CodeSonar Training

Scan Mixed Languages

May 2025

Agenda

- Why Does CodeSonar Scan the Code?
- How Does CodeSonar Scan Mixed Language Code?
- CodeSonar and Mixed Language Projects
- Building the CodeSonar Command Line
- What to Expect
- Required Third Party Tools

Why Does CodeSonar Scan the Code?

- To see all the components in your project
- To learn how each component is constructed
- To provide a single view into the parts of the project

How does CodeSonar Scan Mixed Language Projects?

- For each language in the project scan the files in that language
  - Use codesonar build to compile the different scans into one project
  - Use codesonar analyze to analyze all the components once compiled
- The scan uses the appropriate tools to scan each component
- Each analysis is only aware of the scan in it's own language
- Separate models are passed for each language

CodeSonar and Multiple Languages

- Use codesonar build with the correct scan command for each language
- For N languages, that will be N codesonar build commands
- Each codesonar build uses the same project and prj_files information
- Don't use –clean for any but possibly the first
- This builds up scan information from each language
- Finally run codesonar analyze
- Still the same project and prj_files information
- No –clean, and no scan information
- All the languages will be analyzed and reported in a single project

Building the CodeSonar Command Lines

- There will be more than one command line
- For N languages, there will be N+1 total command lines
- One codesonar build command line for each language
- Each codesonar build is done following the rules for that language
- Finally one codesonar analyze command line with no scanning instruction
- All have the same options, project name and prj_files location

Building the CodeSonar Command Lines (2)

- Start from simple
  - Call the project mixed_app
  - It contains Rust, and C++
  - A subdirectory called rust has cargo.toml
  - A subdirectory called cpp has the C++ makefile
  - Use the installation default Hub URL
  - Scan each then analyze

```
codesonar build mixed_app codesonar rust_scan.py rust
codesonar build mixed_app cpp/make all
codesonar analyze mixed_app
```

Building the CodeSonar Command Line (3)

- Hide both scan commands in a script
  - This can make it easier to read
  - Can be any scripting language
- In the script cs_scan.sh put

```
codesonar rust_scan.py rust
make all
```

- Make the script executable

```
codesonar analyze mixed_app sh cs_scan.sh
```

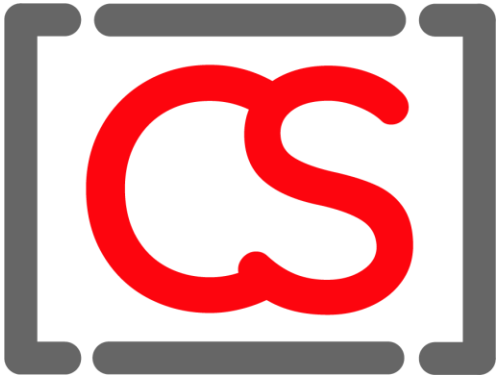
What to Expect

- Command line feedback from each command as the scan runs
- When the analyze command starts a link to the analysis in the Hub
- Typical analysis results with warnings and relevant code

Required Third Party Tools

- Appropriate tools for each component in the mixed language project
- For example, if the project is Kotlin and Go, you will need the third party tools for Kotlin and Go

Thank You



Contact Info

CodeSecure

Customer Support

support@codesecure.com

https://support.codesecure.com

More Info

www.codesecure.com

Shift Left Academy