

Jenkins 가이드

소프트웨어공학센터
SW공학기술팀

Jenkins 개요

소개	젠킨스 (Jenkins)는 지속 빌드 및 배포 공개 소프트웨어 시스템이다. 소프트웨어 프로젝트를 지속적으로 빌드 및 테스트 하도록 해주는 자동화 도구이며, 배포에 필요한 프로세스를 통합하여 제공한다		
주요 기능	빌드 및 테스트 자동화 (Continuous build , Continuous test , Automation)		
카테고리(용도)	Continuous Integration	세부 카테고리	-
커버리지	Continuous Integration	도구 난이도	상급
License 형태 / 비용	MIT License / 무료	사전 설치 도구	-
사용 환경 / 개발 상태	운영체제	Multi-platform	
	Eclipse 환경	Multi-platform	
특징	<ul style="list-style-type: none">• 빌드 프로젝트의 구성, 관리• 프로젝트의 구축, 테스트, 배포 및 자동화를 지원하는 다양한 플러그인들과의 연동 제공• Linux , Windows OS 지원• war 파일로 간단하게 배포 및 간편한 Web GUI 제공		
적용 회사 / 프로젝트	-		
관련 도구	Subversion , JUnit , Ant, PMD		
제작사	-		
공식 홈페이지	https://jenkins-ci.org/		
개발자	-		

목차

□ Jenkins 가이드

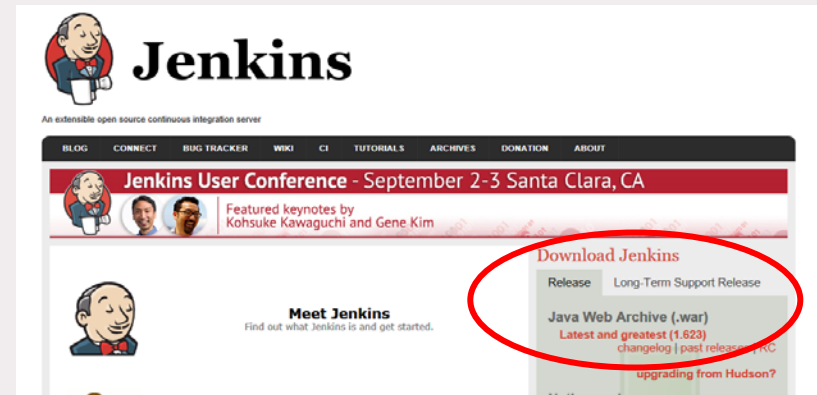
- 설치
- 기본 설정
- 플러그인 설정
 - gcov , cppcheck , xUnit
 - NSIQ , Sonarqube

Jenkins 설치 - 1

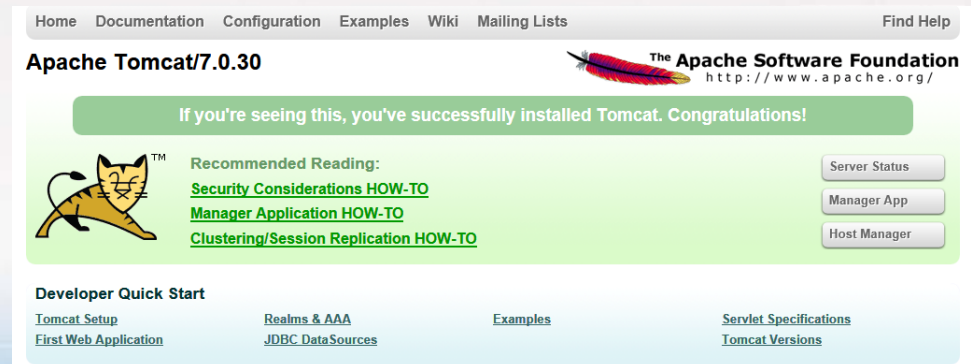
- Jenkins 의 설치 는 일반적인 WAS (WebShere Application Server) 솔루션과 동일한 방식

1) Jenkins 최신 버전 다운로드

(사이트 주소 : <https://jenkins-ci.org/> 파일명 : jenkins.war)

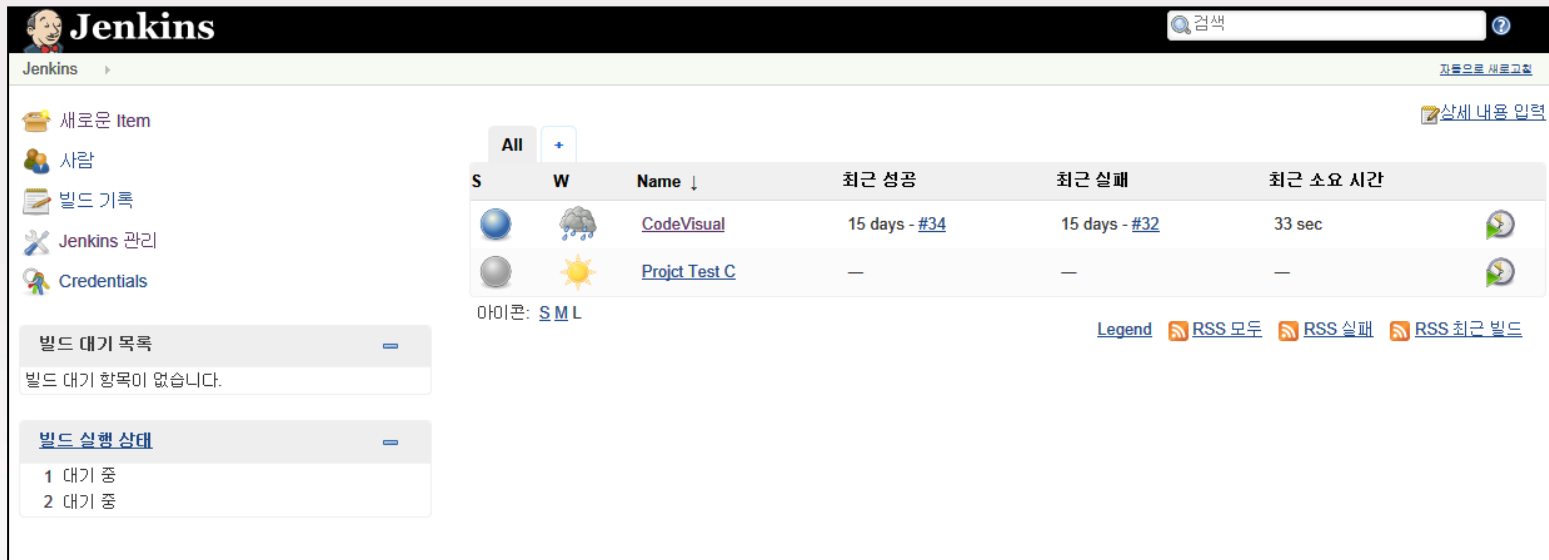


2) WAS 서버 (tomcat , jeus , weblogic) 의 구축



Jenkins 설치 - 2

- 3) WAS 서버에 jenkins 어플리케이션을 추가
(tomcat 의 경우 : tomcat 설치 경로의 webapps 폴더로 jenkins.war 를 복사
- 4) WAS 서버 재시작 후 jenkins 동작 확인
(tomcat 의 경우 기본 URL 은 <http://localhost:8080/jenkins/>)



The screenshot shows the Jenkins web interface. On the left is a sidebar with navigation links: '새로운 Item', '사람', '빌드 기록', 'Jenkins 관리', and 'Credentials'. Below these are sections for '빌드 대기 목록' (empty) and '빌드 실행 상태' (showing 1 waiting and 2 running). The main area displays a table of build jobs. The table has columns for status (S), weather icon (W), name, recent success, recent failure, and duration. Two jobs are listed: 'CodeVisual' and 'Project Test C'. Below the table are links for '아이콘: S M L' and a legend for RSS feeds.

S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
		CodeVisual	15 days - #34	15 days - #32	33 sec
		Project Test C	—	—	—

아이콘: [S](#) [M](#) [L](#)

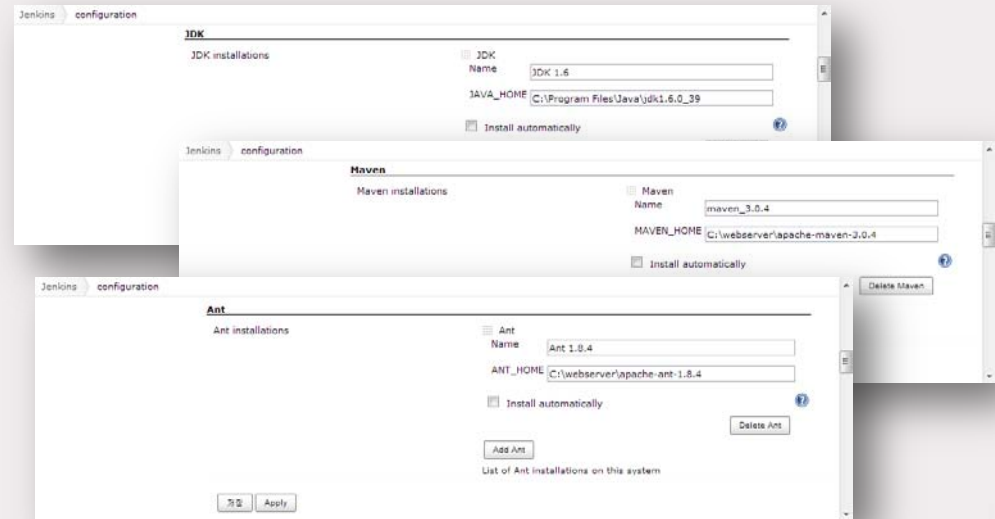
Legend RSS 모두 RSS 실패 RSS 최근 빌드

<동작 화면>

Jenkins 기본 설정 (Jenkins 관리)

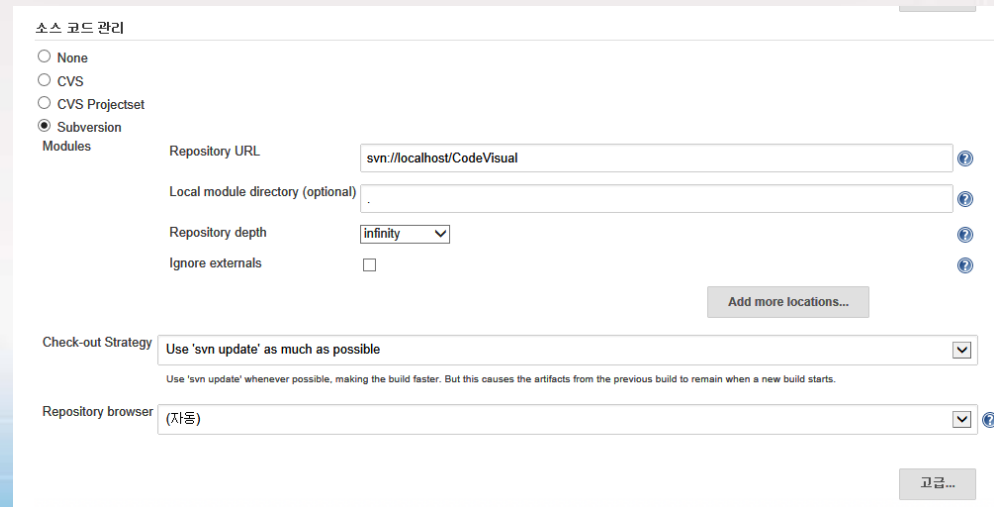
1. JDK , ANT , MAVEN 설정

- Jenkins 관리 메뉴에서 설치된 버전 및 경로 설정
- 도구에 따라 세부 설정 추가



2. Subversion 설정

- Jenkins 관리 메뉴에서 설치 버전 설정
- 프로젝트 메뉴에서 URL 및 세부 정보 설정
(※ 보안 관련 이슈는 경고메시지 링크창을 통해 사용자 / 암호 입력)



Jenkins 기본 설정 (프로젝트 관리)

1. 프로젝트 생성

- Jenkins 메인 화면에서 좌측 새로운 item 클릭
- 사용자에게 적합한 프로젝트 유형 선택
- (기본 설정으로 Freestyle project 권장)



2. Build 설정

- 프로젝트가 생성 후 프로젝트 메뉴에서 선택
- 자동 빌드 타임 간격 입력



<빌드 타임 입력 방식>

분	시	일	월	요일
0-59	0-23	1-31	1-12	0-6 (0=일요일)

<예시>

20	2	*	*	6	매주 토요일 02:20
0	4-6	*	*	*	매일 4,5,6시
5	*/2	*	*	*	매일 2시간 간격으로 5분에
15	1	1	*	*	매월 1일 01:15
30	0	1	1,7	*	1,7월 1일 00:30

Jenkins 기본 설정 (프로젝트 빌드)

1. 프로젝트 운영 화면

Jenkins

Jenkins SWV1 ENABLE AUTO REFRESH

[대시보드로 가기](#)
[상태](#)
[변경 사항](#)
[작업 공간](#)
[Build Now](#)
[삭제 Project](#)
[설정](#)
[Modules](#)
[Javadoc](#)
[Sonar](#)

Project SWV1
Language : Java
Build : Maven
Test : Sonar

[내용 수정](#)
[프로젝트 중지하기](#)

[Javadoc](#)
[Sonar](#)
[Workspace](#)
[Recent Changes](#)

주요 링크

- [Last build, \(#8\), 10 days 전](#)
- [Last stable build, \(#8\), 10 days 전](#)
- [Last successful build, \(#8\), 10 days 전](#)
- [Last failed build, \(#3\), 3 mo 18 days 전](#)
- [Last unsuccessful build, \(#5\), 2 mo 5 days 전](#)

Build History (동기)

#	Build Time
#8	2013. 10. 22 오후 2:27:10
#7	2013. 10. 10 오후 5:09:19
#6	2013. 8. 28 오전 11:23:34
#5	2013. 8. 28 오전 11:21:50
#4	2013. 7. 18 오후 1:19:06
#3	2013. 7. 16 오후 6:43:57
#2	2013. 4. 9 오후 2:23:42
#1	2013. 4. 9 오후 2:22:28

[RSS 모든 것에 대해](#) [RSS 실패에 대해](#)

[이 페이지 한글화 도와주기](#)

작성된 페이지: 2013. 11. 1 오후 7:50:14 [REST API](#) [Jenkins ver. 1.502](#)

1. GCOV (Cobertura)

- 품질 지표

소스 코드의 line , class , condition , file 에 대한 **count** 및 **coverage** 를 측정 및 계산

실제 소스 코드가 얼마나 실행되었는지(coverage)와, 구문(statement), 조건(condition), 결과(decision) 구조의 완결성

- plugin 설정

* 파일 경로 설정

* 빌드 상태 설정 (health , stability , target)

* 화면 display 설정 (zoom , encoding)

- 설치 및 동작

Publish Cobertura Coverage Report
Cobertura xml report pattern:
This is a file name pattern that can be used to locate the cobertura xml report files (for example with Maven2 use `**/target/site/cobertura/coverage.xml`). The path is relative to the module root unless you have configured your SCM with multiple modules, in which case it is relative to the workspace root. Note that the module root is SCM-specific, and may not be the same as the workspace root. Cobertura must be configured to generate XML reports for this plugin to function.

Consider only stable builds: ☐
Include only stable builds, i.e. exclude unstable and failed ones.

Fail builds if no reports: ☒
fail builds if No coverage reports are found.

Fail unhealthy builds: ☐
Unhealthy projects will be failed.

Fail unstable builds: ☐
Unstable projects will be failed.

Health auto update: ☐
Auto update threshold for health on successful build.

Stability auto update: ☐
Auto update threshold for stability on successful build.

Zoom coverage chart: ☐
Zoom the coverage chart and crop area below the minimum and above the maximum coverage of the past reports.

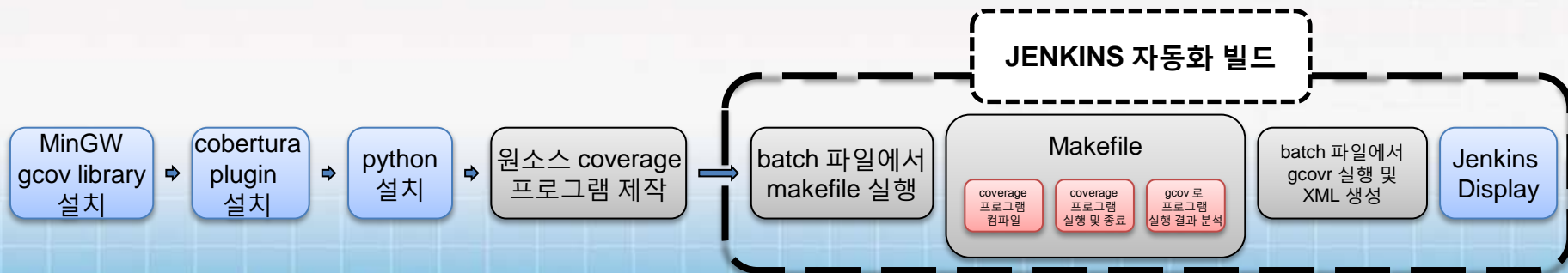
Source Encoding:
Encoding when showing files.

Coverage Metric Targets

Metric	Threshold	Health	Stability	Target
Conditionals	70.0	0.0	0.0	
Lines	80.0	0.0	0.0	
Methods	80.0	0.0	0.0	

Add
Configure health reporting thresholds.
For the row, leave blank to use the default value (i.e. 80).
For the and rows, leave blank to use the default values (i.e. 0).

Delete



1. CPP Check

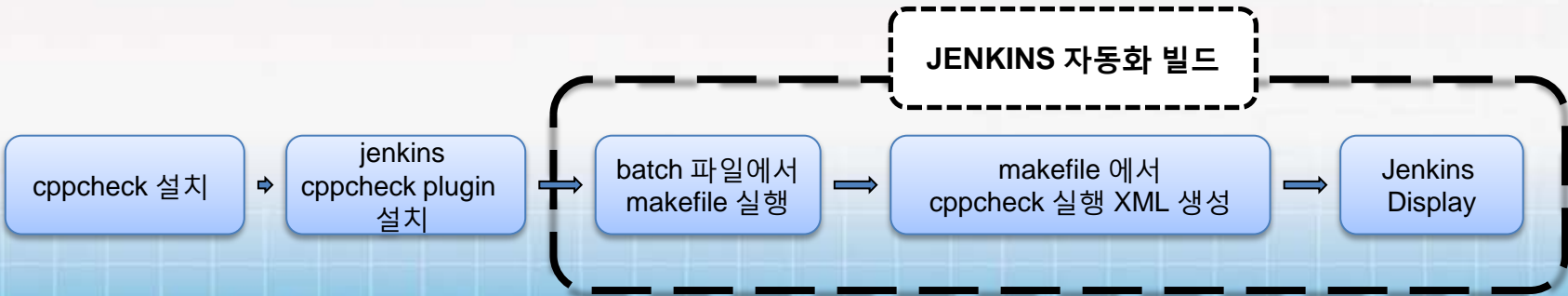
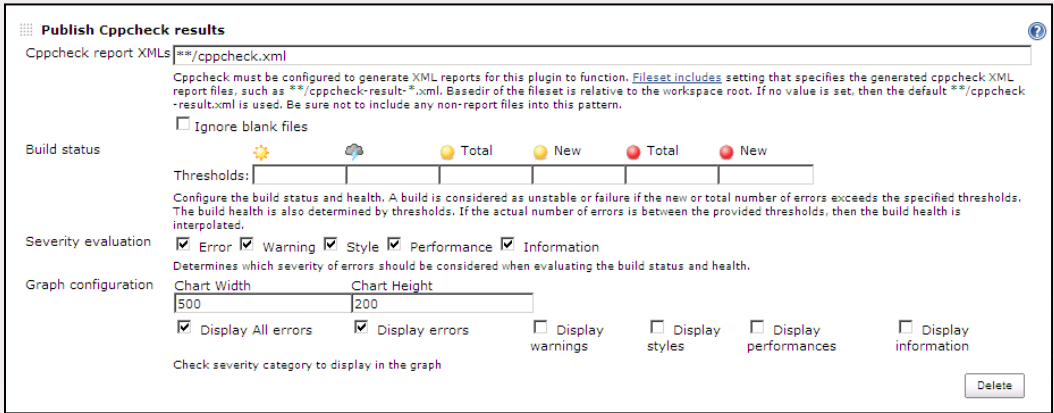
- 품질 지표

소스 코드를 정적분석하여, **엄격함** (Severity) 수준을 5단계 (6종류) 로 구분하여 check 해냄
(**ERROR / WARNING / STYLE / PERFORMANCE / INFORMATION / NO CATEGORY**)
180 rules 기본 제공 + customization rule 기능 제공

- plugin 설정

- * 파일 경로 설정
- * 룰 위반 경고 수준(thresholds) 설정
- * 화면 display 설정

- 설치 및 동작



Jenkins 플러그인 설정

1. CPP UNIT (xUnit)

- 품질 지표

단위 테스트를 통하여 소스 코드의 완성도 체크

테스트 케이스의 **fail count / percentage** 를 이용하여 현재의 기능 구현 및 빌드 상태를 확인

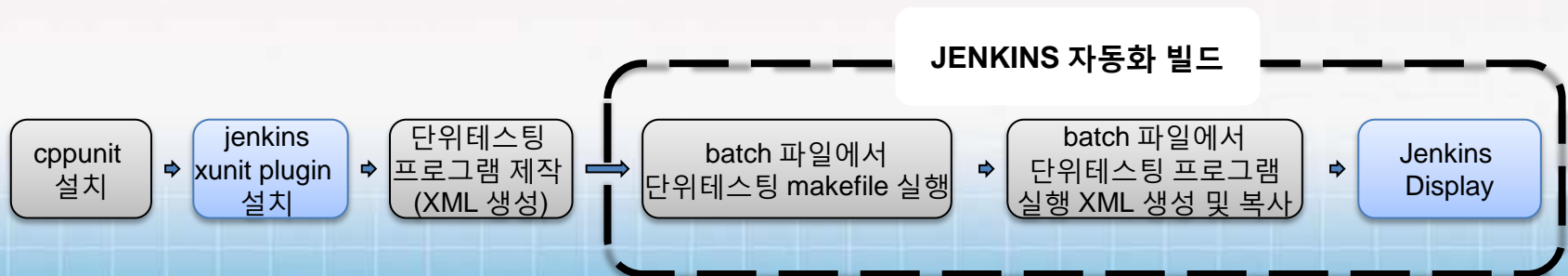
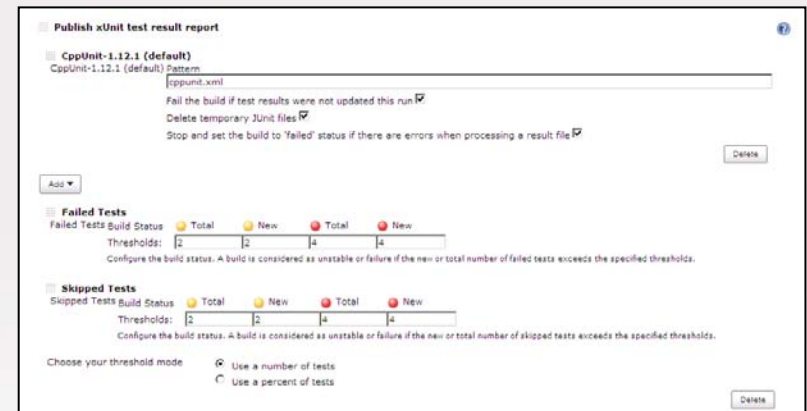
- plugin 설정

* 파일 경로 설정

* 테스트의 수준(thresholds) 설정 (fail / skip)

* 화면 display 설정

- 설치 및 동작



1. NSIQ

- 품질 지표

소스 코드에 대한 정적 분석으로 **LOC**, **Complexity** 를 계산

- plugin 설정

* (실행 설정) 파일 경로 설정 / 파일 필터 설정

* 화면 display 설정 (지표, encoding, high 수준)

- 설치 및 동작

