

# SonarCFamily Building on Linux

이 문서는 Linux 환경에서 SonarQube Scanner를 이용하여 C 빌드를 수행하는 방법에 대하여 가이드한다.

## 1. SonarQube Scanner 설치

다음 링크를 통해 SonarQube Scanner를 설치한다.

- <https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner>

## 2. 빌드 환경 준비

C/C++ 프로젝트는 SonarQube Build Wrapper가 필요하다. Build Wrapper는 C/C++ 프로젝트로부터 정보를 수정하는 역할을 하며, 빌드에 영향을 주지 않는다.

### 2.1 Build Wrapper 다운로드

다음 경로를 통해 Build Wrapper를 다운로드 할 수 있다.

- <http://소나큐브주소/static/cpp/build-wrapper-linux-x86.zip>

Build Wrapper 다운로드 링크

- <https://sonarsource.bintray.com/CommercialDistribution/sonar-cpp-plugin/>

```
wget https://sonarsource.bintray.com/CommercialDistribution/sonar-cpp-plugin/build-wrapper-3.11.zip
```

### 2.2 압축 해제

다운로드 받은 Build Wrapper를 적절한 폴더로 압축 해제한다.

```
unzip build-wrapper-3.11.zip
```

압축 해제 후 다음과 같이 구성된다.

- linux-x86-32
- linux-x86-64
  - build-wrapper-linux-x86-65
- macosx-x86
- win-x86-32
- win-x86-64

자신의 환경에 맞는 build-wrapper를 선택이 필요하다.

### 2.3 PATH에 해당 경로 등록

PATH에 해당 경로를 등록하거나 빌드 시 전체 경로를 이용하여 수행할 수 있다.

## 3. Build Wrapper를 통한 빌드 및 분석

### 3.1 Sample 파일 다운로드

만약 C/C++ 샘플 파일은 다음 Github에서 다운로드 받을 수 있다.

- <https://github.com/SonarSource/sonar-scanning-examples>

## 3.2 sonar-project.properties 수정

C/C++ 프로젝트의 루트에 sonar-project.properties 파일을 만들고 적당한 값을 입력한다.

아래 Sample 파일을 참고한다.

```
sonar.projectKey=org.sonarqube:cpp-build-wrapper
sonar.projectName=Example of C/C++ Scan (with BuildWrapper)
sonar.projectVersion=1.0

sonar.sources=.
#sonar.sources=src

sonar.language=cpp

# The build-wrapper output dir
sonar.cfamily.build-wrapper-output=bw-outputs

# Encoding of the source files
sonar.sourceEncoding=UTF-8

# SonarQube location
sonar.host.url=http://10.0.1.2

# Key
# sonar.login=cbcf225a4f5015201228c0fdf9

# ID/PW
sonar.login = admin
sonar.password = admin
```

## 3.3 빌드 및 분석 수행

다음과 같이 build-wrapper-linux-x86-64를 통해 빌드를 수행한다. 먼저 Build-wrapper를 통해 Build 수행하고 sonar-scanner 수행한다.

```
build-wrapper-linux-x86-64 --out-dir bw_output make clean all
sonar-scanner
```

만약 상단의 예제 파일을 이용하였다면, 다음과 같이 구성할 수 있다.

```
/build-wrapper-linux-x86-64 --out-dir bw_output ./build.sh
sonar-scanner
```

분석 완료 후 bw-outputs 폴더에 다음 파일이 생성된다.

- build-wrapper-dump.json
- build-wrapper.log.

## 4. Bamboo에서 빌드 플랜 구성

## 4.1 Source Code Checkout

저장소로부터 소스 코드를 Checkout하기 위한 Task 설정한다.

3 agents have the capabilities to run this job

**Source Code Checkout**
✕

Checkout Default Repository

Script
 ✕

build

Sonar Scanner
 ✕

**Final tasks** Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

### Source Code Checkout configuration How to use the Source Code Checkout task

Task description

Checkout Default Repository

☐ Disable this task

You can check out one or more repositories with this Task. You can choose to check out the Plan's *Default Repository* or specify a *Specific Repository*. You can add additional repositories to this Plan via the [Plan configuration](#).

Repository\*

sonarqube-scanner-build-wrapper-l ▾

Default always points to Plans default repository.

Checkout Directory

(Optional) Specify an alternative sub-directory to which the code will be checked out.

☐ Force Clean Build

Removes the source directory and checks it out again prior to each build. This may significantly increase build times.

[+ Add repository](#)

## 4.2 빌드 수행

빌드를 위한 Task를 구성한다. 아래와 같이 구성할 수 있다.

```
/data/sonarqube/build-wrapper-linux-x86/build-wrapper-linux-x86-64 --out-dir bw-outputs ./build.sh
```

Source Code Checkout  
Checkout Default Repository

**Script**  
build

Sonar Scanner

**Final tasks** Are always executed even if a previous task fails  
*Drag tasks here to make them final*

Add task

3 agents have the capabilities to run this job

### Script configuration

How to use the Script task

Task description  
build

☐ Disable this task

Interpreter  
Shell

An interpreter is chosen based on the shebang line of your script.

Script location  
Inline

Script body\*  

```
1 #!/bin/bash
2
3 /data/sonarqube/build-wrapper-linux-x86/build-wrapper-linux-x86-64 --c
```

## 4.3 Sonar Scanner 수행

기본 Sonar Scanner 수행한다.