The LLVM coding standard and clang-tidy

Jonas Nilsson

The LLVM coding standards

http://llvm.org/docs/CodingStandards.html

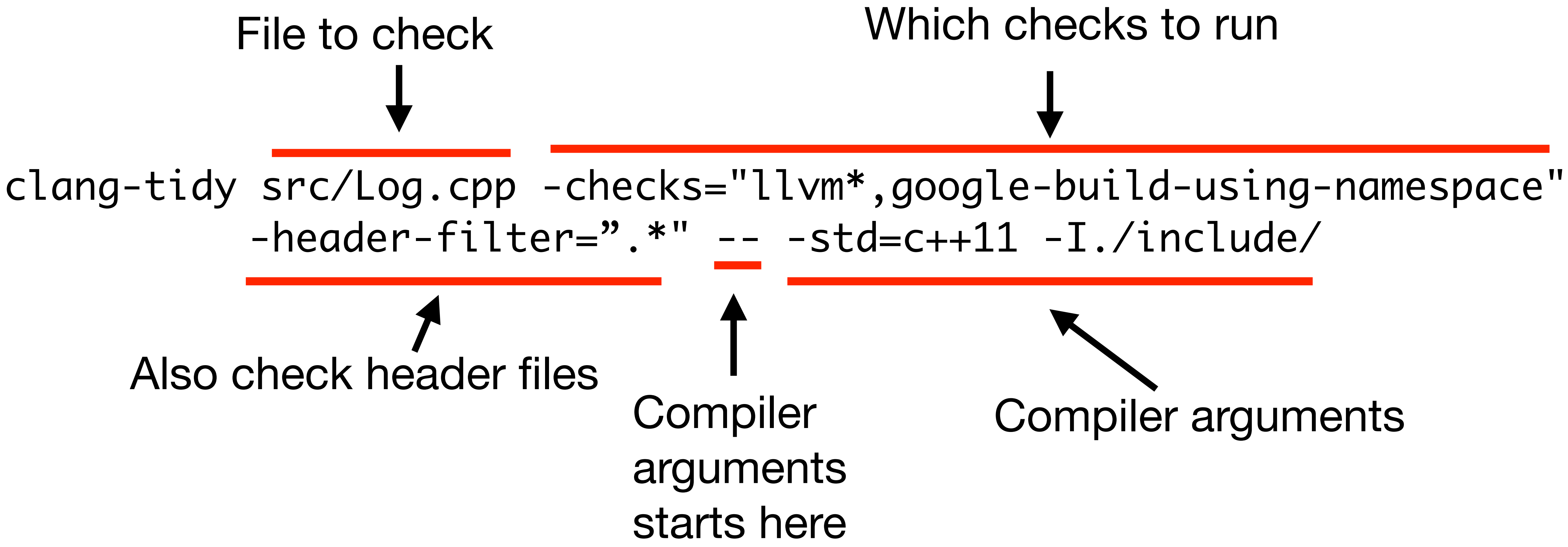
Summary

- Use spaces
- Treat warnings as errors
- Ignore the ban on RTTI and exceptions
- Avoid static constructors
- Use **auto** when appropriate
- Use camel case
- Functions start with a lower case letter
- Types and variables start with an upper case letter
- Do not use ”**using namespace xxx**”
- Ignore the ban on **iostream**
- Put space before parenthesis in flow statement
- Do not put space before parenthesis in function call
- Use file suffixes **.cpp** *and* ***.h***

clang-tidy

- Does not check for everything
- Use in conjunction with clang-format

Usage



File to check

Which checks to run

```
clang-tidy src/Log.cpp -checks="llvm*,google-build-using-namespace"
           -header-filter=”.*" -- -std=c++11 -I./include/
```

Also check header files

Compiler arguments starts here

Compiler arguments

run-clang-tidy.py

1.

```
cmake -DCMAKE_EXPORT_COMPILE_COMMANDS=ON ...
```

compile_commands.json is created

2.

```
run-clang-tidy.py -header-filter='.*'
-checks=’llvm*,google-build-using-namespace'
```

Alt.

```
run-clang-tidy.py -header-filter='.*'
-checks=’llvm*,google-build-using-namespace’ -fix
```