorepo

- ▶ 😞 scalability
- ▶ 😏 global operations are possible
- ▶ 😬 private dependencies are non-trivial

## binary polyrepo

- ▶ 😢 ABI stability
- ▶ 😱 global operations do not exist
- ▶ 😊 private dependencies are trivial

# Comparison

## @HEAD monorepo

- ▶ 😞 scalability
- ▶ 😊 global operations are possible
- ▶ 😬 private dependencies are non-trivial
- ▶ 🤔 shared dependencies need to be carefully managed

## binary polyrepo

- ▶ 😢 ABI stability
- ▶ 😱 global operations do not exist
- ▶ 😊 private dependencies are trivial
- ▶ 😞 shared dependencies need to be carefully managed

# Comparison

@HEAD monorepo

binary polyrepo

# Comparison

@HEAD monorepo

- ▶ cooperative

binary polyrepo

- ▶ competitive

# Comparison

@HEAD monorepo

- ▶ cooperative
- ▶ conforming

binary polyrepo

- ▶ competitive
- ▶ independent

# Comparison

## @HEAD monorepo

- ▶ cooperative
- ▶ conforming
- ▶ biased towards users

## binary polyrepo

- ▶ competitive
- ▶ independent
- ▶ biased towards providers

# Bazel

# Hello World Demo

The sidebar contains several sections:

- EXPLORER**: Shows the file system structure.
- BAZEL BUILD TARGETS**: A tree view of Bazel build targets:
  - //artefacts
  - //hello\_world
  - //platforms
  - //sanitizers
  - //structured
- MEETINGCPP22-BAZEL**: A tree view of a meeting container:
  - bazel-out
  - bazel-testlogs
  - hello\_world
    - other\_hello
    - BUILD
  - main.cc
  - platforms
  - sanitizers
  - slides
  - structured
  - testing
- DEV CONTAINER**: Shows the current container is "meetingcpp22-bazel".
- STATUS**: Shows 0 errors, 0 warnings, and 1 info message.
- OUTPUT**: Shows the output of the last command run.
- PROBLEMS**: Shows 3 problems.
- SEARCH**: Shows the search interface.
- REFRESH**: Shows the refresh interface.
- HELP**: Shows the help interface.

## main.cc

hello\_world &gt; main.cc &gt; main()

```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello World!\n";
5 }
```





EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured
- > //wasm



3

## MEETIN...

- > structured
- > testing
- > wasm
- .bazelignore

.bazeliskrc

.bazelrc

.clang-tidy

.gitignore

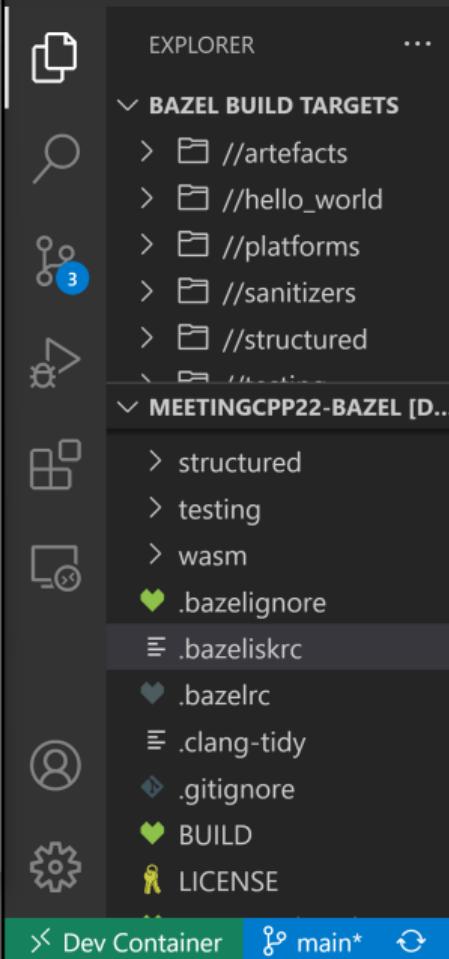
BUILD

LICENSE

.bazeliskrc X

.bazeliskrc

1 USE\_BAZEL\_VERSION=6.0.0-pre.20220922.1



-  EXPLORER
- ...
-  PROBLEMS
-  OUTPUT
-  DEBUG CONSOLE
-  TERMINAL
-  PORTS 1
-  bash
-  +
-  ↵
-  ⌘
-  X

 **BAZEL BUILD TARGETS**

- >  //artefacts
- >  //hello\_world
- >  //platforms
- >  //sanitizers
- >  //structured
- >  ...

 **MEETINGCPP22-BAZEL [D...**

- > structured
- > testing
- > wasm
-  .bazelignore
-  .bazeliskrc
-  .bazelrc
-  .clang-tidy
-  .gitignore
-  BUILD
-  LICENSE

● root@64b276268b89:/workspaces/meetingcpp22-bazel# bazel --version  
bazel 6.0.0-pre.20220922.1

○ root@64b276268b89:/workspaces/meetingcpp22-bazel#

EXPLORER ...  
 BAZEL BUILD TARGETS > //artefacts  
> //hello\_world  
> //platforms  
> //sanitizers  
> //structured  
...  
 MEETINGCPP22-BAZEL [D...] .bazelignore  
≡ .bazeliskrc  
♥ .bazelrc  
≡ .clang-tidy  
◊ .gitignore  
♥ BUILD  
👤 LICENSE  
♥ MODULE.bazel  
ⓘ README.md  
♥ WORKSPACE

WORKSPACE X  
 WORKSPACE  
1 workspace(name = "bazel\_demo")  
2  
3 # [3] · Testing  
4 load("@bazel\_tools//tools/build\_defs/repo:http.bzl", "http\_archive")  
5  
6 # Catch2  
7 http\_archive(  
8 name = "com\_github\_catchorg\_catch2",  
9 strip\_prefix = "Catch2-3.1.1",  
10 urls = [<https://github.com/catchorg/Catch2/archive/v3.1.1.tar.gz>],  
11 sha256 = "2106bccfec18c8ce673623d56780220e38527dd8f283ccba26aa4b"  
12 )  
13  
14 # [5] · Static analyzers  
15  
16 # Compilation database  
17 # <https://github.com/hedronvision/bazel-compile-commands-extractor>  
18 http\_archive(  
19 name = "hedron\_compile\_commands",  
20 url = "<https://github.com/hedronvision/bazel-compile-commands-extractor>"



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured

MEETINGCPP22-BAZEL [D...  
  .bazelignore

.bazeliskrc

.bazelrc

.clang-tidy

.gitignore

BUILD

LICENSE

MODULE.bazel

README.md

WORKSPACE

.bazelrc X

.bazelrc

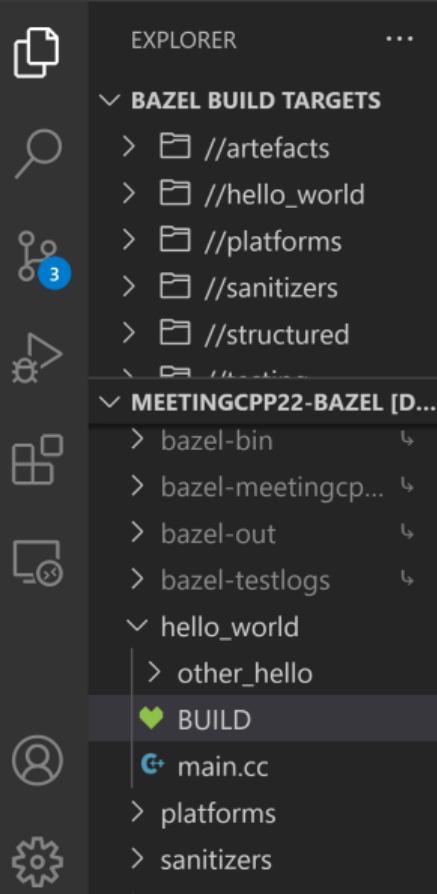
```
1  # [1] Hello World!
2  build --client_env=CC=clang-16
3  build --cxxopt="-std=c++20" --cxxopt="-pedantic" --cxxopt="-Wall"
   --cxxopt="-Wextra" --cxxopt="-Werror"
4
5  # [2] Structured Hello World!
6  build --features=layering_check
7  build --experimental_cc_implementation_deps
8
9  # [3] Testing
10 build --experimental_enable_bzlmod
11 build --java_runtime_version=remotejdk_18
12
13 # [4] Sanitizers
14 build:asan --cxxopt="-Og" --cxxopt="-g"
   --cxxopt="-fsanitize=address" --cxxopt="-fno-omit-frame-pointer"
15 build:asan --linkopt="-fsanitize=address"
16 build:asan --platform_suffix="-asan"
17
18 # [5] Static analyzers
```

The Explorer sidebar displays the following structure:

- BAZEL BUILD TARGETS**
  - > //artefacts
  - > //hello\_world
  - > //platforms
  - > //sanitizers
  - > //structured
  - > ...
- MEETINGCPP22-BAZEL [D...]**
  - > bazel-bin
  - > bazel-meetingcp...
  - > bazel-out
  - > bazel-testlogs
  - > hello\_world
    - > other\_hello
    - BUILD** (highlighted)
    - main.cc**
  - > platforms
  - > sanitizers
  - > slides

### BUILD

```
hello_world > BUILD > ...
1 cc_binary([
2   name = "hello",
3   srcs = ["main.cc"],
4 ])
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2 bash + ×
```

- root@64b276268b89:/workspaces/meetingcpp22-bazel# bazel build //hello\_world:hello  
INFO: Analyzed target //hello\_world:hello (0 packages loaded, 0 targets configured).  
INFO: Found 1 target...  
Target //hello\_world:hello up-to-date:  
 bazel-bin/hello\_world/hello  
INFO: Elapsed time: 0.122s, Critical Path: 0.00s  
INFO: 1 process: 1 internal.  
INFO: Build completed successfully, 1 total action
- root@64b276268b89:/workspaces/meetingcpp22-bazel#



EXPLORER ...

## PROBLEMS

## OUTPUT

DEBUG CONSOLE

## TERMINAL

PORTS

>

ash +

▼

1

X

## ▼ BAZEL BUILD TARGETS

- >  //artefacts
  - >  //hello\_world
  - >  //platforms
  - >  //sanitizers
  - >  //structured

MEETINGCPP22-BAZEL [D..

- ```
> bazel-bin          ↴  
> bazel-meetingcp... ↴  
> bazel-out         ↴  
> bazel-testlogs   ↴
```

```
● root@64b276268b89:/workspaces/meetingcpp22-bazel# bazel run //hello_world:hello
INFO: Analyzed target //hello_world:hello (0 packages loaded, 0 targets configured).
INFO: Found 1 target...
Target //hello_world:hello up-to-date:
    bazel-bin/hello_world/hello
INFO: Elapsed time: 0.118s, Critical Path: 0.00s
INFO: 1 process: 1 internal.
INFO: Build completed successfully, 1 total action
INFO: Build completed successfully, 1 total action
Hello World!
○ root@64b276268b89:/workspaces/meetingcpp22-bazel#
```

Repo, Package, Target

@repo//package/path:target

Repo, Package, Target

@// package/path:target

Repo, Package, Target

//package/path:target

Repo, Package, Target

:target

Repo, Package, Target

//package/path:path

Repo, Package, Target

// package/path

EXPLORER    ...

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS (2)    bash - hello\_world    +   

BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured
- > ...

MEETINGCPP22-BAZEL [D...]

- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs
- > hello\_world
  - > other\_hello
  - BUILD
  - main.cc
  - > platforms
  - > sanitizers
  - > slides

● root@64b276268b89:/workspaces/meetingcpp22-bazel# cd hello\_world/  
● root@64b276268b89:/workspaces/meetingcpp22-bazel/hello\_world# bazel run :hello  
INFO: Analyzed target //hello\_world:hello (0 packages loaded, 0 targets configured).  
INFO: Found 1 target...  
Target //hello\_world:hello up-to-date:  
    bazel-bin/hello\_world/hello  
INFO: Elapsed time: 0.101s, Critical Path: 0.00s  
INFO: 1 process: 1 internal.  
INFO: Build completed successfully, 1 total action  
INFO: Build completed successfully, 1 total action  
Hello World!  
○ root@64b276268b89:/workspaces/meetingcpp22-bazel/hello\_world#

Dev Container    main\*       0 △ 0    2    Ln 4, Col 2    Spaces: 4    UTF-8    LF    Starlark

The sidebar contains the following sections:

- EXPLORER**: Shows the Bazel build targets.
- BAZEL BUILD TARGETS**:
  - > //artefacts
  - > //hello\_world
  - > //platforms
  - > //sanitizers
  - > //structured
  - > //slides
- MEETINGCPP22-BAZEL [D...]**:
  - > bazel-testlogs
  - > hello\_world
    - > other\_hello
      - | BUILD
      - | main.cc
      - | BUILD
      - | main.cc
  - > platforms
  - > sanitizers
  - > slides
  - > structured

A **Dev Container** icon is present at the bottom left.

### BUILD

```
hello_world > other_hello > BUILD > ...
1 cc_binary([
2   name = "other_hello",
3   srcs = ["main.cc"],
4 ])
```



EXPLORER ...

## BAZEL BUILD TARGETS

- > `//artefacts`
- > `//hello_world`
- > `//platforms`
- > `//sanitizers`
- > `//structured`
- > `//.../other_hello`

## MEETINGCPP22-BAZEL [D...]

> `bazel-testlogs`> `hello_world`> `other_hello`  | `BUILD`  | `main.cc`  | `BUILD`  | `main.cc`> `platforms`> `sanitizers`> `slides`> `structured`

PROBLEMS

OUTPUT

TERMINAL

...

bash - hello\_world



```
● root@64b276268b89:/workspaces/meetingcpp22-bazel/hello_world# bazel run //hello_world/other_hello:other_hello
INFO: Analyzed target //hello_world/other_hello:other_hello (0 packages loaded,
0 targets configured).
INFO: Found 1 target...
Target //hello_world/other_hello:other_hello up-to-date:
  bazel-bin/hello_world/other_hello/other_hello
INFO: Elapsed time: 0.103s, Critical Path: 0.00s
INFO: 1 process: 1 internal.
INFO: Build completed successfully, 1 total action
INFO: Build completed successfully, 1 total action
Hello World!
```

○ root@64b276268b89:/workspaces/meetingcpp22-bazel/hello\_world#



EXPLORER ...

## BAZEL BUILD TARGETS

- > `//artefacts`
- > `//hello_world`
- > `//platforms`
- > `//sanitizers`
- > `//structured`
- > `//.../other_hello`

## MEETINGCPP22-BAZEL [D...]

> `bazel-testlogs`↳ < `hello_world`↳ < `other_hello`↳ < `BUILD`↳ < `main.cc`↳ < `BUILD`↳ < `main.cc`> `platforms`> `sanitizers`> `slides`> `structured`

PROBLEMS OUTPUT TERMINAL ...

bash - hello\_world + × ☰

- root@64b276268b89:/workspaces/meetingcpp22-bazel/hello\_world# `bazel run //hello_world/other_hello`  
`INFO:` Analyzed target `//hello_world/other_hello:other_hello` (0 packages loaded, 0 targets configured).  
`INFO:` Found 1 target...  
Target `//hello_world/other_hello:other_hello` up-to-date:  
  `bazel-bin/hello_world/other_hello/other_hello`  
`INFO:` Elapsed time: 0.113s, Critical Path: 0.00s  
`INFO:` 1 process: 1 internal.  
`INFO:` Build completed successfully, 1 total action  
`INFO:` Build completed successfully, 1 total action  
Hello World!
- root@64b276268b89:/workspaces/meetingcpp22-bazel/hello\_world#

# Structured Code



{ } c cpp properties.json 1 ● ⌂ hello.cc



```
> bazel-meetingcp... ↵      structured > ⚡ hello.cc > 📂 main()
> bazel-out          ↵      1     #include "greeter/greet.h"
> bazel-testlogs     ↵      2
> hello_world        ↵      3     int main() {
> platforms           ↵      4     ...     greeter::greet();
> ...
```

The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar contains icons for Explorer, Search, Open, Slides, and Settings. The main area displays a file named `c_cpp_properties.json` with the following content:

```
structured > BUILD > ...
1 package(default_visibility = ["//visibility:public"])
2
3 cc_binary(
4     name = "hello",
5     srcs = ["hello.cc"],
6     deps = [
7         "//structured/greeter:greet",
8     ]
9 )
10
11 cc_binary(
12     name = "hello_private",
13     srcs = ["hello_private.cc"],
14     deps = [
15         "//structured/greeter:greet",
16     ]
17 )
18
19 cc_binary(
20     name = "hello_transitive"
```

The file is 20 lines long and defines three `cc_binary` targets: `hello`, `hello_private`, and `hello_transitive`. The `hello` target depends on `//structured/greeter:greet`. The `hello_private` target depends on `//structured/greeter:greet`. The `hello_transitive` target is partially visible at the bottom.

 EXPLORER ...  c\_cpp\_properties.json 1  greet.cc   
         
BAZEL BUILD TARGETS > //artefacts > //hello\_world > //platforms > //sanitizers > //structured  
MEETINGCPP22-BAZEL [D...] > slides > structured > greeter > BUILD > format.cc > format.h > greet\_defs.h > greet.cc > greet.h > BUILD

structured > greeter > greet.cc > greeter  
1 #include "greet.h"  
2 #include "greet\_defs.h"  
3 #include "format.h"  
4  
5 #include <iostream>  
6  
7 namespace greeter {  
8 void greet() {  
9 std::cout << format(message);  
10 }  
11 }

< Dev Container  main   0  △ 1 

Ln 11, Col 2 Spaces: 4 UTF-8 LF C++ Linux  

EXPLORER ...

BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured
- > .....

MEETIN... D+ E+ ⌂ ⌂

- > slides
- > structured
  - > greeter

format.cc  
format.h  
greet\_defs.h  
greet.cc  
greet.h  
BUILD

BUILD

c\_cpp\_properties.json 1 ● BUILD X

structured > greeter > BUILD > ...

```
1 cc_library(  
2     name = "greet",  
3     srcs = [  
4         "greet.cc",  
5         "greet_defs.h"  
6     ],  
7     hdrs = ["greet.h"],  
8     implementation_deps = [  
9         ":format"  
10    ],  
11    visibility = ["//visibility:public"],  
12 )  
13  
14 cc_library(  
15     name = "format",  
16     srcs = ["format.cc"],  
17     hdrs = ["format.h"],  
18 )
```

Ln 10, Col 7 Spaces: 4 UTF-8 LF Starlark ↻ ⟲



EXPLORER

...

## BAZEL BUILD TARGETS

- > `//artefacts`
- > `//hello_world`
- > `//platforms`
- > `//sanitizers`
- > `//structured`
- > `...`

## MEETINGCPP22-BAZEL [D...]

- > slides
- > structured
  - > greeter
  - green heart icon `BUILD`
  - blue arrow icon `hello_private.cc`
  - blue arrow icon `hello_transitive.cc`
  - blue arrow icon `hello.cc`
- > testing
- > wasm
- green heart icon `.bazelignore`

PROBLEMS

1

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

2

bash

+

▼

□

×

```
● root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel run //structured:hello
INFO: Analyzed target //structured:hello (0 packages loaded, 0 targets configured).
INFO: Found 1 target...
Target //structured:hello up-to-date:
  bazel-bin/structured/hello
INFO: Elapsed time: 0.120s, Critical Path: 0.00s
INFO: 1 process: 1 internal.
INFO: Build completed successfully, 1 total action
INFO: Build completed successfully, 1 total action
>>> Hello World <<<
○ root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#
```



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers

## MEETINGCPP22-BAZEL [D...]

- > greeter
- ♥ BUILD
- ⌚ hello\_private.cc
- ⌚ hello\_transitive.cc
- ⌚ hello.cc
- > testing
- > wasm
- ♥ .bazelignore
- ≡ .bazeliskrc
- ♥ .bazelrc
- ≡ .clang-tidy

⌚ hello\_private.cc Xstructured > ⌚ hello\_private.cc > 📦 main()

```
1 #include "greeter/greet.h"
2 #include "greeter/greet_defs.h"
3
4 int main() {
5     greeter::greet();
6 }
```



EXPLORER ...

## BAZEL BUILD TARGETS

- > `//artefacts`
- > `//hello_world`
- > `//platforms`
- > `//sanitizers`
- > `//structured`
- > `//wasm`

## MEETINGCPP22-BAZEL [D...]

- > slides
- ✓ structured
  - > greeter
  - GREEN BUILD
  - ↳ `hello_private.cc`
  - ↳ `hello_transitive.cc`
  - ↳ `hello.cc`
- > testing
- > wasm
- GREEN `.bazelignore`

PROBLEMS

1

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

2

bash

+

-

X

configured).

**INFO:** Found 1 target...**ERROR:** /workspaces/meetingcpp22-bazel/structured/BUILD:11:10: Compiling structured/hello\_private.cc failed: (Exit 1): clang failed: error executing command (from target //structured:hello\_private) /usr/lib/llvm-16/bin/clang -U\_FORTIFY\_SOURCE -fstack-protector -Wall -Wthread-safety -Wself-assign -Wunused-but-set-parameter -Wno-free-nonheap-object -fcolor-diagnostics -fno-omit-frame-pointer ... (remaining 36 arguments skipped)

Use --sandbox\_debug to see verbose messages from the sandbox and retain the sandbox build root for debugging

**structured/hello\_private.cc:2:10: error:** use of private header from outside its module: 'greeter/greet\_defs.h' [-Wprivate-header]

#include "greeter/greet\_defs.h"

^

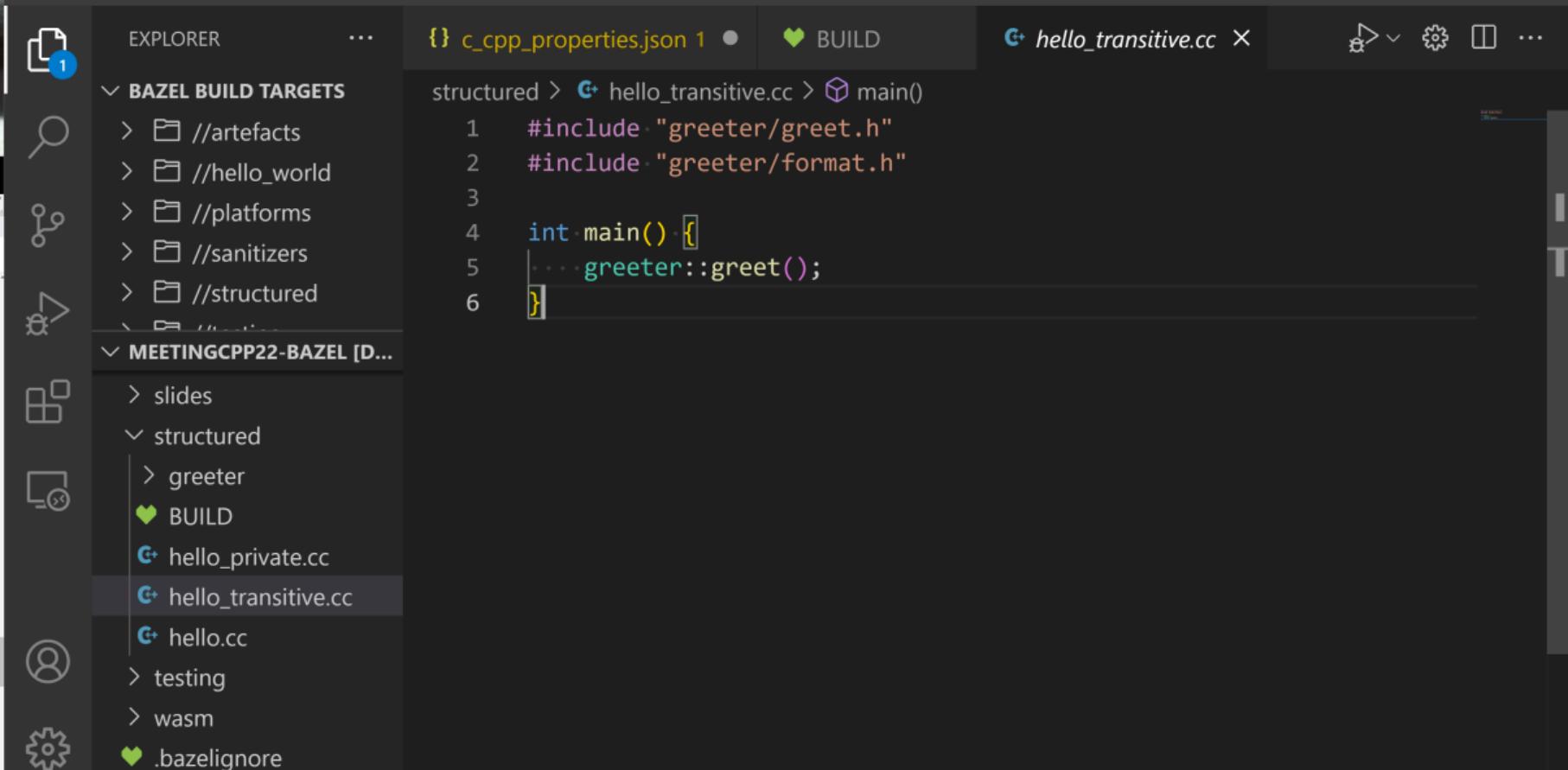
1 error generated.

Target //structured:hello\_private failed to build

Use --verbose\_failures to see the command lines of failed build steps.

**INFO:** Elapsed time: 0.384s, Critical Path: 0.24s**INFO:** 6 processes: 6 internal.**FAILED:** Build did NOT complete successfully

o root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#



The screenshot shows the Visual Studio Code (VS Code) interface running in a Dev Container. The title bar indicates the file is "hello\_transitive.cc" from the "meetingcpp22-bazel" repository. The left sidebar contains several icons: a file with a blue border, a magnifying glass, a circular arrow, a gear, and a person icon. The main area displays the code for "hello\_transitive.cc".

**EXPLORER** 

**BAZEL BUILD TARGETS**

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured
- > .....

**MEETINGCPP22-BAZEL [D...**

- > slides
- > structured
  - > greeter
  - BUILD**
  - hello\_private.cc**
  - hello\_transitive.cc**
  - hello.cc**
- > testing
- > wasm
- .bazelignore**

**c\_cpp\_properties.json** 1 ● **BUILD** **hello\_transitive.cc** X

structured > **hello\_transitive.cc** > **main()**

```
1 #include <greeter/greet.h>
2 #include <greeter/format.h>
3
4 int main() {
5     greeter::greet();
6 }
```





EXPLORER

...

## BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured

## MEETINGCPP22-BAZEL [D...

- /wdstt
- .bazelignore
- .bazeliskrc
- .bazelrc
- .clang-tidy
- .gitignore
- BUILD
- LICENSE
- MODULE.bazel
- README.md
- WORKSPACE

{ } c\_cpp\_properties.json 1

● BUILD

● .bazelrc X

...

## .bazelrc

```
1  # [1] Hello World!
2  build --client_env=CC=clang-16
3  build --cxxopt="-std=c++20" --cxxopt="-pedantic" --cxxopt="-Wall" --
4
5  # [2] Structured Hello World!
6  build --features=layering_check
7  build --experimental_cc_implementation_deps
8
9  # [3] Testing
10 build --experimental_enable_bzlmod
11 build --java_runtime_version=remotejdk_18
12
13 # [4] Sanitizers
14 build:asan --cxxopt="-Og" --cxxopt="-g" --cxxopt="-fsanitize=address"
15 build:asan --linkopt="-fsanitize=address"
16 build:asan --platform_suffix="-asan"
17
18 # [5] Static analyzers
19 # Required for bazel_clang_tidy to operate as expected
20 build:clang-tidy --aspects @bazel_clang_tidy//clang_tidy:clang_tidy
```

# Testing



EXPLORER

...



## BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured



## MEETIN...

.bazelignore



.bazeliskrc



.bazelrc



.clang-tidy



.gitignore



BUILD



LICENSE



MODULE.bazel

README.md

WORKSPACE

MODULE.bazel X

MODULE.bazel

```
1 # [3] Testing
2 bazel_dep(name = "googletest", version = "1.11.0")
3 # Required for Catch2
4 bazel_dep(name = "bazel_skylib", version = "1.3.0")
5
6 # [6] Platform support
7 bazel_dep(name = "platforms", version = "0.0.5")
```



## WORKSPACE X

### WORKSPACE

```
1 workspace(name = "bazel_demo")
2
3 # [3] Testing
4 load("@bazel_tools//tools/build_defs/repo:http.bzl", "http_archive")
5
6 # Catch2
7 http_archive(
8     name = "com_github_catchorg_catch2",
9     strip_prefix = "Catch2-3.1.1",
10    urls = ["https://github.com/catchorg/Catch2/archive/v3.1.1.tar.gz"],
11    sha256 = "2106bccfec18c8ce673623d56780220e38527dd8f283ccba26aa4b"
12 )
13
14 # [5] Static analyzers
15
16 # Compilation database
17 # https://github.com/hedronvision/bazel-compile-commands-extractor
18 http_archive(
19     name = "hedron_compile_commands",
20     url = "https://github.com/hedronvision/bazel-compile-commands-e...
```



EXPLORER ...

BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured

MEETINGCPP22-BAZEL [D...]

- BLACKBOX\_TEST.LIST
- BUILD
- data\_test.cc
- file.txt
- library\_catch\_test.cc
- library\_test.cc
- library.cc
- library.h
- > wasm
- .bazelignore
- .bazeliskrc

## BUILD

```
testing > BUILD > ...
1 cc_library(
2     name = "library",
3     srcs = ["library.cc"],
4     hdrs = ["library.h"],
5 )
6
7 cc_test(
8     name = "library_test",
9     srcs = ["library_test.cc"],
10    size = "small",
11    deps = [
12        ":library",
13        "@googletest//:gtest",
14        "@googletest//:gtest_main",
15    ]
16 )
17
18 cc_test(
19     name = "library_catch_test",
20     srcs = ["library_catch_test.cc"]
```





EXPLORER

BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured
- > ...

MEETINGCPP22-BAZEL [D...]

- BLACKBOX\_TEST.LIST
- BUILD
- data\_test.cc
- file.txt
- library\_catch\_test.cc
- library\_test.cc
- library.cc
- library.h
- > wasm
- .bazelignore
- .bazeliskrc

BUILD

testing > BUILD > ...

```
14     "@googletest//:gtest_main",
15     ...
16 )
17
18 cc_test(
19     name = "library_catch_test",
20     srcs = ["library_catch_test.cc"],
21     size = "small",
22     deps = [
23         ":library",
24         "@com_github_catchorg_catch2//:catch2",
25         "@com_github_catchorg_catch2//:catch2_main"
26     ]
27 )
28
29 cc_binary(
30     name = "binary",
31     srcs = ["binary.cc"],
32 )
33
```

The code editor shows a Bazel BUILD file. It defines a cc\_test target named 'library\_catch\_test' with source code from 'library\_catch\_test.cc'. It also defines a cc\_binary target named 'binary' with source code from 'binary.cc'. The code is syntax-highlighted with colors for different language constructs.





EXPLORER ...

BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured
- > .....

MEETINGCPP22-BAZEL [D...]

- Blackbox\_test.sh
- BUILD
- data\_test.cc
- file.txt
- library\_catch\_test.cc
- library\_test.cc
- library.cc
- library.h
- > wasm
- .bazelignore
- .bazeliskrc

## BUILD

```
testing > BUILD > ...
28
29 cc_binary(
30     name = "binary",
31     srcs = [ "binary.cc", ],
32 )
33
34 sh_test(
35     name = "blackbox_test",
36     size = "small",
37     srcs = [ "blackbox_test.sh" ],
38     data = [":binary"],
39     args = ["$(location :binary)"],
40 )
41
42 cc_test(
43     name = "data_test",
44     srcs = [ "data_test.cc" ],
45     deps = [ "@bazel_tools//tools/cpp/runfiles" ],
46     data = [ "file.txt" ],
```



EXPLORER ...

BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured
- > ...

MEETINGCPP22-BAZEL [D...]

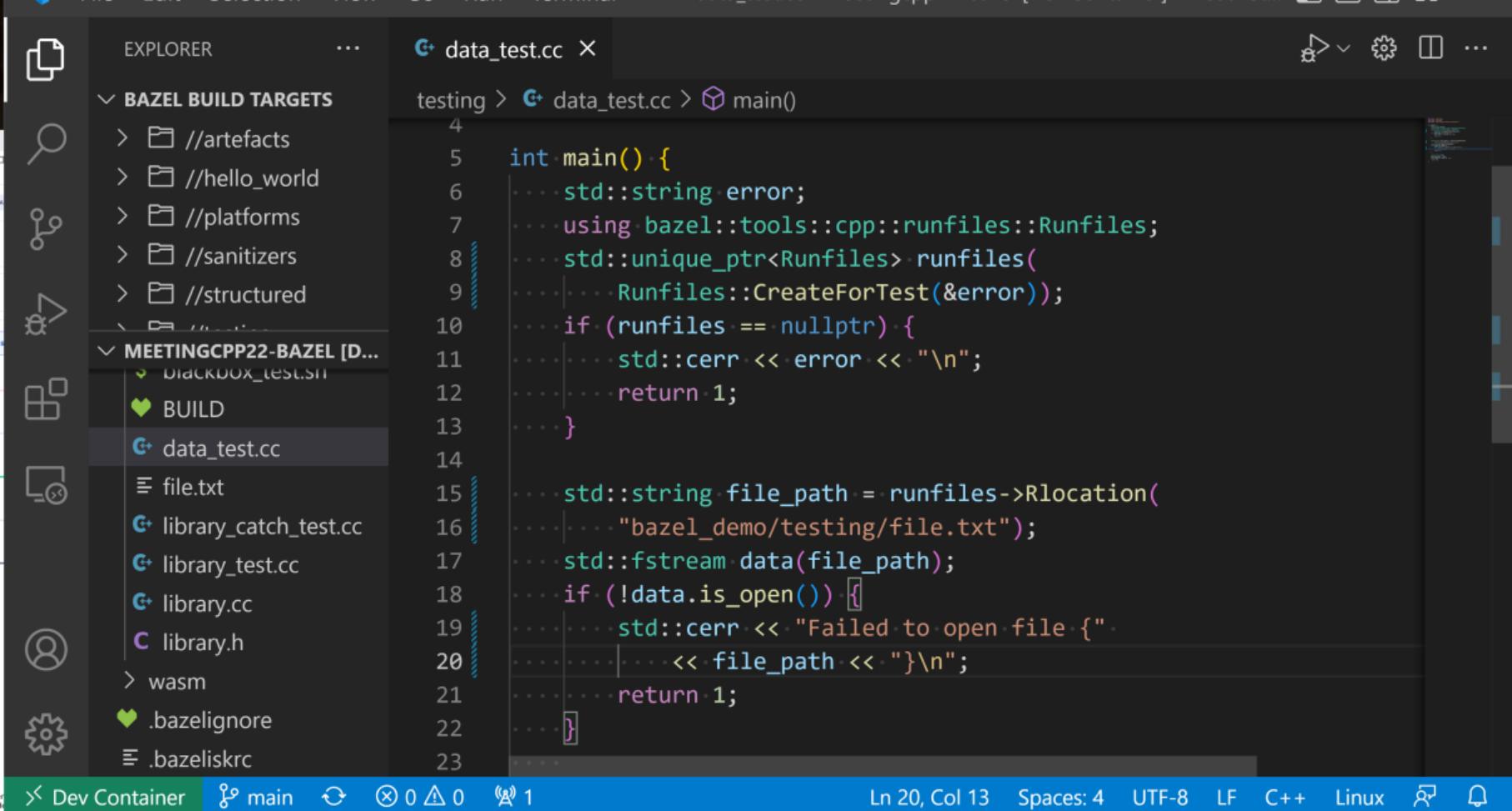
- ↳ BAZELDOX\_TEST.LST
- ↳ BUILD
- ↳ data\_test.cc
- ≡ file.txt
- ↳ library\_catch\_test.cc
- ↳ library\_test.cc
- ↳ library.cc
- C library.h
- > wasm
- ↳ .bazelignore
- ≡ .bazeliskrc

BUILD X

testing > BUILD > ...

```
41
42 cc_test([
43     name = "data_test",
44     srcs = ["data_test.cc"],
45     deps = ["@bazel_tools//tools/cpp/runfiles"],
46     data = ["file.txt"],
47 ])
```





EXPLORER

BAZEL BUILD TARGETS

- > //artefacts
- > //hello\_world
- > //platforms
- > //sanitizers
- > //structured

MEETINGCPP22-BAZEL [D...]

- BLACKBOX\_TEST.LIST
- BUILD
- data\_test.cc**
- file.txt
- libraryCatch\_test.cc
- library\_test.cc
- library.cc
- library.h
- wasm
- .bazelignore
- .bazeliskrc

data\_test.cc

testing > data\_test.cc > main()

```
4
5 int main() {
6     std::string error;
7     using bazel::tools::cpp::runfiles::Runfiles;
8     std::unique_ptr<Runfiles> runfiles(
9         Runfiles::CreateForTest(&error));
10    if (runfiles == nullptr) {
11        std::cerr << error << "\n";
12        return 1;
13    }
14
15    std::string file_path = runfiles->Rlocation(
16        "bazel_demo/testing/file.txt");
17    std::fstream data(file_path);
18    if (!data.is_open()) {
19        std::cerr << "Failed to open file {" <<
20                    file_path << "}\n";
21        return 1;
22    }
23
```

Ln 20, Col 13 Spaces: 4 UTF-8 LF C++ Linux



E ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

1

bash

+

✖

✖

✖

✖

- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs
- > hello\_world
- > platforms
- > sanitizers
- > slides
- > structured
- > testing
- > wasm
- .bazelignore
- ≡ .bazeliskrc
- .bazelrc
- ≡ .clang-tidy
- ◆ .gitignore
- BUILD
- LICENSE

```
● root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel test //testing:data_test
INFO: Analyzed target //testing:data_test (77 packages loaded, 1006 targets configured).
INFO: Found 1 test target...
Target //testing:data_test up-to-date:
  bazel-bin/testing/data_test
INFO: Elapsed time: 4.622s, Critical Path: 1.10s
INFO: 14 processes: 8 internal, 6 processwrapper-sandbox.
INFO: Build completed successfully, 14 total actions
//testing:data_test
D in 0.0s
PASSED

Executed 1 out of 1 test: 1 test passes.
There were tests whose specified size is too big. Use the --test_verbose_timeout_in_s
INFO: Build completed successfully, 14 total actions
○ root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#
```



E ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs
- > hello\_world
- > platforms
- > sanitizers
- > slides
- > structured
- > testing
- > wasm
- .bazelignore
- .bazeliskrc
- .bazelrc
- .clang-tidy
- .gitignore
- BUILD
- LICENSE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

1

bash

+

-

I

X

Executed 1 out of 1 test: 1 test passes.

There were tests whose specified size is too big. Use the --test\_verbose\_timeout\_s environment variable to increase the timeout.

INFO: Build completed successfully, 14 total actions

● root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel test //testing/...

INFO: Analyzed 6 targets (2 packages loaded, 334 targets configured).

INFO: Found 2 targets and 4 test targets...

INFO: Elapsed time: 12.266s, Critical Path: 2.52s

INFO: 160 processes: 30 internal, 130 processwrapper-sandbox.

INFO: Build completed successfully, 160 total actions

//testing:data\_test

(cached) PASSED

D in 0.0s

//testing:blackbox\_test

PASSED

D in 0.0s

//testing:library\_catch\_test

PASSED

D in 0.0s

//testing:library\_test

PASSED

D in 0.0s

Executed 3 out of 4 tests: 4 tests pass.

There were tests whose specified size is too big. Use the --test\_verbose\_timeout\_s environment variable to increase the timeout.

INFO: Build completed successfully, 160 total actions

○ root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#

# Sanitizers and Static Analyzers



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers
- > //...



## MEETINGCPP22-BAZEL [D...]

- ↳ .bazelignore



.bazeliskrc



.bazelrc



.clang-tidy



.gitignore



BUILD



LICENSE



MODULE.bazel



README.md



WORKSPACE

.bazelrc X

.bazelrc

```
11 build --java_runtime_version=remotejdk_18
12
13 # [4] Sanitizers
14 build:asan --cxxopt="-Og" --cxxopt="-g" --cxxopt="-fsanitize=address"
15 build:asan --linkopt="--fsanitize=address"
16 build:asan --platform_suffix="-asan"
17
18 # [5] Static analyzers
19 # Required for bazel_clang_tidy to operate as expected
20 build:clang-tidy --aspects //clang_tidy:clang_tidy.bzl%clang_tidy_aspects
21 build:clang-tidy --output_groups=report
22 # Optionally override the .clang-tidy config file target
23 build:clang-tidy --@bazel_clang_tidy//:clang_tidy_config//:clang_tidy_config
24
25 # [6] Platform support
26 build --incompatible_enable_cc_toolchain_resolution
27
28 # [7] WASM
29 build:wasm --platforms=@emsdk//:platform_wasm
```



EXPLORER

...

## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers

## MEETINGCPP22-BAZEL [D...]

- ↳ .bazelignore
- ≡ .bazeliskrc
- ♥ .bazelrc
- ≡ .clang-tidy
- ◊ .gitignore
- ♥ BUILD
- ♫ LICENSE
- ♥ MODULE.bazel
- ⓘ README.md
- ♥ WORKSPACE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

2

bash

+

-

X

```
root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel run //sanitizers:use_after_free --config asan
Starting local Bazel server and connecting to it...
INFO: Analyzed target //sanitizers:use_after_free (64 packages loaded, 268 targets configured).
INFO: Found 1 target...
Target //sanitizers:use_after_free up-to-date:
  bazel-bin/sanitizers/use_after_free
INFO: Elapsed time: 5.913s, Critical Path: 0.28s
INFO: 8 processes: 6 internal, 2 processwrapper-sandbox.
INFO: Build completed successfully, 8 total actions
INFO: Build completed successfully, 8 total actions
=====
==5253==ERROR: AddressSanitizer: heap-use-after-free on address 0x614000000044
  at pc 0x561dedb07e9f bp 0x7ffe901b5810 sp 0x7ffe901b5808
READ of size 4 at 0x614000000044 thread T0
    #0 0x561dedb07e9e in main (/root/.cache/bazel/_bazel_root/76b162f7ce1faedd
e3a9d91bdb798ae7/execroot/bazel_demo/bazel-out/k8-fastbuild--asan/bin/sanitiz
ers/use_after_free+0x125e9e) (BuildId: a47d1e847c660546)
    #1 0x7f57e61ead8f  (/lib/x86_64-linux-gnu/libc.so.6+0x29d8f) (BuildId: 693
89d485a9793dbe873f0ea2c93e02efaa9aa3d)
    #2 0x7f57e61eae3f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x
```



EXPLORER ...

BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers

MEETINGCPP22-BAZEL [D...]

- ↳ .bazelignore
- ≡ .bazeliskrc
- ♥ .bazelrc
- ≡ .clang-tidy
- ◊ .gitignore
- GREEN HEART BUILD
- PERSON LICENSE
- GREEN HEART MODULE.bazel
- INFO README.md
- GREEN HEART WORKSPACE

GREEN HEART WORKSPACE X

```
GREEN HEART WORKSPACE
10   .... urls = [
11     .... sha256 = "2106bccfec18c8ce673623d56780220e38527dd8f283ccba26aa41"
12   )
13
14   # [5] Static analyzers
15
16   # Compilation database
17   # https://github.com/hedronvision/bazel-compile-commands-extractor
18   http_archive(
19     .... name = "hedron_compile_commands",
20     .... url = "https://github.com/hedronvision/bazel-compile-commands-extractor-7082b8e31e6a4fe"
21     .... strip_prefix = "bazel-compile-commands-extractor-7082b8e31e6a4fe"
22     .... sha256 = "5591c1da51f0c824b6186bfe674c868eef135e8740467f851c79f"
23   )
24   load("@hedron_compile_commands//:workspace_setup.bzl", "hedron_compile_commands_setup")
25   hedron_compile_commands_setup()
26   # To run: bazel run @hedron_compile_commands//:refresh_all
27
28   # Clang-tidy direct integration
29   load("@bazel_tools//tools/build_defs/repo:git.bzl", "git_repository")
```



EXPLORER

...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers



## MEETINGCPP22-BAZEL [D...]

- ↳ .bazelignore



.bazeliskrc



.bazelrc



.clang-tidy



.gitignore



BUILD



{ compile\_commands.json }



LICENSE



MODULE.bazel

README.md

WORKSPACE

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

2

bash

+

X

- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel run @hedron\_compile\_commands//:refresh\_all  
**INFO:** Build option --features has changed, discarding analysis cache.  
**INFO:** Analyzed target @hedron\_compile\_commands//:refresh\_all (0 packages loaded, 522 targets configured).  
**INFO:** Found 1 target...  
Target @hedron\_compile\_commands//:refresh\_all up-to-date:  
    bazel-bin/external/hedron\_compile\_commands/refresh\_all.py  
    bazel-bin/external/hedron\_compile\_commands/refresh\_all  
**INFO:** Elapsed time: 0.435s, Critical Path: 0.00s  
**INFO:** 1 process: 1 internal.  
**INFO:** Build completed successfully, 1 total action  
**INFO:** Running command line: bazel-bin/external/hedron\_compile\_commands/refresh\_all  
**INFO:** Build completed successfully, 1 total action  
->>> Analyzing commands used in @//...  
->>> Finished extracting commands for @//...

○ root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#



- EXPLORER
- ...
- BAZEL BUILD TARGETS
  - >  //artefacts
  - >  //clang\_tidy
  - >  //hello\_world
  - >  //platforms
  - >  //sanitizers
  - >  //...
- MEETINGCPP22-BAZEL [D...
  - ↳  .bazelignore
  - ≡  .bazeliskrc
  - ♥  .bazelrc
  - ≡  .clang-tidy
  - ◊  .gitignore
  - GREEN  BUILD
  - { }  compile\_commands.json
  - PERSON  LICENSE
  - GREEN  MODULE.bazel
  - INFO  README.md
  - GREEN  WORKSPACE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2  bash    

- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# ls -la compile\_commands.json  
-rw-r--r-- 1 root root 2297308 Nov 6 09:31 compile\_commands.json
- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# 

EXPLORER ...

BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers
- > //...

MEETINGCPP22-BAZEL [D...]

- ↳ .bazelignore
- ↳ .bazeliskrc
- ↳ .bazelrc
- ↳ .clang-tidy
- ↳ .gitignore
- ↳ BUILD
- ↳ compile\_commands.json
- ↳ LICENSE
- ↳ MODULE.bazel
- ↳ README.md
- ↳ WORKSPACE

WORKSPACE X

```
WORKSPACE
24   load("@hedron_compile_commands//:workspace_setup.bzl", "hedron_compile_commands_setup")
25   hedron_compile_commands_setup()
26   # To run: bazel run @hedron_compile_commands//:refresh_all
27
28   # Clang-tidy direct integration
29   load("@bazel_tools//tools/build_defs/repo:git.bzl", "git_repository")
30
31   git_repository(
32     name = "bazel_clang_tidy",
33     commit = "41d3033cce4f6e42f7b683894e7b78ecf8491417",
34     remote = "https://github.com/erenon/bazel_clang_tidy.git",
35     shallow_since = "1665996490 +0200"
36   )
37
38   # [7] WASM
39   # Patched emsdk
40   http_archive(
41     name = "emscripten",
42     sha256 = "9c96e51385250bb463f14a8b5b23a394fe0ab2cefba413567d4c92"
```



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers



## MEETINGCPP22-BAZEL [D...]

- > testing
- > wasm
- ♥ .bazelignore
- ≡ .bazeliskrc
- ♥ .bazelrc
- ≡ .clang-tidy
- ❖ .gitignore
- ♥ BUILD
- { } compile\_commands.json
- 👤 LICENSE

WORKSPACE

.bazelrc

.bazelrc

```
11 build --java_runtime_version=remotejdk_18
12
13 # [4] Sanitizers
14 build:asan --cxxopt="-Og" --cxxopt="-g" --cxxopt="-fsanitize=address"
15 build:asan --linkopt="--fsanitize=address"
16 build:asan --platform_suffix="-asan"
17
18 # [5] Static analyzers
19 # Required for bazel_clang_tidy to operate as expected
20 build:clang-tidy --aspects //clang_tidy:clang_tidy.bzl%clang_tidy_as...
21 build:clang-tidy --output_groups=report
22 # Optionally override the .clang-tidy config file target
23 build:clang-tidy --@bazel_clang_tidy//:clang_tidy_config//:clang_t...
24
25 # [6] Platform support
26 build --incompatible_enable_cc_toolchain_resolution
27
28 # [7] WASM
29 build:wasm --platforms=@emsdk//:platform_wasm
```



EXPLORER

...

## BAZEL BUILD TARGETS

- > `//artefacts`
- > `//clang_tidy`
- > `//hello_world`
- > `//platforms`
- > `//sanitizers`

MEETINGCPP22-BAZEL [D...]  
/structured

- > testing
- > wasm
- `.bazelignore`
- ≡ `.bazeliskrc`
- `.bazelrc`
- ≡ `.clang-tidy`
- ◆ `.gitignore`
- `BUILD`
- { } `compile_commands.json`
- `LICENSE`

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

2

bash

+

▼

■

✖

```
● root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel build //hello_world:hello --config clang-tidy
INFO: Build option --@bazel_clang_tidy//:clang_tidy_config has changed, discarding analysis cache.
INFO: Analyzed target //hello_world:hello (1 packages loaded, 515 targets configured).
INFO: Found 1 target...
INFO: From Run clang-tidy on hello_world/main.cc:
15381 warnings generated.
Suppressed 15378 warnings (15378 in non-user code).
Use -header-filter=.* to display errors from all non-system headers. Use -system-headers to display errors from system headers as well.
warning: redefining builtin macro [clang-diagnostic-builtin-macro-redefined]
Aspect //clang_tidy:clang_tidy.bzl%clang_tidy_aspect of //hello_world:hello up-to-date:
bazel-bin/hello_world/bazel_clang_tidy_hello_world/main.cc.hello clang-tidy.yaml
INFO: Elapsed time: 2.128s, Critical Path: 1.68s
INFO: 5 processes: 4 internal, 1 processwrapper-sandbox.
INFO: Build completed successfully, 5 total actions
○ root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#
```



EXPLORER ...



## BAZEL BUILD TARGETS

- > `//artefacts`
- > `//clang_tidy`
- > `//hello_world`
- > `//platforms`
- > `//sanitizers`
- > `//third_party`

MEETINGCPP22-BAZEL [D...]  
/root/.cache/bazel/\_bazel\_root/76b162f7ce1faedde3a9d91bdb798ae7/sandbox/proces

- > `testing`
- > `wasm`
- `.bazelignore`
- ≡ `.bazeliskrc`
- `.bazelrc`
- ≡ `.clang-tidy`
- ◆ `.gitignore`
- `BUILD`
- { } `compile_commands.json`
- `LICENSE`



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2

 bash + ×

Use `--sandbox_debug` to see verbose messages from the sandbox and retain the sandbox build root for debugging  
36058 warnings generated.

Suppressed 36065 warnings (36053 in non-user code, 12 NOLINT).  
Use `-header-filter=.*` to display errors from all non-system headers. Use `-system-headers` to display errors from system headers as well.  
1 warning treated as error

**warning:** redefining builtin macro [clang-diagnostic-builtin-macro-redefined]  
`/root/.cache/bazel/_bazel_root/76b162f7ce1faedde3a9d91bdb798ae7/sandbox/proces`  
`wrapper-sandbox/8/execroot/bazel_demo/testing/library_test.cc:7:30: error:` 5  
is a magic number; consider replacing it with a named constant [cppcoreguidelines-avoid-magic-numbers,-warnings-as-errors]

```
std::vector<int> data = {5, 4, 3, 2, 1};  
^
```

Aspect `//clang_tidy:clang_tidy.bzl%clang_tidy_aspect` of `//testing:library_test`  
failed to build

Use `--verbose_failures` to see the command lines of failed build steps.

**INFO:** Elapsed time: 4.965s, Critical Path: 4.73s

**INFO:** 2 processes: 2 internal.

**FAILED:** Build did NOT complete successfully

○ `root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#` █

# Platforms



EXPLORER ...

## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers

## MEETINGCPP22-BAZEL [D...]

- > external
- > hello\_world
- ✓ platforms
  - ✓ BUILD
  - ↳ library\_test.cc
  - ↳ library.cc
  - ↳ library.h
  - ↳ multi\_platform.cc
  - ↳ platform\_linux.cc
  - ↳ platform\_windows.cc

BUILD X

platforms &gt; BUILD &gt; ...

```
1 cc_library(  
2     name = "library",  
3     srcs = ["library.cc"],  
4     hdrs = ["library.h"],  
5 )  
6  
7 # Windows/x86_64-only test  
8 cc_test(  
9     name = "library_test",  
10    srcs = ["library_test.cc"],  
11    deps = [":library"],  
12    target_compatible_with = [  
13        "@platforms//os:windows",  
14        "@platforms//cpu:x86_64"  
15    ]  
16 )  
17  
18 cc_library(  
19     name = "platform_windows",  
20     srcs = ["platform_windows.cc"],  
21 )
```



EXPLORER ...

## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers

## MEETINGCPP22-BAZEL [D...

- > external ↴
- > hello\_world
- platforms
  - BUILD
  - library\_test.cc
  - library.cc
  - library.h
  - multi\_platform.cc
  - platform\_linux.cc
  - platform\_windows.cc

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2

bash +

- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel test //platforms/...  
INFO: Build option --@bazel\_clang\_tidy//:clang\_tidy\_config has changed, discarding analysis cache.  
INFO: Analyzed 5 targets (1 packages loaded, 435 targets configured).  
INFO: Found 4 targets and 1 test target...  
INFO: Elapsed time: 0.825s, Critical Path: 0.44s  
INFO: 19 processes: 11 internal, 8 processwrapper-sandbox.  
INFO: Build completed successfully, 19 total actions  
//platforms:library\_test  
 SKIPPE

- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#



EXPLORER ...

BUILD

multi\_platform.cc X



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers

## MEETINGCPP22-BAZEL [D...

- BUILD
- library\_test.cc
- library.cc
- library.h
- multi\_platform.cc
- platform\_linux.cc
- platform\_windows.cc
- platform.h
- > sanitizers
- > slides

platforms &gt; multi\_platform.cc &gt; main()

```
1 #include "platform.h"
2 #include <iostream>
3
4 int main() {
5     std::cout << platform::get_platform() << "\n";
6 }
```



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers



## MEETINGCPP22-BAZEL [D...]

- BUILD
- library\_test.cc
- library.cc
- library.h
- multi\_platform.cc
- platform\_linux.cc
- platform\_windows.cc
- platform.h
- > sanitizers
- > slides



BUILD

platform\_windows.cc X



platforms &gt; platform\_windows.cc &gt; {} platform

```
1 #include "platform.h"
2
3 namespace platform {
4     std::string get_platform() {
5         return "this is windows";
6     }
7 }
```



EXPLORER ...

## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers

## MEETINGCPP22-BAZEL [D...]

- BUILD
- library\_test.cc
- library.cc
- library.h
- multi\_platform.cc
- platform\_linux.cc
- platform\_windows.cc
- platform.h
- > sanitizers
- > slides



BUILD

platform\_linux.cc X



platforms &gt; platform\_linux.cc

```
1 #include "platform.h"
2
3 namespace platform {
4     std::string get_platform() {
5         return "this is linux";
6     }
7 }
```



EXPLORER ...

## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers

## MEETINGCPP22-BAZEL [D...]

- ✓ BUILD
- ↳ library\_test.cc
- ↳ library.cc
- ↳ library.h
- ↳ multi\_platform.cc
- ↳ platform\_linux.cc
- ↳ platform\_windows.cc
- ↳ platform.h
- > sanitizers
- > slides

BUILD X platform\_linux.cc

platforms &gt; BUILD &gt; ...

```
18 cc_library(  
19     . . . name = "platform_windows",  
20     . . . srcs = [ "platform_windows.cc" ],  
21     . . . hdrs = [ "platform.h" ],  
22     . . . target_compatible_with = [  
23         . . . '@platforms//os:windows'  
24     . . . ]  
25 )  
26  
27 cc_library(  
28     . . . name = "platform_linux",  
29     . . . srcs = [ "platform_linux.cc" ],  
30     . . . hdrs = [ "platform.h" ],  
31     . . . target_compatible_with = [  
32         . . . '@platforms//os:linux',  
33     . . . ]  
34 )  
35  
36 cc_binary(
```



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers



## MEETINGCPP22-BAZEL [D...]

- ✓ BUILD
- ↳ library\_test.cc
- ↳ library.cc
- ↳ library.h
- ↳ multi\_platform.cc
- ↳ platform\_linux.cc
- ↳ platform\_windows.cc
- ↳ platform.h
- > sanitizers
- > slides

BUILD X ↳ platform\_linux.cc

platforms &gt; BUILD &gt; ...

```
31     target_compatible_with = [
32         '@platforms//os:linux',
33     ]
34 )
35
36 cc_binary(
37     name = "multi_platform",
38     srcs = ["multi_platform.cc"],
39     deps = select({
40         "@platforms//os:windows": [":platform_windows"],
41         "@platforms//os:linux": [":platform_linux"],
42     }),
43 )
44 |
```





EXPLORER ...

BAZEL BUILD TARGETS

- >  //artefacts
- >  //clang\_tidy
- >  //hello\_world
- >  //platforms
- >  //sanitizers
- >  //slides

MEETINGCPP22-BAZEL [D...]

-  BUILD
-  library\_test.cc
-  library.cc
-  library.h
-  multi\_platform.cc
-  platform\_linux.cc
-  platform\_windows.cc
-  platform.h
- > sanitizers
- > slides

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2  bash    

- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel run //platforms:multi\_platform  
**INFO:** Analyzed target //platforms:multi\_platform (0 packages loaded, 0 targets configured).  
**INFO:** Found 1 target...  
Target //platforms:multi\_platform up-to-date:  
    bazel-bin/platforms/multi\_platform  
**INFO:** Elapsed time: 0.232s, Critical Path: 0.00s  
**INFO:** 1 process: 1 internal.  
**INFO:** Build completed successfully, 1 total action  
**INFO:** Build completed successfully, 1 total action  
this is linux
- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# 

# Toolchains



EXPLORER ...

## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers

MEETINGCPP22-BAZEL [D...  
= .bazelrc

- .bazelrc
- .clang-tidy
- .gitignore
- BUILD
- compile\_commands.js...
- LICENSE
- MODULE.bazel
- README.md
- WORKSPACE

BUILD

## WORKSPACE

```
37
38 # [7] · WASM
39 # Patched emsdk
40 http_archive(
41     name = "emsdk",
42     sha256 = "9c96e51385250bb463f14a8b5b23a394fe0ab2cefba413567d4c97",
43     strip_prefix = "emsdk-836d2ffe008f53738fd52829be4f6945b37fe265/bazel",
44     url = "https://github.com/figma/emsdk/archive/836d2ffe008f53738fd52829be4f6945b37fe265/bazel"
45 )
46
47 load("@emsdk//:deps.bzl", "emsdk_deps" = "deps")
48 emsdk_deps()
49
50 load("@emsdk//:emscripten_deps.bzl", "emsdk_emscripten_deps" = "emscripten_deps")
51 emsdk_emscripten_deps(emscripten_version = "3.1.8")
52
53 load("@emsdk//:toolchains.bzl", "register_emscripten_toolchains")
54 register_emscripten_toolchains()
```

...

The sidebar contains several icons: a file icon, a magnifying glass, a circular arrow, a gear, and a person icon. The 'EXPLORER' tab is selected. Below it, the 'BAZEL BUILD TARGETS' section lists: //artefacts, //clang\_tidy, //hello\_world, //platforms, //sanitizers, and a collapsed section. The 'MEETINGCPP22-BAZEL [D...' section is expanded, showing: slides, structured, testing, wasm, a collapsed section, and a 'BUILD' item which is selected. Other items in this section include: main.cc (highlighted), .bazelignore, .bazeliskrc, .bazelrc, and .clang-tidy.

The main editor shows a C++ file named 'main.cc'. The code is:

```
wasm > main.cc > main()
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello World!\n";
5 }
```

EXPLORER ...

BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers
- > //---

MEETINGCPP22-BAZEL [D...]

- > slides
- > structured
- > testing
- > wasm
  - ✓ BUILD
  - main.cc
- .bazelignore
- .bazeliskrc
- .bazelrc
- .clang-tidy

BUILD platforms BUILD wasm ...

wasm > BUILD > ...

```
1 load("@emsdk//emscripten_toolchain:wasm_rules.bzl", "wasm_cc_binary")
2
3 cc_binary(
4     name = "main",
5     srcs = ["main.cc"],
6 )
7
8 wasm_cc_binary([
9     name = "main_wasm",
10    cc_target = ":main",
11 ])
```





EXPLORER

...

## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers
- > //---

## MEETINGCPP22-BAZEL [D...

- > slides
- > structured
- > testing
- ✓ wasm
  - BUILD
  - main.cc
  - .bazelignore
  - .bazeliskrc
  - .bazelrc
  - .clang-tidy

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

2

bash

+

▼

X

- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel build //wasm:main\_wasm  
**INFO:** Analyzed target //wasm:main\_wasm (0 packages loaded, 0 targets configured).  
**INFO:** Found 1 target...  
Target //wasm:main\_wasm up-to-date:  
  bazel-bin/wasm/main\_wasm/main.js  
  bazel-bin/wasm/main\_wasm/main.wasm  
  bazel-bin/wasm/main\_wasm/main.wasm.map  
  bazel-bin/wasm/main\_wasm/main.js.mem  
  bazel-bin/wasm/main\_wasm/main\_wasm/main.fetch.js  
  bazel-bin/wasm/main\_wasm/main\_wasm/main.worker.js  
  bazel-bin/wasm/main\_wasm/main\_wasm/main.data  
  bazel-bin/wasm/main\_wasm/main\_wasm/main.js.symbols  
  bazel-bin/wasm/main\_wasm/main\_wasm/main.wasm.debug.wasm  
  bazel-bin/wasm/main\_wasm/main.html  
**INFO:** Elapsed time: 0.248s, Critical Path: 0.00s  
**INFO:** 1 process: 1 internal.  
**INFO:** Build completed successfully, 1 total action
- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# nodejs bazel-bin/wasm/main\_wasm/main.js  
Hello World!
- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#

# Packages



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers



## MEETINGCPP22-BAZEL [D...

- > .devcontainer
- > .vscode
- > artefacts
- | BUILD
- | custom\_command.cc
- | description
- | file.txt



- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs



BUILD

custom\_command.cc X



artefacts &gt; custom\_command.cc &gt; main()

```
1 #include <iostream>
2 #include <fstream>
3
4 int main() {
5     std::cout << "Hello from an installed binary.\n";
6     std::fstream file("/usr/share/custom_command/file.txt");
7     if (!file.is_open()) {
8         std::cerr << "Failed to open file.\n";
9         return 1;
10    }
11    std::string line;
12    std::getline(file, line);
13    std::cout << line << "\n";
14    return 0;
15 }
```



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers



## MEETINGCPP22-BAZEL [D...]

- > .devcontainer
- > .vscode
- > artefacts
  - ✓ BUILD
  - custom\_command.cc
  - description
  - file.txt
- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs



BUILD X ⚙ custom\_command.cc

artefacts &gt; BUILD &gt; ...

```
1 load("@rules_pkg//pkg:deb.bzl", "pkg_deb")
2 load("@rules_pkg//pkg:tar.bzl", "pkg_tar")
3
4 cc_binary(
5     name = "custom_command",
6     srcs = ["custom_command.cc"]
7 )
8
9 pkg_tar(
10     name = "pkg-bin",
11     strip_prefix = "/artefacts",
12     package_dir = "/usr/bin",
13     srcs = ["/artefacts:custom_command"],
14     mode = "0755",
15 )
16
17 filegroup(
18     name = "custom_command_data",
19     srcs = [
20         "file.txt"
21     ]
22 )
```



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers
- > .....



## MEETINGCPP22-BAZEL [D...]

- > .devcontainer
- > .vscode



## artefacts

- BUILD
- custom\_command.cc
- description
- file.txt
- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs



BUILD X ⚙ custom\_command.cc

```
artefacts >  BUILD > ...
14   mode = "0755",
15 )
16
17 filegroup(
18   name = "custom_command_data",
19   srcs = [
20     "file.txt",
21   ]
22 )
23
24 pkg_tar(
25   name = "pkg-data",
26   strip_prefix = "/artefacts",
27   package_dir = "/usr/share/custom_command",
28   srcs = ["/artefacts:custom_command_data"],
29   mode = "0644",
30 )
31
32 pkg_tar(
33   name = "debian-data",
```



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers



## MEETINGCPP22-BAZEL [D...]

- > .devcontainer
- > .vscode
- > artefacts
  - ✓ BUILD
  - custom\_command.cc
  - description
  - file.txt
- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs



BUILD X ⚙ custom\_command.cc

artefacts &gt; BUILD &gt; ...

```
32     pkg_tar(  
33         name = "debian-data",  
34         extension = "tar.gz",  
35         deps = [  
36             ":pkg-bin",  
37             ":pkg-data",  
38         ],  
39     )  
  
41     pkg_deb(  
42         name = "customcommand-debian",  
43         architecture = "amd64",  
44         data = ":debian-data",  
45         description_file = "description",  
46         homepage = "https://simontoth.eu",  
47         maintainer = "Simon Toth (business@simontoth.eu)",  
48         package = "customcommand",  
49         version = "1.0.0",  
50     )
```



EXPLORER ...

## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers
- > .....

## MEETINGCPP22-BAZEL [D...]

- > .devcontainer
- > .vscode
- > artefacts
  - ✓ BUILD
  - ⌚ custom\_command.cc
  - ≡ description
  - ≡ file.txt
- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS 2

bash +



```
● root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# bazel build //artefacts:customcommand-debian
INFO: Analyzed target //artefacts:customcommand-debian (0 packages loaded, 0 targets configured).
INFO: Found 1 target...
Target //artefacts:customcommand-debian up-to-date:
  bazel-bin/artefacts/customcommand_1.0.0_amd64.deb
INFO: Elapsed time: 0.237s, Critical Path: 0.00s
INFO: 1 process: 1 internal.
INFO: Build completed successfully, 1 total action
○ root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#
```



EXPLORER ...



## BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers



## MEETINGCPP22-BAZEL [D...]

- > .devcontainer
- > .vscode
- > artefacts
  - BUILD
  - custom\_command.cc
  - description
  - file.txt
- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2

bash +

```
● root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# dpkg -i bazel-bin/artefacts/customcommand_1.0.0_amd64.deb
Selecting previously unselected package customcommand.
(Reading database ... 27002 files and directories currently installed.)
Preparing to unpack .../customcommand_1.0.0_amd64.deb ...
Unpacking customcommand (1.0.0) ...
Setting up customcommand (1.0.0) ...
○ root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#
```

EXPLORER    ...

BAZEL BUILD TARGETS

- > //artefacts
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > //sanitizers
- > //...

MEETINGCPP22-BAZEL [D...]

- > .devcontainer
- > .vscode
- > artefacts
  - BUILD
  - custom\_command.cc
  - description
  - file.txt
- > bazel-bin
- > bazel-meetingcp...
- > bazel-out
- > bazel-testlogs

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS 2    bash +

- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel# /usr/bin/custom\_command  
Hello from an installed binary.  
Some content...
- root@c3835cd10ffa:/workspaces/meetingcpp22-bazel#

# GitHub



EXPLORER ...

BAZEL BUILD TARGETS

- > //artefacts
- > //cache
- > //clang\_tidy
- > //hello\_world
- > //platforms
- > .....

MEETINGCPP22-BAZEL [D...]

- > .devcontainer
- > .github/workflows
- | ! babel.yml
- > .vscode ●
- > artefacts
- > bazel-bin ↗
- > bazel-meetingcp... ↗
- > bazel-out ↗
- > bazel-testlogs ↗
- > cache
- > clang\_tidy

## ! babel.yml X

```
.github > workflows > ! babel.yml
1   name: Bazel
2
3   on:
4     workflow_dispatch:
5     push:
6
7   jobs:
8     build:
9       runs-on: ubuntu-latest
10
11    steps:
12      - uses: actions/checkout@v3
13
14      - name: Mount bazel cache
15        uses: actions/cache@v3
16        with:
17          path: "/home/runner/.cache/bazel"
18          key: bazel-${{ github.run_id }}
19          restore-keys: |
20            bazel
```



EXPLORER ...

## BAZEL BUILD TARGETS

- > `//artefacts`
- > `//cache`
- > `//clang_tidy`
- > `//hello_world`
- > `//platforms`
- > `//...`

## MEETINGCPP22-BAZEL [D...]

- > `.devcontainer`
- > `.github/workflows`
  - ! `bazel.yml`**
  - > `.vscode` ●
- > `artefacts`
- > `bazel-bin`
- > `bazel-meetingcp...`
- > `bazel-out`
- > `bazel-testlogs`
- > `cache`
- > `clang_tidy`

! `bazel.yml` X`.github > workflows > ! bazel.yml`

```
21      . - name: Install LLVM and Clang
22      .     run: |
23      .       wget https://apt.llvm.org/llvm.sh
24      .       chmod +x llvm.sh
25      .       sudo ./llvm.sh 16 all
26
27
28      . - name: Install Bazel
29      .     run: |
30      .       curl -LO "https://github.com/bazelbuild/bazelisk/releases/d...
31      .       mkdir -p "${GITHUB_WORKSPACE}/bin/"
32      .       mv bazelisk-linux-amd64 "${GITHUB_WORKSPACE}/bin/bazel"
33      .       chmod +x "${GITHUB_WORKSPACE}/bin/bazel"
34
35      . - name: Test
36      .     run: |
37      .       "${GITHUB_WORKSPACE}/bin/bazel" test --test_output=errors //...
38
39      . - name: Test asan
40      .     run: |
```



EXPLORER ...

## BAZEL BUILD TARGETS

- > `//artefacts`
- > `//cache`
- > `//clang_tidy`
- > `//hello_world`
- > `//platforms`
- > `//.../...`

## MEETINGCPP22-BAZEL [D...

- > `.devcontainer`
- > `.github/workflows`
  - | ! `bazel.yml`
  - | > `.vscode` ●
  - | > `artefacts`
  - | > `bazel-bin` ↴
  - | > `bazel-meetingcp...` ↴
  - | > `bazel-out` ↴
  - | > `bazel-testlogs` ↴
  - | > `cache`
  - | > `clang_tidy`

! `bazel.yml` X

```
.github > workflows > ! bazel.yml
24   ... wget https://apt.llvm.org/llvm.sh
25   ... chmod +x llvm.sh
26   ... sudo ./llvm.sh 16 all
27
28   - name: Install Bazel
29     run: |
30       curl -LO "https://github.com/bazelbuild/bazelisk/releases/d...
31       mkdir -p "${GITHUB_WORKSPACE}/bin/"
32       mv bazelisk-linux-amd64 "${GITHUB_WORKSPACE}/bin/bazel"
33       chmod +x "${GITHUB_WORKSPACE}/bin/bazel"
34
35   - name: Test
36     run: |
37       "${GITHUB_WORKSPACE}/bin/bazel" test --test_output=errors ...
38
39   - name: Test asan
40     run: |
41       "${GITHUB_WORKSPACE}/bin/bazel" build --test_output=errors ...
```

[Summary](#)

Jobs

[build](#)[Run details](#)[Usage](#)[Workflow file](#)[build](#)

succeeded 13 minutes ago in 1m 59s

 Search logs[Test](#)

9s

```
8 Analyzing: 2 targets (1 packages loaded, 0 targets configured)
9 Analyzing: 2 targets (5 packages loaded, 6 targets configured)
10 Analyzing: 2 targets (59 packages loaded, 243 targets configured)
11 Analyzing: 2 targets (72 packages loaded, 1015 targets configured)
12 INFO: Analyzed 2 targets (77 packages loaded, 1078 targets configured).
13 INFO: Found 2 test targets...
14 [0 / 4] [Prepa] BazelWorkspaceStatusAction stable-status.txt
15 INFO: Elapsed time: 9.150s, Critical Path: 0.11s
16 INFO: 1 process: 1 internal.
17 INFO: Build completed successfully, 1 total action
18 //cache:cache_test                                     (cached) PASSED in 0.0s
19 //cache:with_deps_test                                (cached) PASSED in 0.0s
20
21 Executed 0 out of 2 tests: 2 tests pass.
22 INFO: Build completed successfully, 1 total action
```

> [Test asan](#)

1s

> [Post Mount bazel cache](#)

23s

> [Post Run actions/checkout@v3](#)

0s

> [Complete job](#)

0s

# Remote Cache

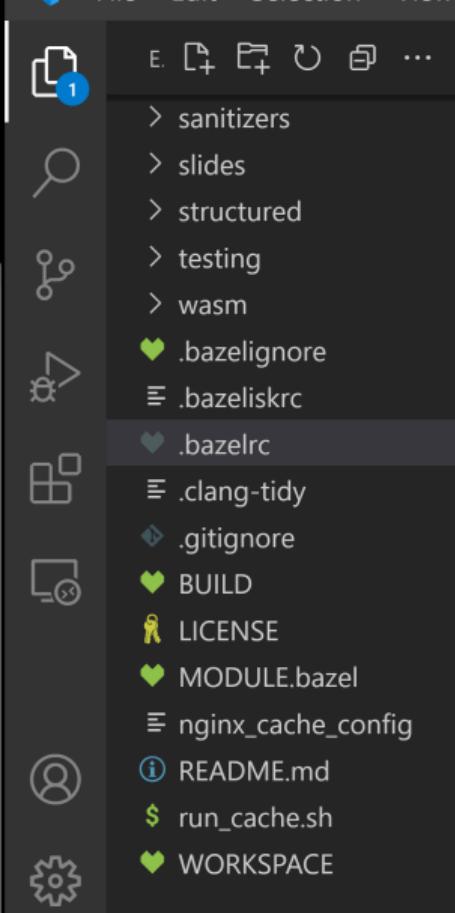


- E      ...
- > .devcontainer
- > .github
- > .vscode
- > artefacts
- > bazel-bin 
- > bazel-meetingcp... 
- > bazel-out 
- > bazel-testlogs 
- > cache
- > clang\_tidy
- > external 
- > hello\_world
- > platforms
- > sanitizers
- > slides
- > structured
- > testing
- > wasm



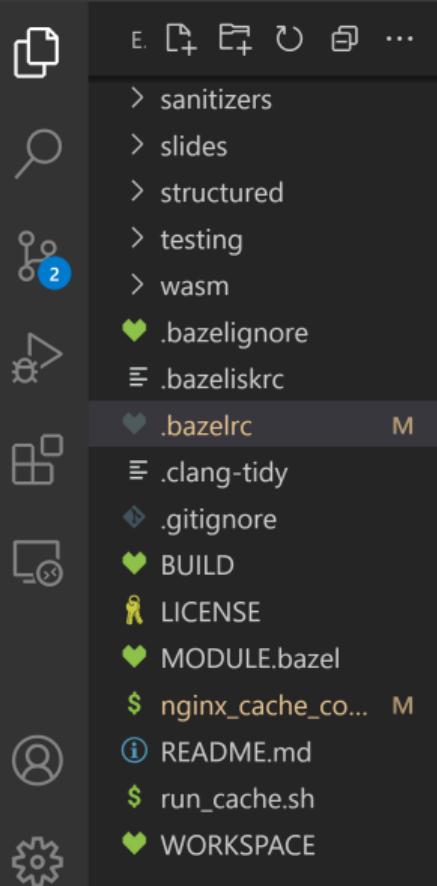
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1  bash    X

root@b13df619b53f:/workspaces/meetingcpp22-bazel# ./run\_cache.sh



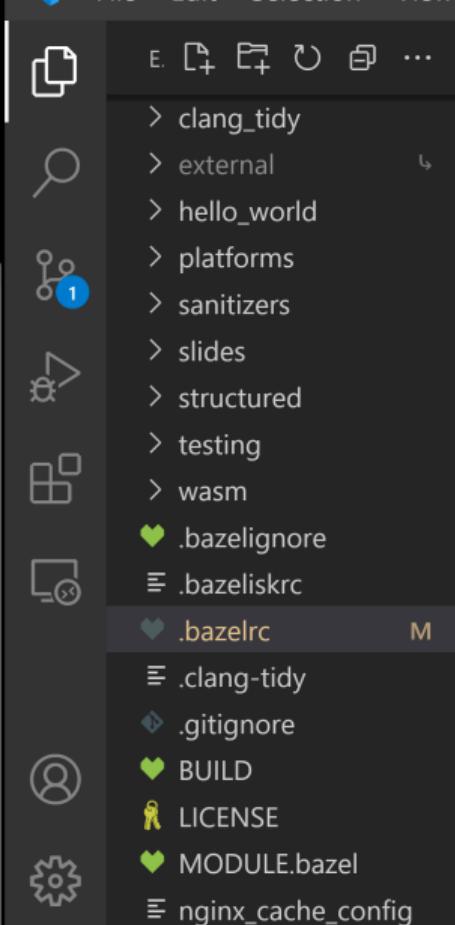
## .bazelrc

```
22  # Optionally override the clang-tidy config file target
23  build:clang-tidy --@bazel_clang_tidy//:clang_tidy_config //:clang_tidy
24
25  # [6] Platform support
26  build --incompatible_enable_cc_toolchain_resolution
27
28  # [7] WASM
29  build:wasm --platforms=@emsdk//:platform_wasm
30
31  # [9] Remote Cache
32  build --remote_cache=http://localhost:80/cache/
33  # build --remote_upload_local_results=false
34
```



```
● root@cbdd97d8ceb1:/workspaces/meetingcpp22-bazel# bazel test //cache/...
2022/11/07 10:11:53 Downloading https://releases.bazel.build/6.0.0/rolling/6.0
.0-pre.20220922.1/bazel-6.0.0-pre.20220922.1-linux-x86_64...
Extracting Bazel installation...
Starting local Bazel server and connecting to it...
INFO: Invocation ID: 789241b4-1cbd-43af-b9cc-a412c7c1c986
INFO: Analyzed 2 targets (77 packages loaded, 1060 targets configured).
INFO: Found 2 test targets...
INFO: Elapsed time: 36.435s, Critical Path: 2.90s
INFO: 38 processes: 15 internal, 23 processwrapper-sandbox.
INFO: Build completed successfully, 38 total actions
//cache:cache_test  PASSE
D in 0.0s
//cache:with_deps_test   PASSE
D in 0.0s

INFO: Build completed successfully, 38 total actions
○ root@cbdd97d8ceb1:/workspaces/meetingcpp22-bazel#
```



## ♥ .bazelrc M X

```
♥ .bazelrc
22 # Optionally override the .clang-tidy config file target
23 build:clang-tidy --@bazel_clang_tidy//:clang_tidy_config //:clang_tidy
24
25 # [6] Platform support
26 build --incompatible_enable_cc_toolchain_resolution
27
28 # [7] WASM
29 build:wasm --platforms=@emsdk//:platform_wasm
30
31 # [9] Remote Cache
32 build --remote_cache=http://localhost:80/cache/
33 build --remote_upload_local_results=false
34
```



E. ...

## PROBLEMS

## OUTPUT DEBT

## TERMINAL

PORTS

> ba

+ ✓

三

1

- |                       |                |
|-----------------------|----------------|
| > clang_tidy          |                |
| > external            | ↳              |
| > hello_world         |                |
| > platforms           |                |
| > sanitizers          |                |
| > slides              |                |
| > structured          |                |
| > testing             |                |
| > wasm                |                |
| ● .bazelignore        |                |
| ≡ .bazeliskrc         |                |
| ● .bazelrc M          | (cached) PASSE |
| ≡ .clang-tidy         |                |
| ◆ .gitignore          |                |
| ● BUILD               | (cached) PASSE |
| ● LICENSE             |                |
| ● MODULE.bazel        |                |
| § nginx cache co... M |                |

# Final remarks

# Windows support?

<https://bazel.build/configure/windows>

Bleeding edge features.

# 3rd party modules?

# C++20 modules?

[https://github.com/rnburn/rules\\_cc\\_module](https://github.com/rnburn/rules_cc_module)

[https://github.com/eomii/rules\\_ll](https://github.com/eomii/rules_ll)

**rules\_ll got an update as I was editing this recording a should work!**

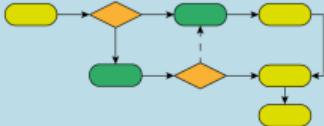
# Dependencies without Bazel support?

[https://github.com/bazelbuild/rules\\_foreign\\_cc](https://github.com/bazelbuild/rules_foreign_cc)

# Integration tests?

# Living comfortably at HEAD with Bazel

# A Complete Guide to Standard C++ Algorithms



RNDr. Šimon Tóth

- ▶ [HappyCerberus/book-cpp-algorithms](#)
- ▶ [leanpub.com/cpp-algorithms-guide](#)
- ▶ [twitter.com/SimonToth83](#)
- ▶ [linkedin.com/in/simontoth](#)

Simon Tóth @SimonToth83 · Oct 29

When using designated initializers, it might be desirable to ensure that some fields are always explicitly initialized. We can use a simple wrapper that cannot be constructed from an empty initializer list to enforce initialization. #cpp #coding #learnprogramming

```
#include <string>

template <typename T>
struct RequiredField {
    RequiredField(const T& value) : value(value) {}
    operator T() const { return value; }
    operator const T() const { return value; }
    T value;
};

struct Options {
    RequiredField<int> id;
    std::string label = "default label";
};

Options options = { .id = 20 };
// options.id == 20, options.label == "default label"
// Options opt = { .label = "some label" }; // Won't compile,
// RequiredField is not constructible from empty initializer list.
```