



code inspector
CODESCROLL

Manual

PART-1

Suresoft

CodeScroll™ Code Inspector Manual Part-1

문서정보

문서 버전: 3.7.4

작성일: 2016년 11월 30일

배포일: 2016년 12월 12일

기술지원

support@suresofttech.com

+82-2-6472-2800

최종고지

Copyright © Suresoft Technologies, Inc., All rights reserved.

Proprietary and Confidential

This material is proprietary to Suresoft Technologies, Inc. It contains trade secret and confidential information which is solely the property of Suresoft Technologies, Inc.

This material is for client's internal use only. This material shall not be used, reproduced, copied, disclosed, and transmitted, in whole or in part, without the express consent of Suresoft Technologies, Inc.

슈어소프트테크㈜ 허락 없이 무단 전재 및 배포하는 것을 금합니다.

본 문서 또는 제품에 대한 제안, 문의 사항은 슈어소프트테크㈜로 연락해주시시오.

본 문서에 포함된 일부 이미지는 버전에 따라 다소 차이가 있을 수 있으나,

설명하는 기능에 해당하는 이미지는 문서정보에 명시된 버전과 동일합니다.

목차

1. 시작하기 전에	7
문서의 목적	7
소프트웨어 최종 사용자 라이선스 계약	7
2. Code Inspector 개요	12
Code Inspector 소개	12
Code Inspector 특징	12
Code Inspector의 전체 화면	13
3. Code Inspector 설치	14
설치 요구 사항	14
윈도우 기반 Code Inspector 설치하기	15
4. Code Inspector 제거	19
윈도우 기반 Code Inspector 제거하기	19
5. Code Inspector 실행	23
6. Code Inspector 라이선스 설정	25
7. 툴체인(분석기) 설정	27
툴체인 추가	28
툴체인 편집	31
툴체인 복제	32
툴체인 삭제	32
8. Code Inspector 프로젝트 생성	33
프로젝트 만들기 전 확인 사항	34
소스 파일 C/C++ 프로젝트	35
비주얼 스튜디오 C/C++ 프로젝트	38
빌드 명령어 C/C++ 프로젝트	40
임베디드 C/C++ 프로젝트	42

기존 CodeScroll 프로젝트 정보로 C/C++ 프로젝트 생성	43
9. 인스펙션의 수행	44
선택적 인스펙션	46
새로 인스펙션	47
10. 인스펙션 결과의 확인	48
위배 요약 뷰	49
위배 상세 뷰	50
편집기	52
컴파일 오류 뷰	52
분석 정보 화면	53
11. 위배 자동 수정	54
위배 자동 수정	55
위배 자동 수정 미리보기	58
위배 자동 수정 되돌리기	61
자동 수정 가능한 위배만 보기	63
12. 위배 무시 기능	64
위배 무시 유지	66
주석으로 위배 무시하기	68
무시된 위배 제외하고 보기	72
위배 무시 내보내기/가져오기	72
위배 무시 초기화	75
13. 인스펙션 결과의 문서화	77
내보내기 기능을 통한 보고서 생성	77
뷰에서 내보내기	79
인스펙션 통합 보고서	80
소스 파일 위배사항 보고서	82
14. 규칙모음 설정	85
전역 규칙 설정	85

신규 규칙모음 생성	87
규칙 편집	91
규칙 로케일 변경	93
규칙모음 내보내기	94
규칙모음 가져오기	95
프로젝트에 도구 전역 규칙 적용하기	97
15. 도움말	98
도움말 메뉴	98
뷰 도움말	99
16. 환경 설정	101
매크로 제외	101
17. 문제해결	103
설치 시 보안프로그램과의 충돌	103
제품 실행 시 작업공간 지정 후 제품이 구동되지 않는 경우	103
규칙 설정 변경 내용이 적용되지 않을 경우	103
일부 규칙 설명이 보이지 않을 경우	104
규칙 유형 290번을 갖는 규칙을 사용하는 방법	104
위배 요약 파이차트에서 한글이 깨질 경우	105
Windows Vista/7 에서 설치 및 실행	105
프로그램 분석 실패 또는 Compile(전처리) 오류	106
프로젝트 초기화 시 유지되는 정보	106
Q-Scroll STAT에 데이터 업로드	107
기술지원	107
18. 부록 A - 위배 무시 유지 기능	108
소스 수정으로 인해 의도하지 않았던 위배를 무시하는 문제	108
소스 수정으로 인해 유효하지 않은 위배 무시를 유지하는 문제	109
19. 부록 B - 동글 라이선스 기능	111
동글 드라이버 설치	111
동글 드라이버 제거	114

1. 시작하기 전에

문서의 목적

CodeScroll™ Code Inspector 제품의 사용법에 관한 정보를 제공합니다.

소프트웨어 최종 사용자 라이선스 계약

중요: 본 최종 사용자 라이선스 계약("EULA")은 사용자(개인 또는 단체)와 본 EULA에 부착된 제품 인식 카드 혹은 인식 라벨로 식별되는 소프트웨어 제품("소프트웨어 제품" 또는 "소프트웨어") 제조업체인 슈어소프트테크 주식회사("슈어소프트테크") 사이의 법적 계약입니다. 소프트웨어 제품에는 컴퓨터 소프트웨어, 관련 미디어, 관련 인쇄물 및 "온라인" 또는 전자 설명서가 포함되어 있습니다. 소프트웨어 제품을 설치, 복사, 다운로드, 백업, 액세스하거나 사용하는 경우 본 EULA의 사용 조건에 동의하여야 하며, 본 EULA의 사용 조건에 동의하지 않으면 소프트웨어 제품에 대한 라이선스를 부여 받을 수 없습니다. 소프트웨어 제품에 대한 라이선스를 부여 받지 않는 경우, 본 소프트웨어 제품을 설치, 복사, 다운로드, 백업, 액세스하거나 사용할 수 없습니다.

소프트웨어 제품 라이선스

소프트웨어 제품은 지적 재산 보호 관련법뿐만 아니라, 저작권, 프로그램저작권 및 국제 저작권 규정에 따라 보호됩니다. 본 계약에서 사용되는 "컴퓨터"라는 용어는 단일 컴퓨터 시스템을 의미하며, 단일 컴퓨터 시스템의 범위에 대해서는 아래에 상술되어 있습니다.

1. 라이선스 부여.

본 EULA는 사용자에게 아래 권한을 부여합니다.

소프트웨어 설치 및 사용. 본 EULA에 명백히 설명한 것을 제외하고, 컴퓨터에 소프트웨어 제품 복사본을 하나만 설치, 사용, 액세스, 실행 또는 상호작용(“실행”)할 수 있습니다. 두 대 이상의 컴퓨터에서 소프트웨어 제품을 설치, 사용 또는 실행하기 위해서는, 각 컴퓨터 상에서 사용 또는 실행되는 소프트웨어의 상호작용 여부와 무관하게 컴퓨터의 대수에 해당되는 적법한 라이선스(들)를 사용자가 부여 받아야 합니다.

복사본 백업. 컴퓨터에 포함된 소프트웨어 제품 백업 본이 없는 경우, 사용자가 소프트웨어 제품의 컴퓨터 소프트웨어 부분에 해당하는 단일 백업 본을 작성할 수 있습니다. 기록의 목적으로 백업 복사본을 사용할 수 있습니다. 단일 백업 본을 작성하기 위해 백업 유틸리티를 사용할 수 있습니다. 본 EULA에 명백히 제공된 것을 제외하고, 소프트웨어 제품에 동봉된 설명서를 포함한 소프트웨어 제품의 복사본을 작성할 수 없습니다.

2. 기타 권리 및 제한사항 설명

컴퓨터 혹은 단일 컴퓨터 시스템. 소프트웨어 제품의 라이선스는 단일 컴퓨터 시스템에 적용됩니다. 단일 컴퓨터 시스템이 다수의 중앙처리장치(CPU)를 갖는 경우, 이 중앙처리장치들이 하나의 회로기판 상에 부착되어 있고 하나의 시스템 버스를 공유하는 대칭적 중앙처리 구조를 지니는 경우에 한하여 단일 컴퓨터 시스템으로 간주됩니다.

언어 버전 선택. 소프트웨어 제품이 한 개 이상의 언어 버전에 포함되어 있는 경우, 제공된 언어 버전 중 하나만 사용할 수 있도록 허가됩니다.

분해모방 제한사항. 디컴파일 및 분해. 소프트웨어 제품을 분해모방을 이용하여 디컴파일하거나 분해할 수 없습니다.

구성요소 분리. 소프트웨어 제품은 단일 제품으로 허가되었습니다. 한 대 이상의 컴퓨터에서 사용하기 위해 구성요소를 분리할 수 없습니다.

임대. 소프트웨어 제품을 임대, 리스 혹은 대여할 수 없습니다.

소프트웨어 제품 권리 이전. 본 EULA에 따라 하드웨어의 판매 또는 권리 이전의 일부분으로 모든 권한을 영구적으로 권리 이전할 수 있으며, 수령인은 본 EULA의 사용 조건에 동의하는 것으로 간주됩니다.

만료. 사용자가 본 EULA의 조항과 조건을 위반한 경우, 다른 권한의 침해 없이, 슈어소프트테크는 본 EULA의 기간 및 조건에 따라 권한을 만료시킬 수 있습니다. 이런 경우, 이미 설치된 소프트웨어 제품을 포함한 모든 소프트웨어 제품의 복사본 및 구성 부품을 폐기시켜야 합니다.

등록 상표. 본 EULA는 소프트웨어 제품 제조업체, 컴퓨터 제조업체, 기타 관련 공급업체의 등록 상표 또는 서비스 상표에 관련된 권한을 부여하지 않습니다.

업그레이드. 본 소프트웨어 제품이 슈어소프트테크에 의해 업그레이드되는 경우, 본 EULA의 사용 조건은 업그레이드된 소프트웨어에 그대로 적용되며 한 대 이상의 컴퓨터에서 사용하기 위해 분리될 수 없습니다.

3. 저작권.

소프트웨어 제품의 모든 권리와 지적소유권(소프트웨어 제품에 내장된 모든 이미지, 사진, 애니메이션, 비디오, 오디오, 음악, 텍스트 및 애플릿을 제한 없이 포함), 동봉된 설명서 및 모든 소프트웨어 제품의 복사본은 소프트웨어 제품 제조업체나 공급업체의 소유입니다. 소프트웨어 제품에 내장되어 있지는 않지만 소프트웨어 제품의 사용을 통해 액세스할 수 있는 내용에 대한 모든 권리와 지적 소유권은 해당 회사의 소유이며, 본 EULA는 이와 같은 내용의 사용에 대한 권한을 부여하지 않습니다. 소프트웨어 제품에 설명서가 파일로 포함되어 있는 경우, 해당 설명서 파일의 복사본을 한 부 인쇄할 수 있습니다. 소프트웨어 제품에 동봉된 인쇄물은 복사할 수 없습니다. 본 EULA에서 명백히 부여하지 않은 모든 권한은 해당 소프트웨어 제품 제조업체와 공급업체의 소유입니다.

4. 이중 미디어 소프트웨어 제품.

한 개 이상의 미디어로 소프트웨어 제품을 전달받을 수 있습니다. 받은 미

디어의 종류 또는 크기와 관계없이, 컴퓨터에 해당 미디어를 하나만 사용할 수 있습니다. 다른 컴퓨터에서 다른 미디어를 실행할 수 없습니다. 소프트웨어 제품의 영구적인 권리 이전(위의 설명대로)을 제외하고 다른 사용자에게 다른 미디어를 임대, 리스, 대여 또는 권리 이전할 수 없습니다.

5. 보증.

소프트웨어 제품의 품질에 대한 보증은 첨부된 보증서에 따르며, 별도의 보증서가 첨부되지 않은 경우 슈어소프트테크는 소프트웨어 제품을 구입일로부터 1년간 정상적으로 사용할 경우 소프트웨어 제조공정상의 결함이 발생하지 않음을 보증합니다. 이는 소프트웨어 제품이 오류를 갖지 않는 소프트웨어임을 의미하지 않으며, 사용자가 작업 도중 의도하지 않았던 결과를 초래할 가능성을 배제하지 않습니다. 별도의 보증계약에 의해 명백히 기술된 경우를 제외하고, 슈어소프트테크는 소프트웨어 제품의 부정확성, 오류 또는 손실에 대해 책임지지 않으며 이로 인하여 발생하는 손상 및 손해에 대해서도 책임지지 않습니다. 슈어소프트테크는 제3자가 제기한 배상 청구나 혹은 제3자를 위해 사용자가 제기한 배상 청구에 대해 책임지지 않습니다. 본 EULA에 언급된 보증제한 규정은 사용자가 상기의 가능성에 대해 슈어소프트테크 또는 슈어소프트테크의 공인 대리업체에 미리 통고한 경우에도 유효합니다.

6. 수출 제한사항.

소프트웨어 제품의 권리 이전은 대한민국 내에서만 가능하며, 타 국가로 수출될 수 없습니다.

7. JAVA 지원 참고.

소프트웨어 제품에는 JAVA로 작성된 프로그램에 대한 지원이 포함되어 있습니다. JAVA 기술은 오류를 허용하지 않으며, JAVA 기술 오류로 인하여 사망, 부상 또는 심각한 물리적, 환경적 손상을 유발할 수 있는 핵 설비, 항공기 탐색이나 통신 시스템, 항공 제어, 의약 또는 무기 시스템과 같은 오류 방지 성능이 필요한 위험한 환경에서 온라인 제어 장치로 설계되거나

제조 또는 사용, 판매되도록 제작되지 않았습니다.

주의사항

© 슈어소프트테크 주식회사. 모든 저작권 소유. 한국에서 인쇄.

Suresoft Technologies 및 CodeScroll은 슈어소프트테크 주식회사의 상표입니다.

이 문서 및 관련 문서에 언급된 제품명 혹은 상표명은 제품의 저작권을 소유한 회사의 상표 및 등록상표입니다. 이 문서에 기술되어 있는 소프트웨어는 사용 계약 및 비공개 계약을 통해 제공된 것입니다.

소프트웨어는 본 계약의 사용 조건에 따라서만 사용 또는 복사될 수 있습니다.

2. Code Inspector 개요

Code Inspector 소개

Code Inspector는 정적 코드 분석(static code analysis) 도구입니다.

Code Inspector 특징

검증 대상 소스 코드가 주어진 가이드라인, 패턴, 규칙을 따르는지 검사하고, 논리적 오류를 찾아 사용자에게 결과를 보여주므로, 개발자는 소프트웨어 소스 코드를 수정하기가 용이합니다.

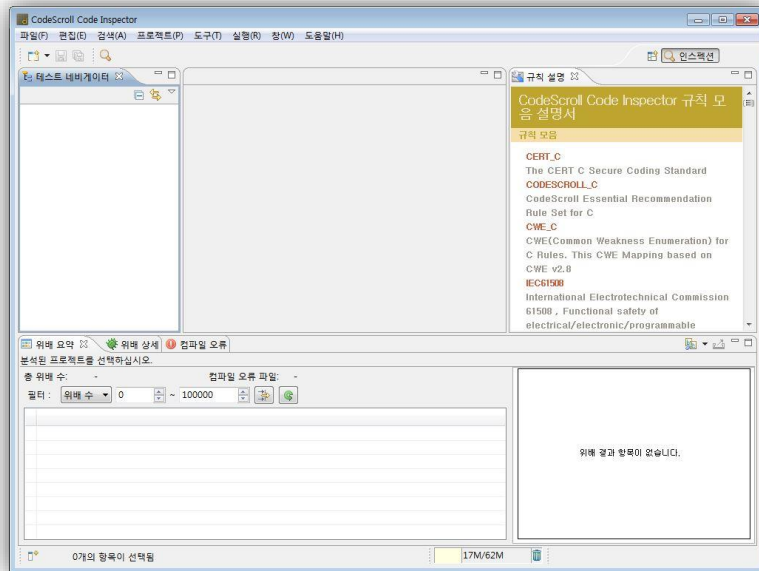
검증 대상 소스를 분석하여 그 결과를 다양한 차트와 그래프로 제공합니다.

다양한 개발 환경(컴파일러 등)을 지원하고 신속한 환경 설정이 가능하여 검증 소요 시간을 대폭 줄일 수 있습니다.

기존 규칙들을 이용하여 사용자만의 규칙모음을 만들 수 있습니다. 또한, Code Inspector에서 제공하는 규칙 유형들로 사용자만의 새로운 규칙을 만들 수 있습니다.

Code Inspector의 전체 화면

Eclipse RCP로 제작되어 자유도가 높은 뷰들을 제공합니다. [창] 메뉴와 각 뷰에서 사용자가 설정을 할 수 있습니다.



3. Code Inspector 설치

설치 요구 사항

OS	Microsoft Windows Vista / 7 (32bit/64bit)
RAM	최소 2GB 이상
HDD	약 500MB 의 여유 공간 (실제 분석이 수행되면 프로젝트의 결과물들의 저장으로 인해 HDD 사용량은 더 늘어납니다)

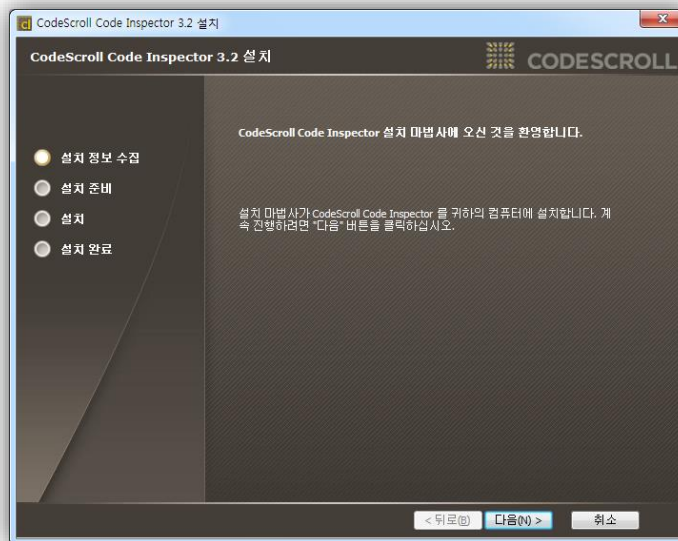
윈도우 기반 Code Inspector 설치하기

Code Inspector를 사용하기 위해서 배포된 설치 패키지를 이용하여 사전에 컴퓨터에 프로그램을 설치합니다. 설치 방법은 다음과 같습니다.

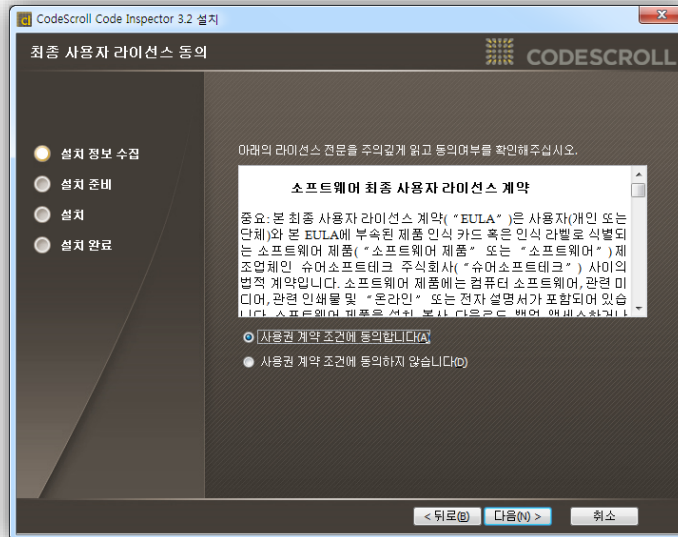
1. Setup.exe 파일을 실행 시킵니다.

이름	유형	크기
CodeScroll Code Inspector_3.x.msi	Windows Installer...	168,337KB
setup.exe	응용 프로그램	1,349KB

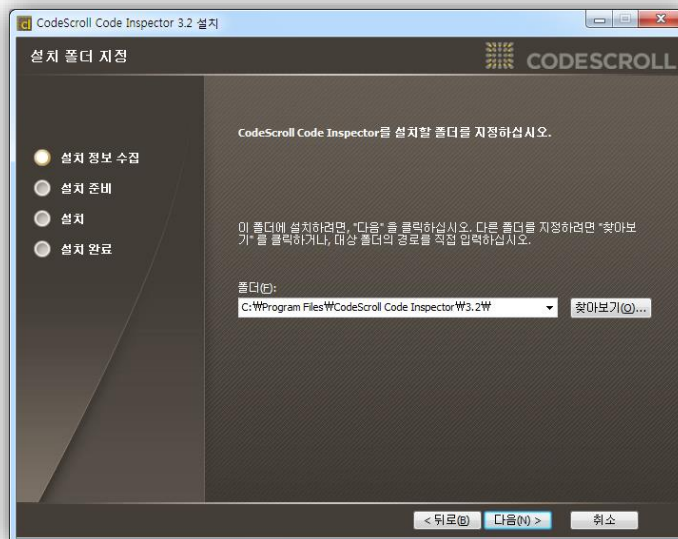
2. 설치 마법사가 실행되고 설치 정보를 수집합니다. [다음]을 클릭합니다.



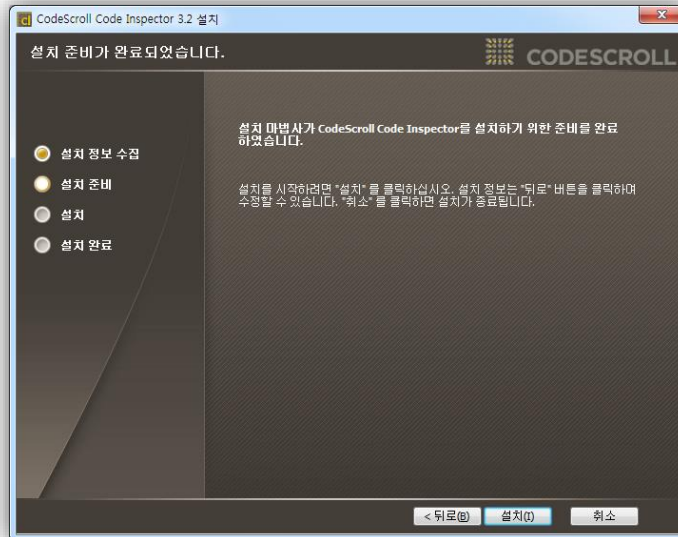
3. 최종 사용자 라이선스를 동의하고 [다음]을 클릭합니다.



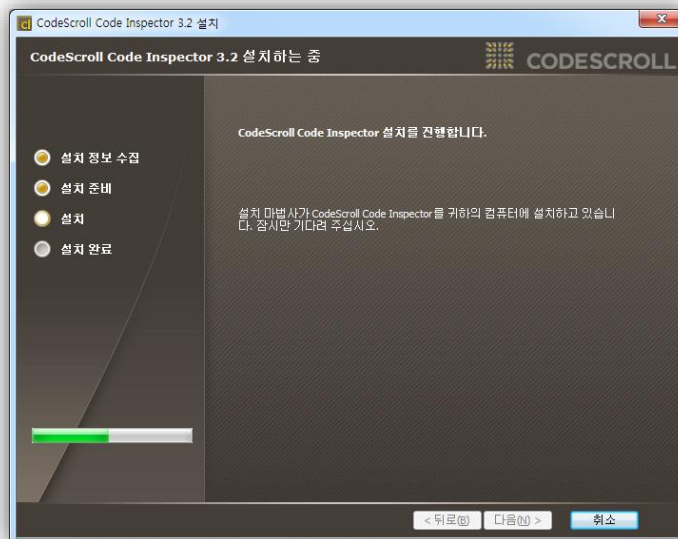
4. Code Inspector를 설치할 경로를 설정하고 [다음]을 클릭합니다.



5. 설치에 필요한 정보를 모두 수집하였습니다. [설치]를 클릭합니다.



6. Code Inspector가 설치됩니다.



-
7. 설치가 완료되었습니다. Code Inspector를 바로 실행할 때는 [CodeScroll Code Inspector 시작]을 체크하고 [마침]을 클릭하여 실행합니다.

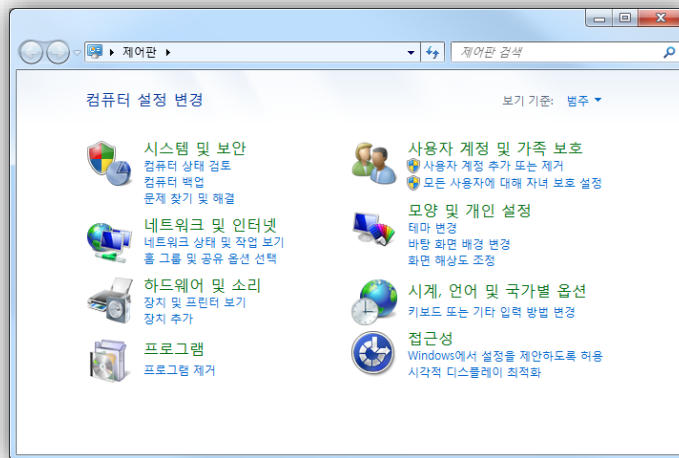


4. Code Inspector 제거

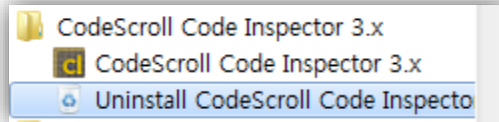
윈도우 기반 Code Inspector 제거하기

Code Inspector를 제거하는 방법은 3가지가 있습니다.

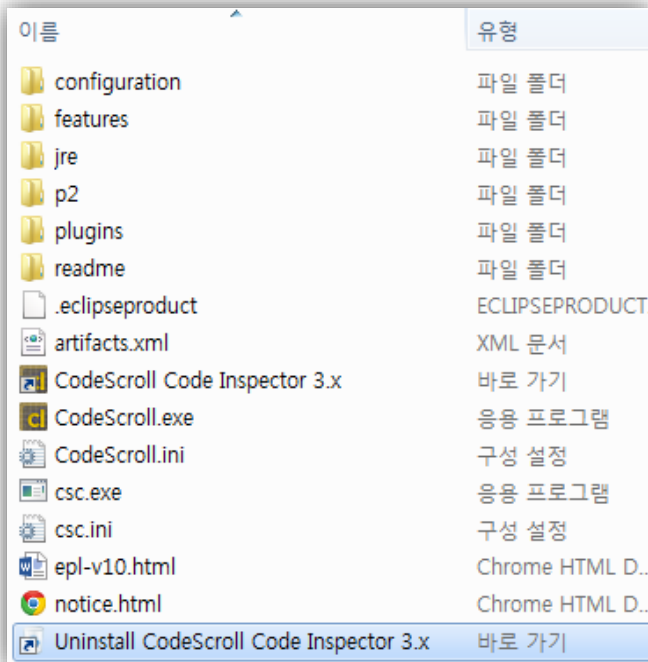
A. [제어판] -> [프로그램 제거]를 이용한 제거



-
- B. [시작] -> [모든 프로그램] -> [CodeScroll Code Inspector 3.x] -> [Uninstall CodeScroll Code Inspector 3.x]를 이용한 제거

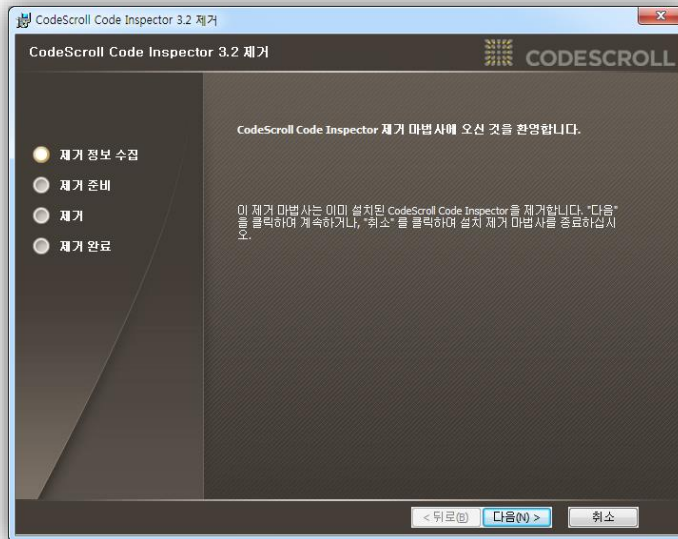


- C. [설치 경로] -> [Uninstall CodeScroll Code Inspector 3.x]를 이용한 제거

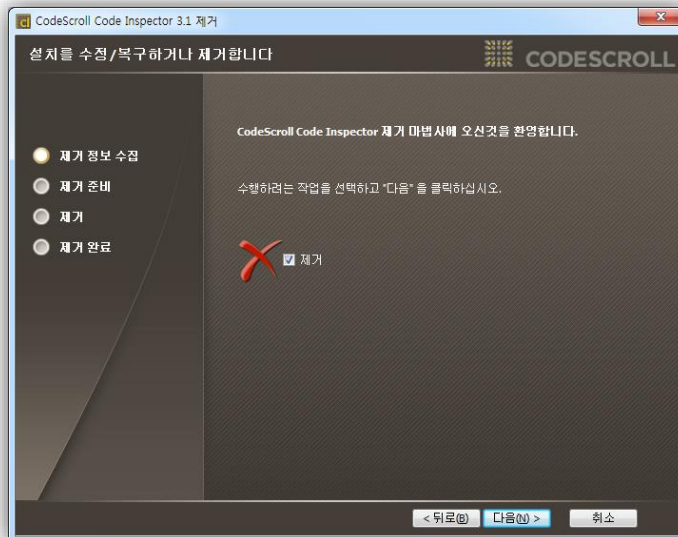


“Code Inspector 제거하기”는 Code Inspector가 동작하기 위해 필요한 파일들을 제거 합니다. 사용자가 설정한 워크스페이스는 제거되지 않습니다.

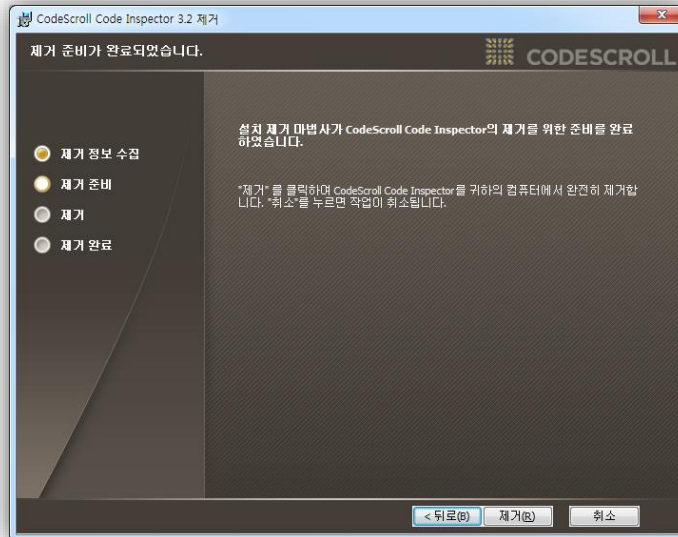
-
1. [CodeScroll Code Inspector 3.x 제거]를 실행하면 제거에 필요한 정보를 수집합니다. [다음]을 클릭합니다.



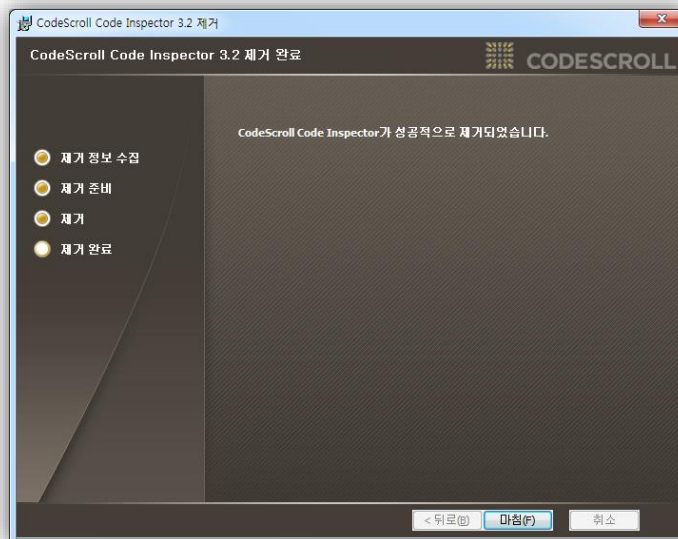
2. [제거]를 체크하고 [다음]을 클릭합니다.



3. 제거에 필요한 정보를 모두 수집하였습니다. [제거]를 클릭합니다.



4. Code Inspector가 성공적으로 제거 되면 [마침]을 클릭합니다.



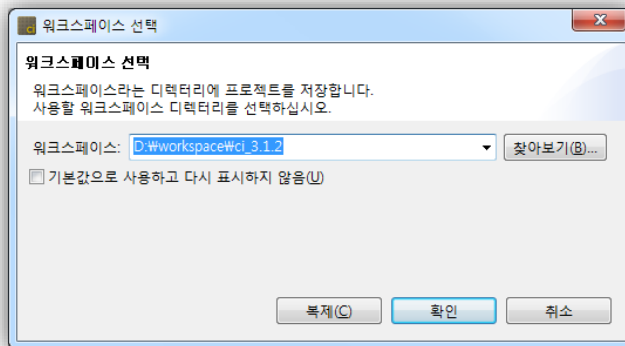
5. Code Inspector 실행

바탕 화면의 단축 아이콘 혹은 [시작] -> [모든 프로그램] -> [CodeScroll Code Inspector 3.x] -> [CodeScroll Code Inspector 3.x]를 선택합니다.



Code Inspector의 워크스페이스(작업 공간)를 설정합니다. 워크스페이스는 사용자가 지정할 수 있으며, 지정된 워크스페이스에 대한 설정은 저장할 수 있습니다. 경로 설정 후 [확인] 버튼을 클릭합니다.

- ✓ “기본값으로 사용하고 다시 표시하지 않음”을 체크하면, 이후 제품 구동 시 워크스페이스 선택창이 나타나지 않습니다.



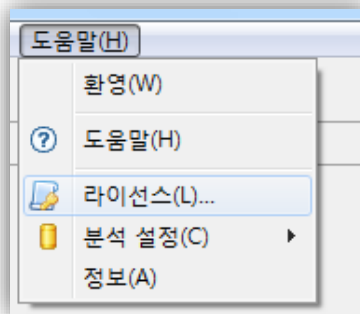
Code Inspector가 실행되면 [환영] 페이지가 나오고, 페이지를 닫으면 제품 화면이 나타납니다.



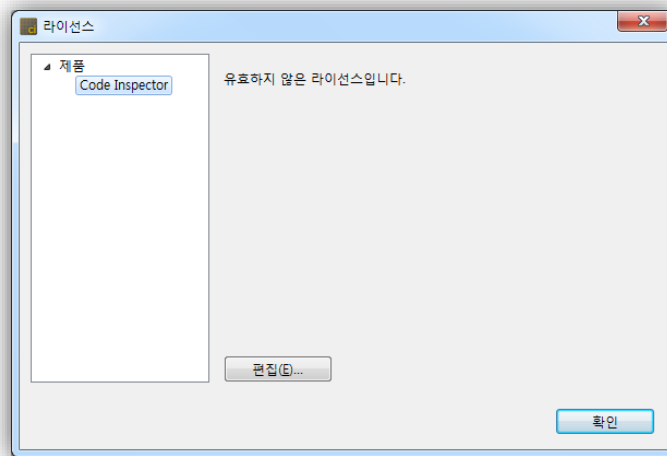
6. Code Inspector 라이선스 설정

Code Inspector를 사용하려면 먼저 라이선스를 등록해야 합니다. 라이선스 등록 과정은 다음과 같습니다.

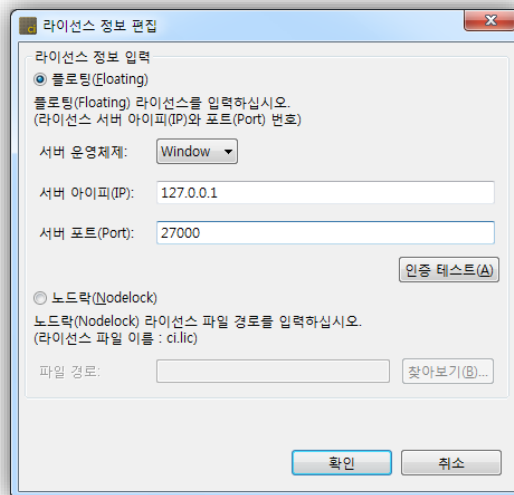
1. 메뉴의 [도움말] -> [라이선스]를 선택합니다.



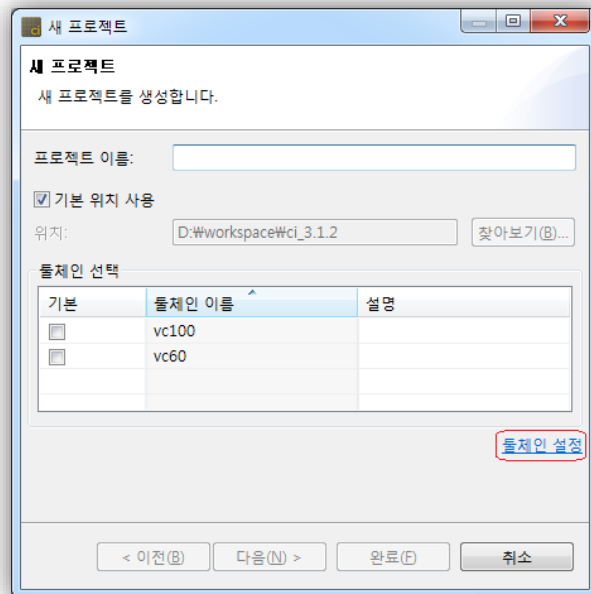
2. [제품] -> [Code Inspector]를 선택하고 [편집]을 클릭합니다.



3. 라이선스 정보 편집 창에서 제공받은 방식으로 라이선스를 입력하고 [확인] 버튼을 클릭하여 라이선스를 등록합니다.

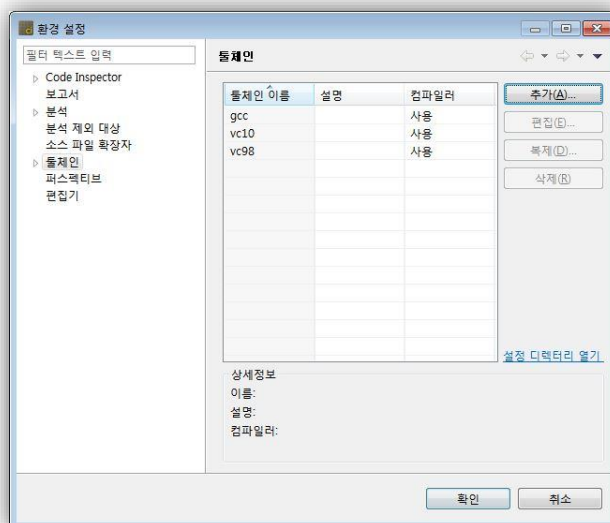


- A. 플로팅 방식 : 라이선스 서버를 통해 등록합니다. 서버 정보(운영 체제, 아이피(IP), 포트(Port))를 입력하고, [인증 테스트] 버튼을 클릭합니다.
 - B. 노드락 방식 : 제공받은 라이선스 파일을 통해 등록합니다.
- ※ 동글 라이선스 관련 설명은 부록 B를 참고바랍니다.

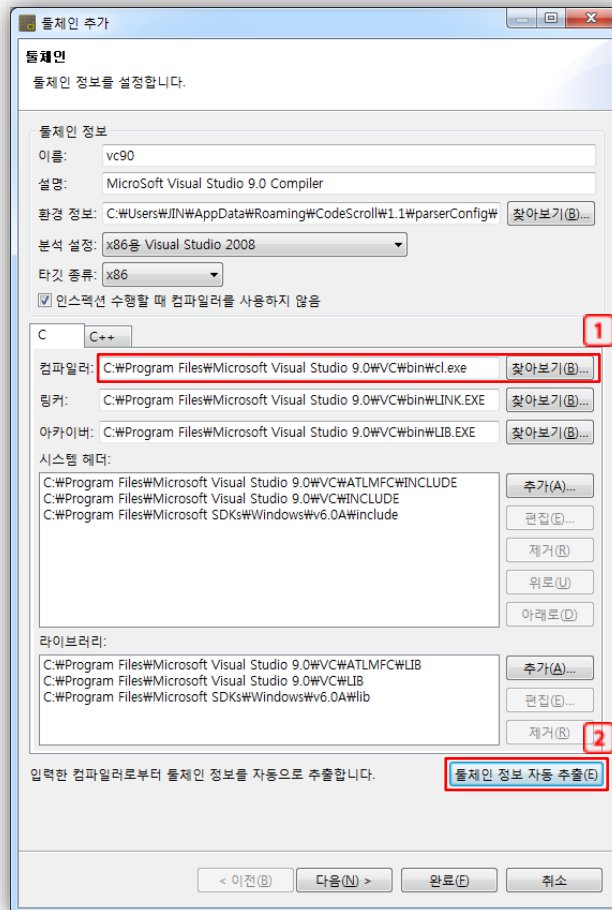


툴체인 추가

1. [환경 설정] -> [툴체인] 창에서 [추가] 버튼을 클릭합니다.



2. [툴체인 추가] 창에서 툴체인 정보를 입력합니다.



- ✓ [컴파일러] 1 입력 후 [툴체인 정보 자동 추출] 1 버튼을 클릭하면 입력한 컴파일러로부터 툴체인 정보를 자동으로 추출합니다.
- ✓ [컴파일러] 1란에는 컴파일러 경로를 찾아 입력합니다.

■ Ex)

- 비주얼 스튜디오 기반일 경우:

C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\bin\wd.exe

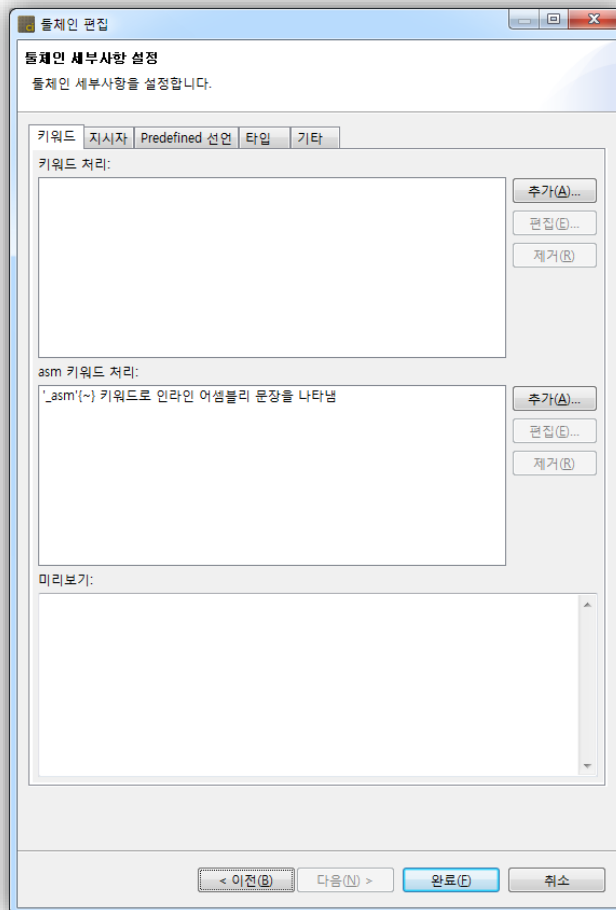
- GCC 기반일 경우:

C:\MinGW\bin\gcc.exe

- Freescale HC(S)12용 CodeWarrior 5.x 이상인 경우:

C:\Program Files (x86)\Freescale\WCWS12v5.1\Prog\Wchc12.exe

- A. [이름]에 생성하려는 툴체인 이름을 입력합니다.
- B. [설명]에 생성하려는 툴체인에 대한 설명을 입력합니다.
- C. [환경정보], [분석설정], [타겟 종류], [C/C++]의 입력정보들은 컴파일러로부터 정보를 추출하여 자동으로 설정되거나 사용자가 직접 입력할 수 있습니다.
- D. 툴체인 정보 입력 후 [다음>] 또는 [완료] 버튼을 클릭합니다.

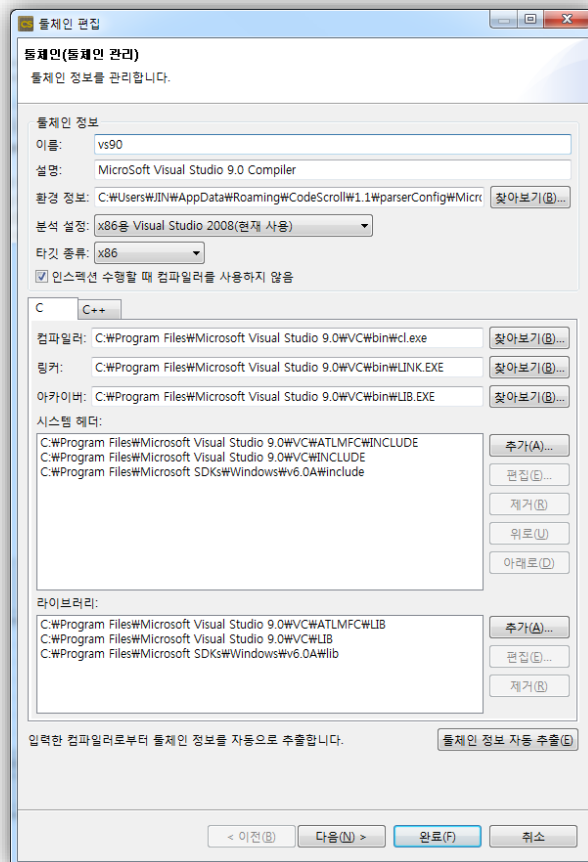


- E. [다음>] 버튼을 클릭하면 나오는 [툴체인 세부사항 설정] 창에서

는 [분석 설정]에서 선택한 분석 설정에 관련된 세부 항목들을 보여줍니다. 각 항목들은 사용자가 변경할 수 있습니다. 변경된 사항은 [미리보기]를 통해 어떻게 적용되는지 확인할 수 있습니다. 그 외 툴체인 세부사항에 대한 상세 설정에 대한 방법은 본 문서 2페이지에 있는 기술지원 연락처를 통해 문의해주시기 바랍니다.

툴체인 편집

1. 편집할 툴체인을 선택하고 [편집] 버튼을 클릭합니다.



2. 툴체인 정보를 편집합니다.

-
3. [완료] 버튼을 클릭합니다.

툴체인 복제

1. 복제할 툴체인을 선택한 후 [복제] 버튼을 클릭합니다.
(컴파일러가 입력되지 않은 툴체인만 복제가 가능)
2. 변경할 툴체인 정보가 있다면 편집합니다.
3. [완료] 버튼을 클릭합니다.

툴체인 삭제

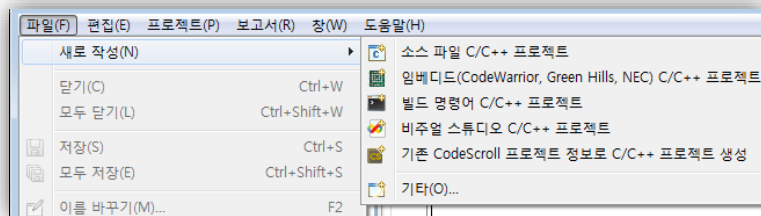
삭제할 툴체인을 선택한 후 [삭제] 버튼을 클릭합니다.
(삭제하려는 툴체인으로 기존에 생성된 프로젝트가 있을 경우에, 해당 프로젝트가 정상적으로 동작하지 않게 됩니다.)

8. Code Inspector 프로젝트 생성

Code Inspector 프로젝트를 생성하기 위해 [프로젝트 생성 마법사]를 실행합니다.

- A. 소스 파일 C/C++ 프로젝트(소스 파일을 직접 선택)
- B. 임베디드 C/C++ 프로젝트(임베디드 프로젝트 파일의 정보 추출)
- C. 빌드 명령어 C/C++ 프로젝트(Makefile 또는 빌드 스크립트 파일의 정보 추출)
- D. 비주얼 스튜디오 C/C++ 프로젝트(비주얼 스튜디오 프로젝트 파일의 정보 추출)
- E. 기존 CodeScroll 프로젝트 정보로 C/C++ 프로젝트 생성(기존 CodeScroll 프로젝트의 정보 추출)

메뉴의 [파일] -> [새로 작성]을 선택하거나 해당 단축 아이콘을 클릭합니다.





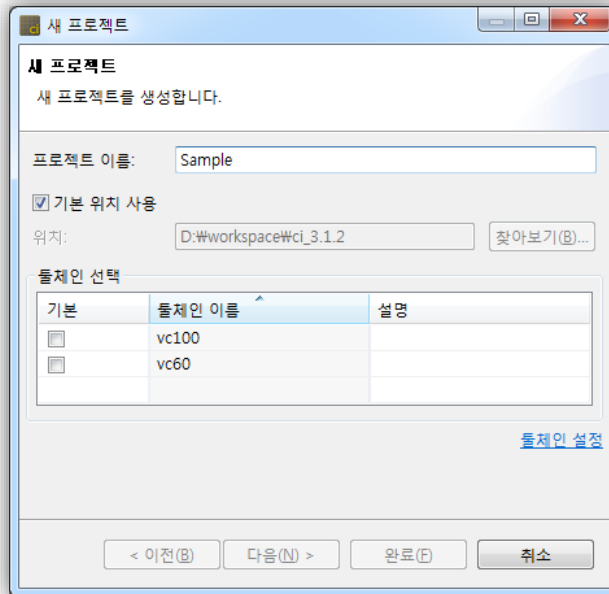
프로젝트 만들기 전 확인 사항

Code Inspector는 소스 코드를 컴파일하여 오류를 검출하는 도구이므로, 정상적인 테스트를 진행하려면 테스트 대상 소스 코드는 컴파일이 가능해야 합니다.

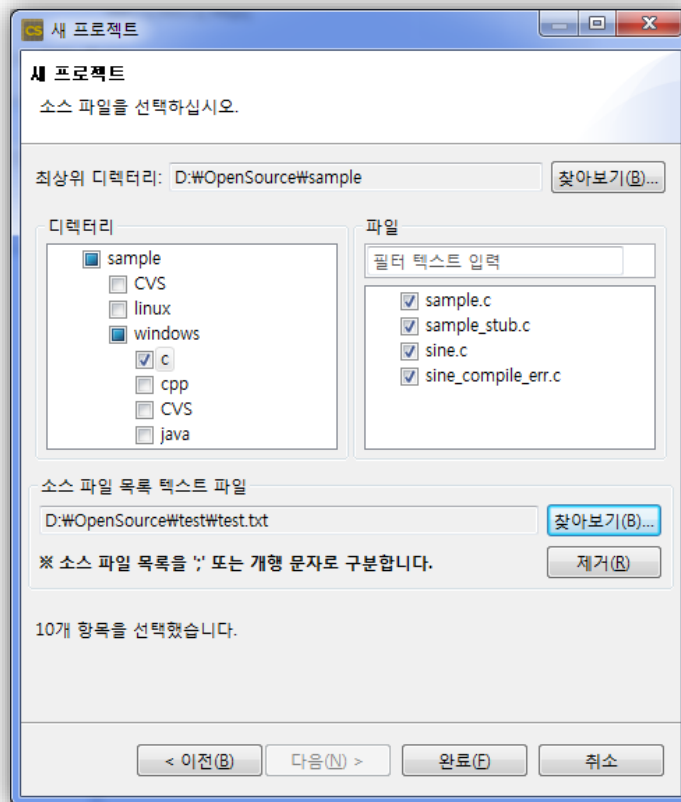
프로젝트를 생성하여 테스트를 진행하기 전에 대상 소프트웨어의 개발 환경을 고려한 ‘툴체인’ 설정 과정이 필요합니다. 자세한 ‘툴체인’ 설정 방법은 27p 툴체인(분석기)의 설정 부분을 참고하시기 바랍니다.

소스 파일 C/C++ 프로젝트

C 또는 C++ 소스 코드 파일만으로 프로젝트를 생성할 경우 선택합니다.
소스 코드 빌드에 필요한 정보를 사용자가 직접 입력해야 합니다.



1. [프로젝트 이름]에 프로젝트 이름을 입력합니다.
2. [기본 위치 사용] 항목 체크 유무로 프로젝트 저장 경로를 지정할 수 있습니다. [기본 위치 사용]을 체크했을 경우는 기본으로 설정한 워크스페이스를 사용합니다. 체크를 해지했을 경우는 [찾아보기...]를 이용해서 사용자가 경로를 지정할 수 있습니다.
3. [툴체인 선택]은 해당 프로젝트에서 사용할 툴체인을 선택합니다. 생성한 툴체인이 없다면 [툴체인 설정]을 통해 툴체인을 생성합니다. 툴체인 생성에 대해서는 **27p 툴체인(분석기)의 설정**을 참고 하시기 바랍니다.
4. 모든 설정을 완료한 후에 [다음>]을 클릭하여 소스 파일 설정 단계로 진행합니다.



- ✓ 소스 파일을 직접 선택하거나, 파일 경로가 쓰인 텍스트 파일을 통해 선택할 수 있습니다.

5. 소스 파일을 직접 선택

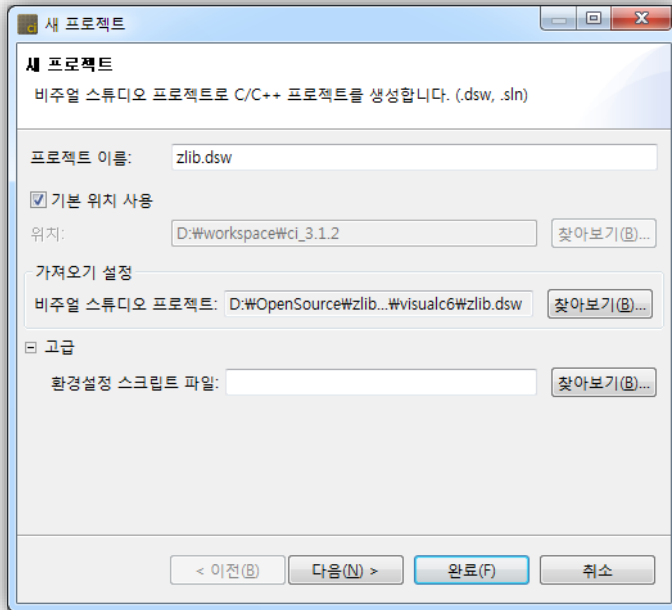
- A. [찾아보기...]를 클릭하여 아래 디렉터리 화면에 보일 [최상위 디렉터리]를 지정합니다.
- ✓ 최상위 디렉터리는 특별한 경우를 제외하고, 선택하려는 소스 파일을 담고 있는 디렉터리보다 한 단계 높은 수준의 디렉터리를 지정합니다.
- B. 왼쪽에 보이는 [디렉터리] 화면에서 프로젝트 생성에 사용할 소스 파일이 있는 디렉터를 선택합니다.
- C. 선택한 디렉터리에 있는 파일이 오른쪽에 보이는 [파일] 화면에

나타납니다. 추가할 파일을 체크합니다.

6. 파일 경로가 쓰인 텍스트 파일을 통해 선택
 - A. 아래에 보이는 [소스 파일 목록 텍스트 파일] 화면에서 [찾아보기...]를 클릭하여 소스 파일 목록(파일 절대 경로)이 기록된 텍스트 파일을 선택합니다.
 - B. [제거] 버튼을 클릭하면 지정한 텍스트 파일과 파일을 통해 선택된 소스 파일을 선택 해제할 수 있습니다.
7. 모든 설정을 완료한 후 [완료]를 클릭하여 프로젝트를 생성합니다.

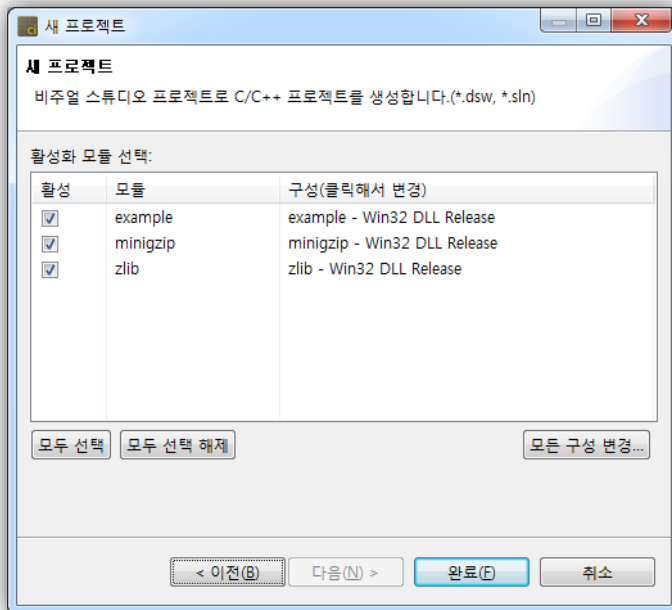
비주얼 스튜디오 C/C++ 프로젝트

비주얼 스튜디오 프로젝트 파일(.dsw, .sln)로 프로젝트를 생성할 때 선택합니다. 특별한 경우를 제외하고, 빌드에 필요한 정보는 자동으로 입력되므로 사용자가 별도로 지정할 내용이 없습니다.



1. [프로젝트 이름]에 프로젝트 이름을 입력합니다.
 2. [기본 위치 사용] 항목 체크 유무로 프로젝트 저장 경로를 지정할 수 있습니다. [기본 위치 사용]을 체크했을 경우는 기본으로 설정한 워크스페이스를 사용합니다. 체크를 해제했을 경우는 [찾아보기...]를 이용해서 사용자가 경로를 지정할 수 있습니다.
 3. [가져오기 설정] -> [비주얼 스튜디오 프로젝트]에서 인스펙션 대상 비주얼 스튜디오 프로젝트 파일을 지정합니다.
 4. [고급] -> [환경설정 스크립트 파일]은 등록된 툴체인이 사용하는 컴파일러가 특정 환경 변수를 사용하거나 환경 설정 스크립트 파일을 필요로 하는 프로젝트일 경우, 컴파일러를 호출하기 전에 반드시 설정해야 하는 값들이 저장된 스크립트 파일을 지정합니다.
-

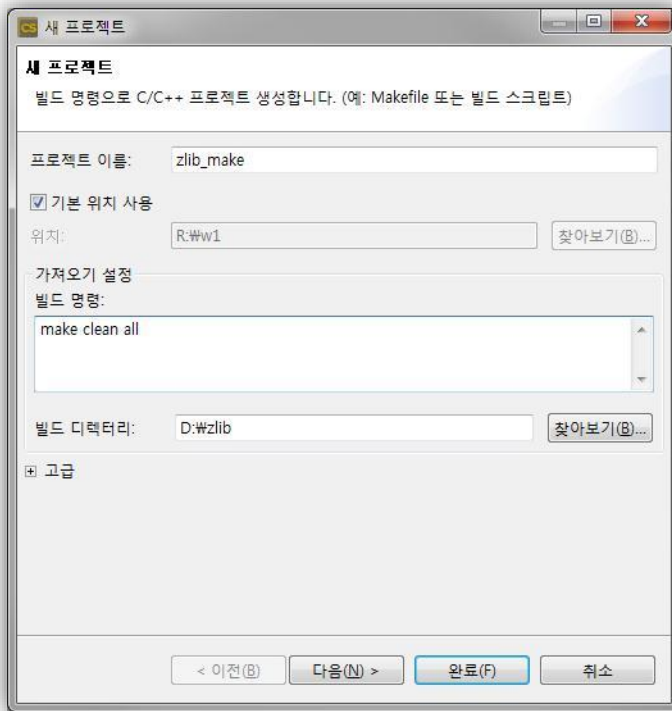
-
5. 모든 설정을 완료하고 [완료]를 클릭하여 프로젝트를 생성하거나 [다음 >]을 클릭하여 모듈 설정 단계로 진행합니다.



6. 비주얼 스튜디오 프로젝트에 포함된 모듈에서 활성화할 모듈을 선택합니다.
7. 각 모듈의 구성을 변경할 수 있습니다.
8. 모든 설정을 완료하고 [완료]를 클릭하여 프로젝트를 생성합니다.
- Windows 탐색기에서 비주얼 스튜디오 프로젝트 파일을 끌어서 테스트 네비게이터 뷰에 놓는 방법으로도 프로젝트를 생성할 수 있습니다.

빌드 명령어 C/C++ 프로젝트

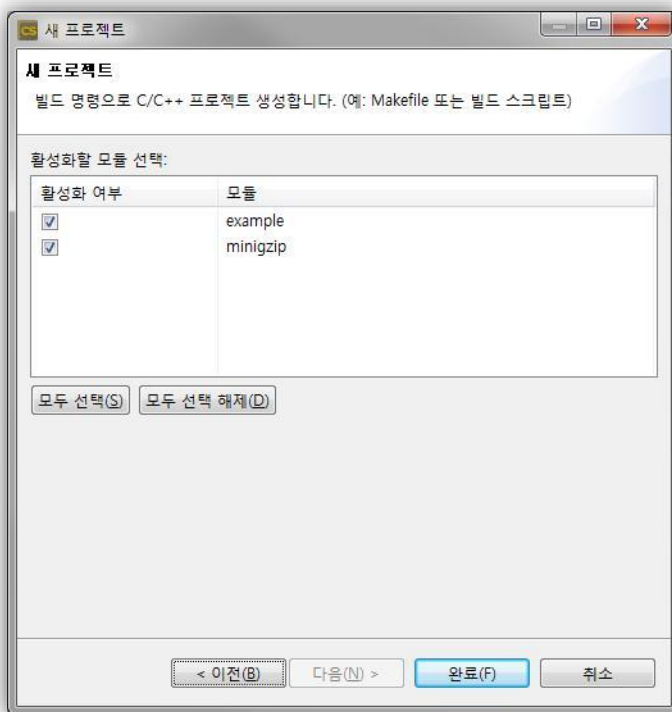
빌드 명령(makefile, build script)으로 C/C++ 프로젝트를 생성할 때 선택합니다. 사용자는 프로젝트의 빌드 명령으로 프로젝트를 생성합니다.



1. [프로젝트 이름]에 프로젝트 이름을 입력합니다.
 2. [기본 위치 사용] 항목 체크 유무로 프로젝트 저장 경로를 지정할 수 있습니다. [기본 위치 사용]을 체크했을 경우는 기본으로 설정한 워크스페이스를 사용합니다. 체크를 해제했을 경우는 [찾아보기...]를 이용해서 사용자가 경로를 지정할 수 있습니다.
 3. [가져오기 설정] -> [빌드 명령]에서 인스펙션 대상 프로젝트의 빌드 명령어를 입력합니다. (Ex. make all / make clean)
 4. [가져오기 설정] -> [빌드 디렉터리]에서 makefile이 저장되어 있는 경로를 지정해 줍니다.
 5. [고급] -> [환경설정 스크립트 파일]은 등록된 툴체인이 사용하는 컴파
-

일러가 특정 환경 변수를 사용하거나 환경 설정 스크립트 파일을 필요로 하는 프로젝트일 경우, 컴파일러를 호출하기 전에 반드시 설정해야 하는 값들이 저장된 스크립트 파일을 지정합니다.

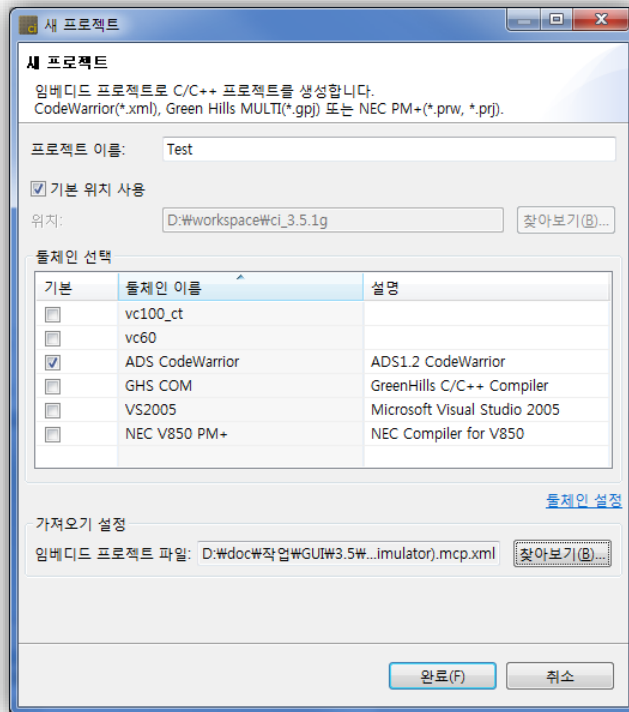
6. 모든 설정을 완료하고 [완료]를 클릭하여 프로젝트를 생성하거나 [다음>]을 클릭하여 모듈 선택 단계로 진행합니다.



7. makefile 포함된 모듈 목록 중에서 활성화 시킬 모듈을 선택합니다.
8. 모든 설정이 완료된 후 [완료]를 클릭하여 프로젝트를 생성합니다.

임베디드 C/C++ 프로젝트

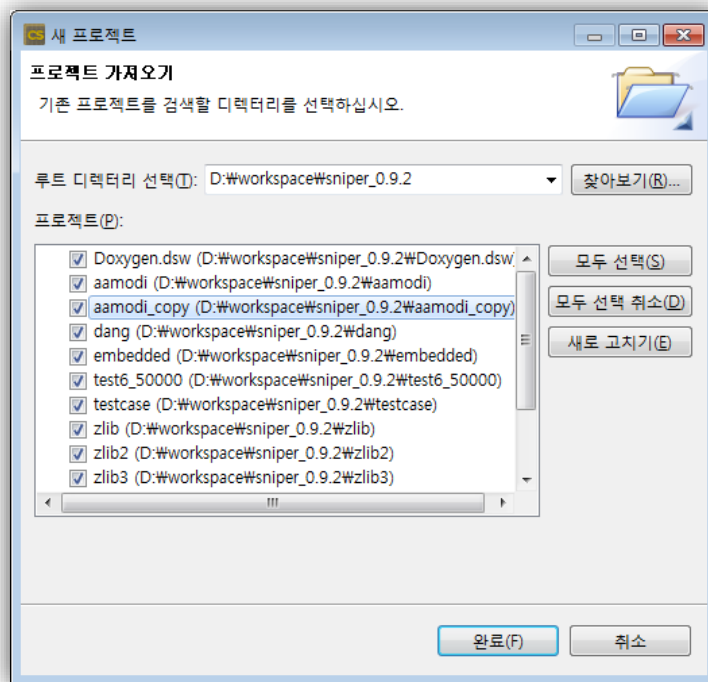
임베디드 프로젝트 파일(.xml, .gpj, .prw, .prj)로 프로젝트를 생성할 때 선택합니다. 특별한 경우를 제외하고, 빌드에 필요한 정보는 자동으로 입력되므로 사용자가 별도로 지정할 내용이 없습니다.



1. [프로젝트 이름]에 프로젝트 이름을 입력합니다.
 2. [기본 위치 사용] 항목 체크 유무로 프로젝트 저장 경로를 지정할 수 있습니다. [기본 위치 사용]을 체크했을 경우는 기본으로 설정한 워크스페이스를 사용합니다. 체크를 해제했을 경우는 [찾아보기...]를 이용해서 사용자가 경로를 지정할 수 있습니다.
 3. 생성할 임베디드 프로젝트에 맞는 툴체인을 선택합니다.
 4. [가져오기 설정] -> [임베디드 프로젝트 파일]에서 인스펙션 대상 임베디드 프로젝트 파일을 지정합니다.
 5. 모든 설정을 완료하고 [완료]를 클릭하여 프로젝트를 생성합니다.
-

기존 CodeScroll 프로젝트 정보로 C/C++ 프로젝트 생성

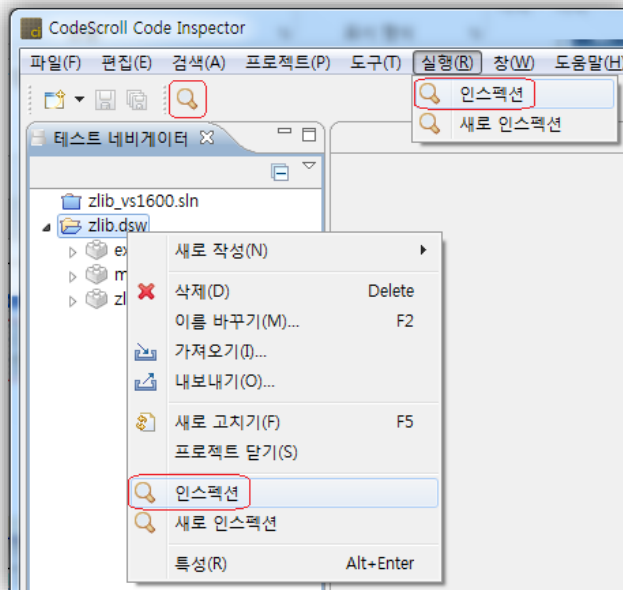
기존 CodeScroll 프로젝트를 임포트하여 해당 정보로 새로운 프로젝트를 생성합니다. 기존 프로젝트에 대한 여러 프로젝트 설정(소스파일, 분석 제외/포함, 컴파일 플래그)을 그대로 구성하여 같은 정보의 새 프로젝트를 쉽게 생성할 수 있습니다.




1. [찾아보기...] 버튼을 이용하여 CodeScroll 작업공간이나 CodeScroll 프로젝트 경로를 선택합니다.
 2. 선택한 경로에 존재하는 프로젝트가 [프로젝트] 목록에 표시됩니다.
 - ✓ CodeScroll 작업공간을 선택했을 경우, 하위의 모든 프로젝트들이 표시됩니다
 3. [완료]를 클릭하여 프로젝트를 생성합니다.
-

9. 인스펙션의 수행

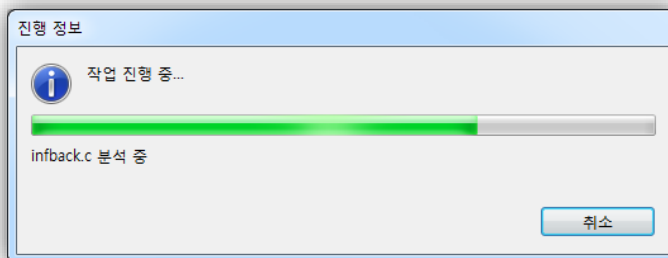
프로젝트를 생성하였으면 대상 프로젝트에 대한 인스펙션을 수행합니다.



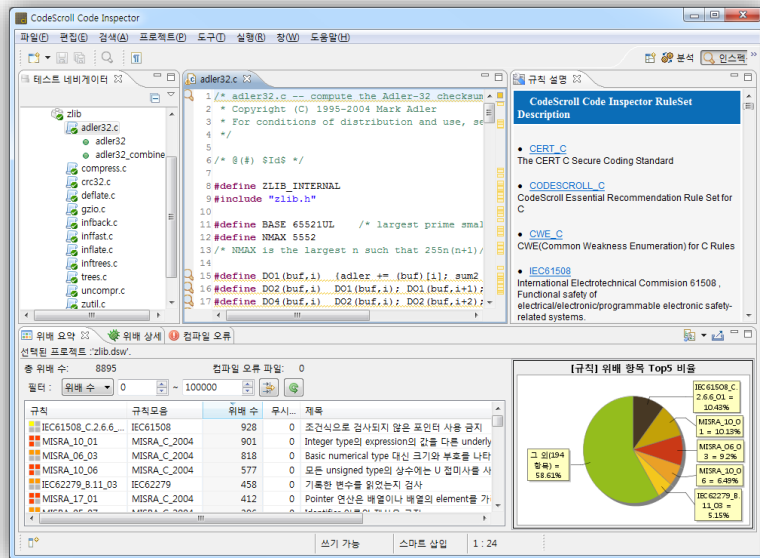
1. [테스트 네비게이터]에서 인스펙터를 수행할 프로젝트를 선택합니다.
2.  아이콘을 클릭하거나, 메뉴의 [실행] -> [인스펙션]을 클릭 또는 선택한 프로젝트의 오른쪽 클릭 메뉴에서 [인스펙션]을 클릭합니다.
3. 메뉴의 [창] -> [환경 설정] -> [Code Inspector] -> [전역 규칙 설정]에서 인스펙션에 포함할 규칙을 체크하지 않고 프로젝트를 생성했을 경우, 아래와 같이 [규칙 선택] 창이 열립니다.



4. [규칙 선택]에서 인스펙션 규칙모음(복수 선택 가능)을 선택합니다.
5. [전역 규칙 설정에 적용합니다.]를 체크하여 기본값으로 지정할 수 있습니다(기본값으로 지정된 규칙은 이후 생성하는 프로젝트에 기본으로 선택됨).
6. 규칙모음의 선택을 마쳤으면 [인스펙션]을 클릭하여 진행합니다.

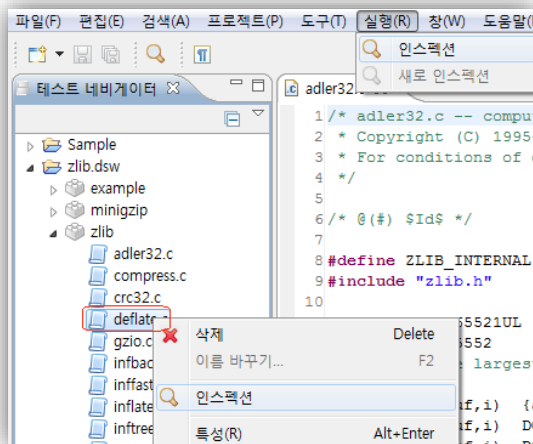



7. 인스펙션 수행을 완료하면 인스펙션 화면으로 전환합니다.



선택적 인스펙션

프로젝트 하위 모델인 모듈과 소스 파일을 선택하여 인스펙션 할 수 있습니다. 원하는 항목에 대해서만 인스펙션을 수행하므로 인스펙션 수행 시간을 단축시킬 수 있습니다.

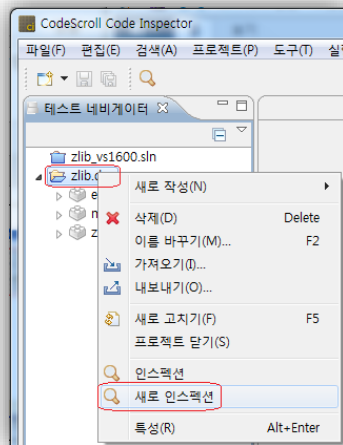


-
1. [테스트 네비게이터] 뷰에서 인스펙션을 수행할 모듈이나 소스 파일을 선택합니다.
 2. 인스펙션 아이콘()을 클릭하거나, 메뉴의 [실행] -> [인스펙션]을 클릭 또는 선택한 모델의 오른쪽 클릭 메뉴에서 [인스펙션]을 클릭합니다.
 3. 프로젝트에 기존에 인스펙션을 수행한 결과가 있으면 선택한 모델에 대해서만 인스펙션 결과가 최신화됩니다.

새로 인스펙션

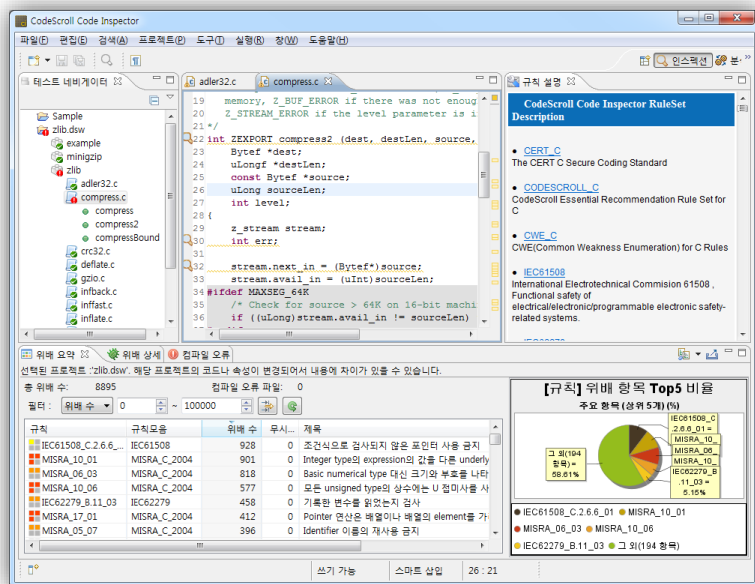
프로젝트에 기존 인스펙션 결과를 초기화한 후 인스펙션을 수행합니다.

1. [테스트 네비게이터] 뷰에서 인스펙션을 수행할 프로젝트를 선택합니다.
2. 메뉴의 [실행] -> [새로 인스펙션]을 클릭하거나, 선택한 프로젝트의 오른쪽 클릭 메뉴에서 [새로 인스펙션]을 클릭합니다.
 - ✓ 새로 인스펙션은 프로젝트 단위로만 수행할 수 있습니다.



10. 인스펙션 결과의 확인

인스펙션 결과 화면은 다음과 같이 구성됩니다.

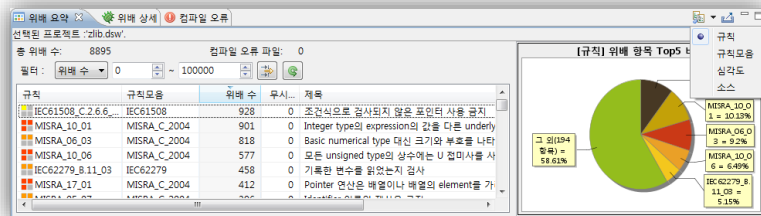




1. [테스트 네비게이터] 뷰: 인스펙션이 완료된 모델에 대해서는 초록 체크 표시가 보입니다. 또 소스 하위의 함수도 나타나게 됩니다.
2. [편집기]: 소스를 사용자가 수정할 수 있습니다. 수정한 소스를 저장하면 [테스트 네비게이터] 뷰에 해당 소스의 초록 체크 표시가 빨간 느낌표 표시로 변경됩니다.
3. [규칙 설명] 뷰: 상세한 규칙 설명을 표시하여 규칙 내용을 확인할 수 있습니다.
4. 인스펙션 결과를 보여주는 뷰는 [위배 요약], [위배 상세], [컴파일 오류]

뷰입니다.

위배 요약 뷰

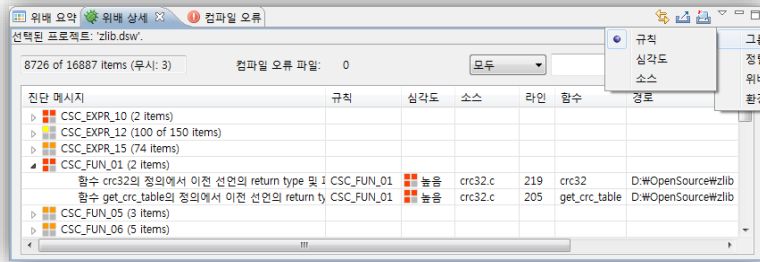
규칙모음을 이용하여 얻은 인스펙션 결과를 요약 기준에 맞게 도식화하여 보여줍니다.






1. 뷰의 오른쪽 위에 위치한 [위배 요약 모드]  에서 [규칙], [규칙모음], [심각도], [소스] 별 위배 요약 기준으로 위배 요약 확인할 수 있고 이에 맞는 형식의 위배 항목 파이 차트를 출력합니다.
2. [필터]에서는 위배 요약 모드 별 항목에 따라 제공되는 정보들을 기준으로 위배 요약 리스트를 필터링합니다.
3. [뷰 내용 내보내기]  로 위배 요약 보고서를 생성할 수 있습니다.

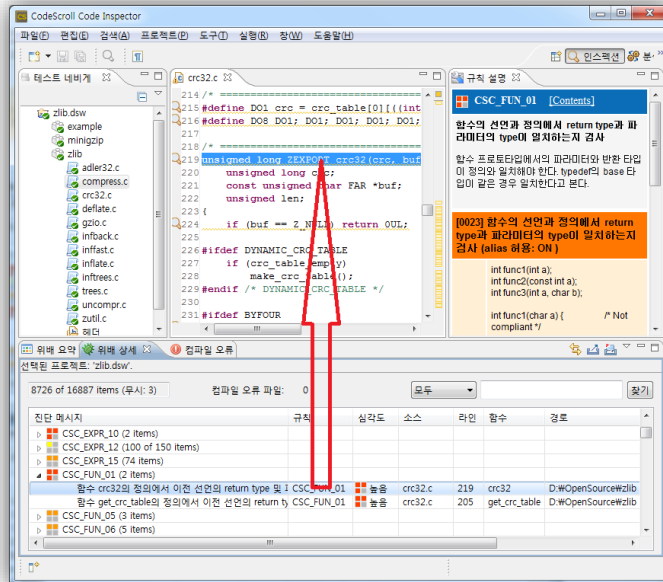
위배 상세 뷰

규칙모음을 이용하여 얻은 인스펙션 결과를 자세하게 도식화합니다.



1. 위배사항 리스트는 [진단 메시지], [규칙], [심각도], [소스], [라인], [함수], [경로]와 같은 정보를 보여줍니다.
2. 뷰의 오른쪽 위에 위치한 [보기 메뉴]  -> [그룹 별 보기]에서 [규칙], [심각도], [소스]를 선택하면 각 그룹 별로 진단 메시지를 보여줍니다.
3. [보기 메뉴] -> [정렬]에서 [진단 메시지], [규칙], [심각도], [소스], [라인], [함수], [경로]를 선택하면 그룹 별로 해당 항목 순으로 정렬하여 진단 메시지를 보여줍니다.
4. [보기 메뉴] -> [위배 무시 초기화]를 선택하면 위배 항목에 설정한 위배 무시들이 모두 삭제됩니다(소스 코드의 위배 무시 주석 제외).
5. [보기 메뉴] -> [환경 설정]을 선택하면 그룹 당 보이는 항목의 수를 조정할 수 있습니다.
6. [뷰 내용 내보내기]  로 위배 상세 보고서와 위배 무시 상세 보고서를 생성할 수 있습니다.
7. [찾기] 기능으로 원하는 위배사항을 찾을 수 있습니다. 찾기 설정은 규칙, 진단 메시지, 함수, 소스 중에서 가능합니다..
8. 각 항목을 더블 클릭([편집기와 연결]  이 On이면 원클릭)하면 편집기에서 선택된 위배를 보여줍니다.
 - A. [위배 상세] 뷰에서 각 항목을 클릭/더블클릭합니다.
 - B. [편집기]에서 규칙 위배사항이 발생한 부분으로 이동하며 해당하

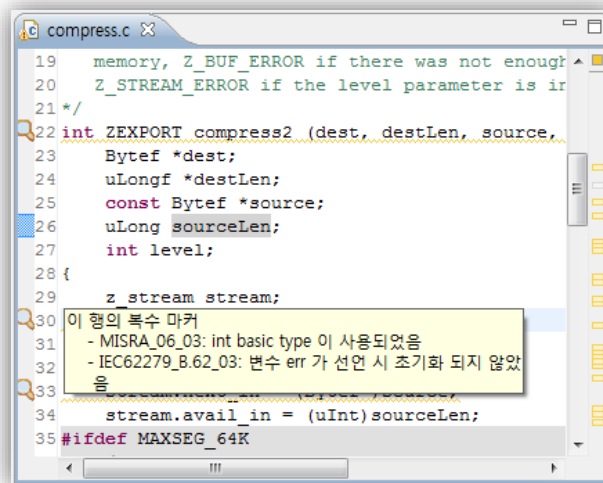
는 소스를 블록으로 보여줍니다.




C. [규칙 설명] 뷰에서 선택한 위배 항목의 규칙 설명을 보여줍니다.

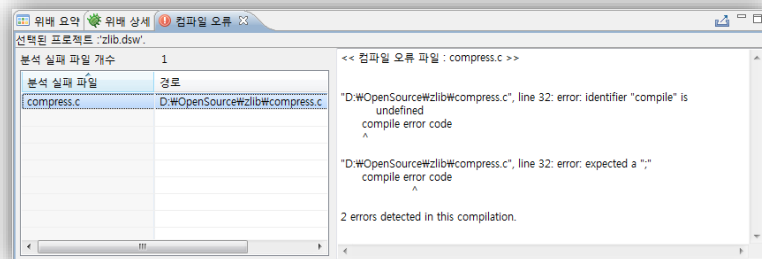
편집기

사용자는 소스 코드 편집기를 이용하여 위배 코드를 수정할 수 있습니다. 또한, 위배사항에 대해 라인 별 표시를 제공합니다.



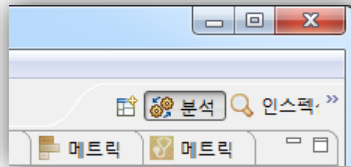
컴파일 오류 뷰

선택한 프로젝트에서 인스펙션 수행 시에 컴파일 오류가 발생한 파일에 대한 컴파일 오류 메시지를 보여줍니다. [뷰 내용 내보내기]  로 컴파일 오류 보고서를 생성할 수 있습니다

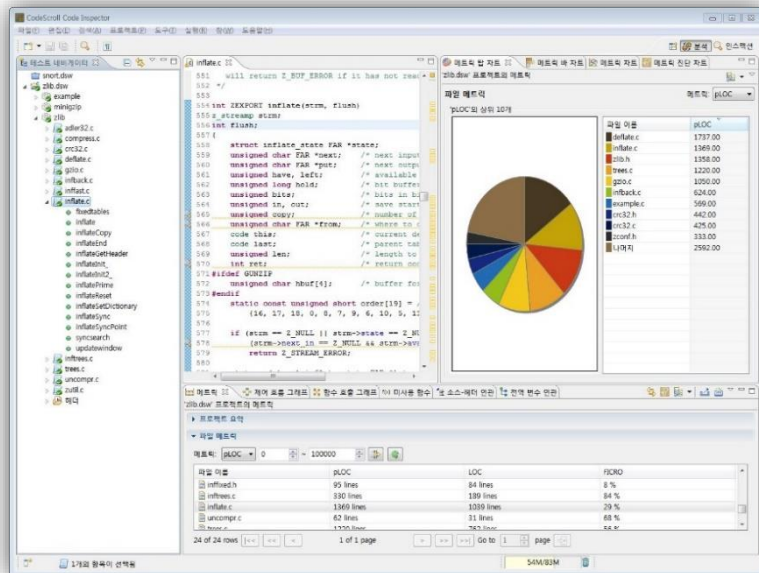


분석 정보 화면

인스펙션 대상인 소스 파일 정보를 정량화한 그래프와 차트로 확인할 수 있습니다. 분석 정보 화면에 포함된 뷰에 대한 상세 설명은 “PART-2 메뉴얼”을 통해 확인할 수 있습니다.



1. 제품 오른쪽 위에 위치한 [분석]을 클릭하면 분석 정보 화면을 보여줍니다.
2. 메트릭 차트 뷰에서 [메트릭 차트], [메트릭 탭 차트], [메트릭 바 차트], [메트릭 진단 차트]를 선택하면 다양한 그래프를 보여줍니다.
3. 화면 하단 뷰의 [메트릭], [제어 흐름 그래프], [함수 호출 그래프], [미사용 함수], [소스-헤더 연관], [전역 변수 연관] 탭을 선택하면 프로젝트 분석 정보와 소스나 함수의 흐름과 연관관계를 보여줍니다.



11. 위배 자동 수정

‘위배 자동 수정’은 검출되는 수가 많아 일일이 수정하기 어려운 위배를 자동으로 수정하는 기능입니다.

인스펙션을 수행한 뒤에 자동 수정이 가능한 위배가 존재하면 자동 수정을 제안하는 창이 나타나며, 이 창을 통해 위배를 자동으로 수정할 수 있습니다.

또한, [테스트 네비게이터] 뷰와 [위배 상세] 뷰에서 프로젝트 또는 규칙이나 위배의 오른쪽 클릭 메뉴를 통해 위배를 자동으로 수정할 수 있습니다.

다음은 자동으로 수정할 수 있는 규칙의 종류입니다.

1. C 스타일 주석만 사용하였는지 검사
2. 반복문 또는 선택문의 body는 복합문이어야 함
3. else if 가 있다면 else가 반드시 있는지 검사
4. switch문에 default절의 존재 검사
5. Literal의 접미사 ‘L’이 대문자인지 검사
6. 한 줄에 두 개 이상의 문장 사용 금지

대표적인 규칙은 다음과 같습니다(아래 나열한 규칙 외에도 같은 종류의 규칙은 자동으로 수정할 수 있습니다).

MISRA 2004

- MISRA_02_02, MISRA_14_08, MISRA_14_09, MISRA_14_10

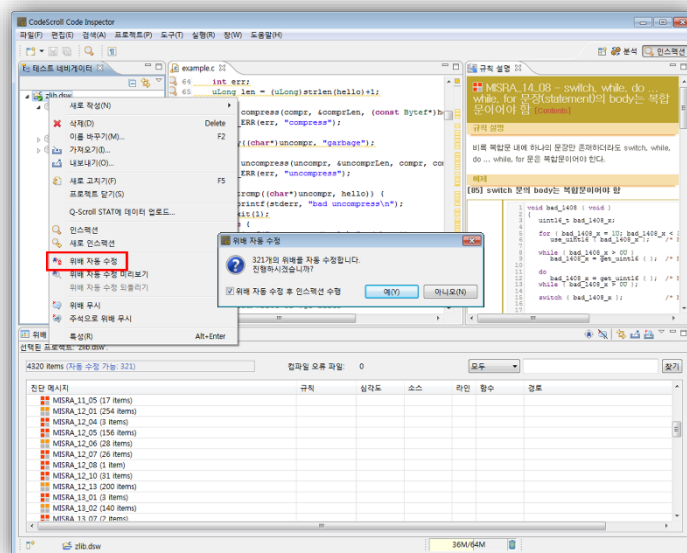
MISRA 2012

- MISRA_C_2012_07_03, MISRA_C_2012_15_06, MISRA_C_2012_15_07, MISRA_C_2012_16_04

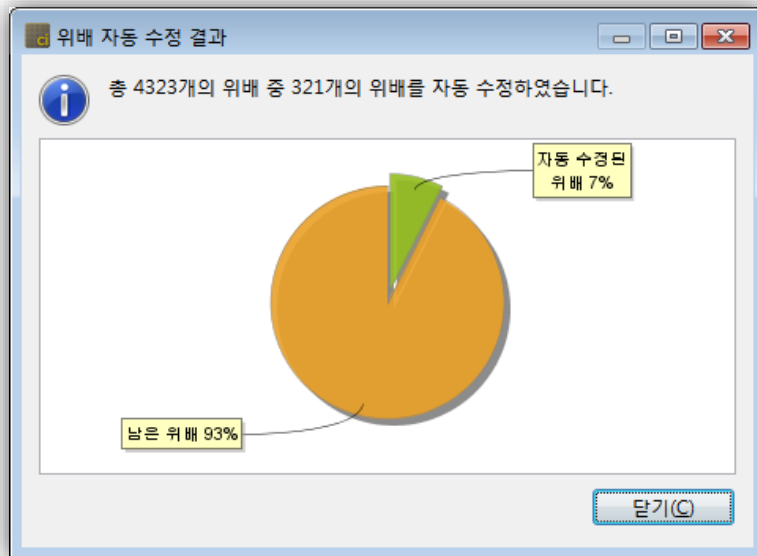
위배 자동 수정

[테스트 네비게이터] 뷰나 [위배 상세] 뷰의 오른쪽 클릭 메뉴에서 위배를 자동으로 수정할 수 있습니다.

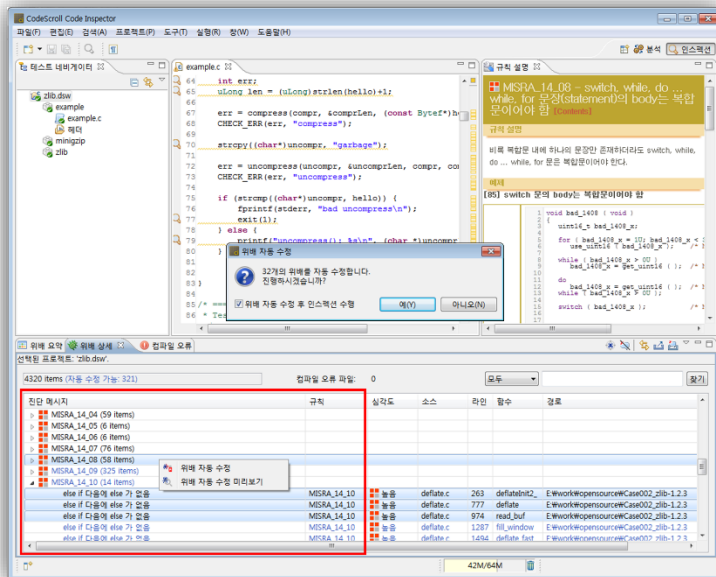
[프로젝트 전체 위배를 자동 수정하기]



1. [테스트 네비게이터] 뷰에서 분석이 완료된 프로젝트를 선택한 뒤, 오른쪽 클릭 메뉴에서 [위배 자동 수정]을 클릭합니다.
2. 수정되는 위배의 개수를 확인하고, [예] 버튼을 클릭합니다.
3. 위배를 자동으로 수정한 뒤에 인스펙션이 수행됩니다(이전 단계에서 [위배 자동 수정 후 인스펙션 수행] 버튼을 체크한 경우).
4. 위배 자동 수정 결과 창이 열리고, 결과를 확인할 수 있습니다.

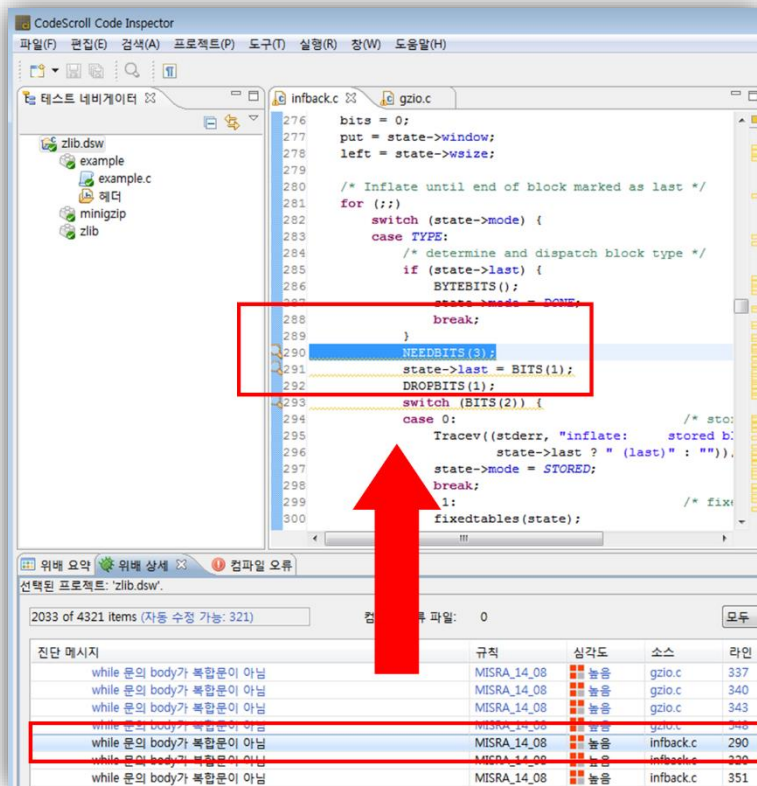


[선택한 위배만 자동 수정하기]

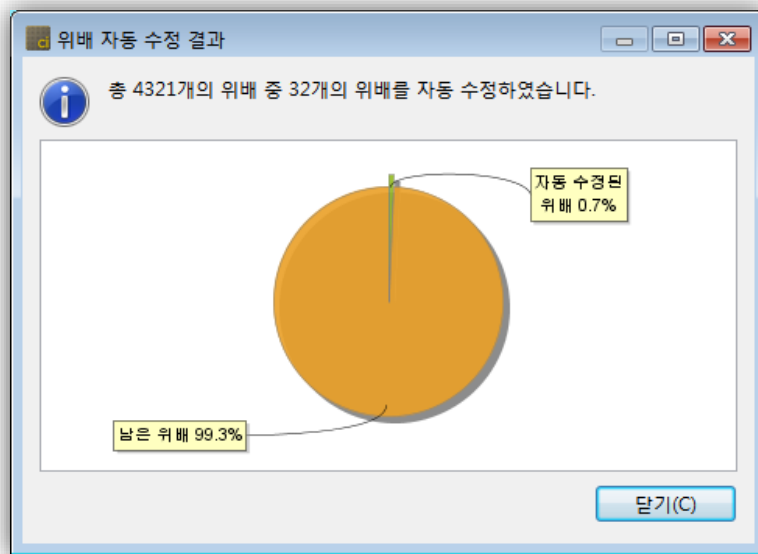


a

1. [위배 상세] 뷰에서 수정하려는 규칙이나 위배를 선택한 뒤, 오른쪽 클릭 메뉴에서 [위배 자동 수정]을 클릭합니다.
2. 수정되는 위배의 개수를 확인하고, [예] 버튼을 클릭합니다.
※ 위배가 발생한 라인에 매크로 함수가 포함된 경우에는 위배를 자동으로 수정할 수 없습니다.



- ※ 자동으로 수정할 수 있는 위배 또는 자동으로 수정할 수 있는 위배가 포함된 상위 그룹은 파란색으로 표시됩니다(항목을 선택하면 검은색으로 표시됨).
3. 위배를 자동으로 수정한 뒤에 인스펙션이 수행됩니다(이전 단계에서 [위배 자동 수정 후 인스펙션 수행] 버튼을 체크한 경우).
4. 위배 자동 수정 결과 창이 열리고, 결과를 확인할 수 있습니다.

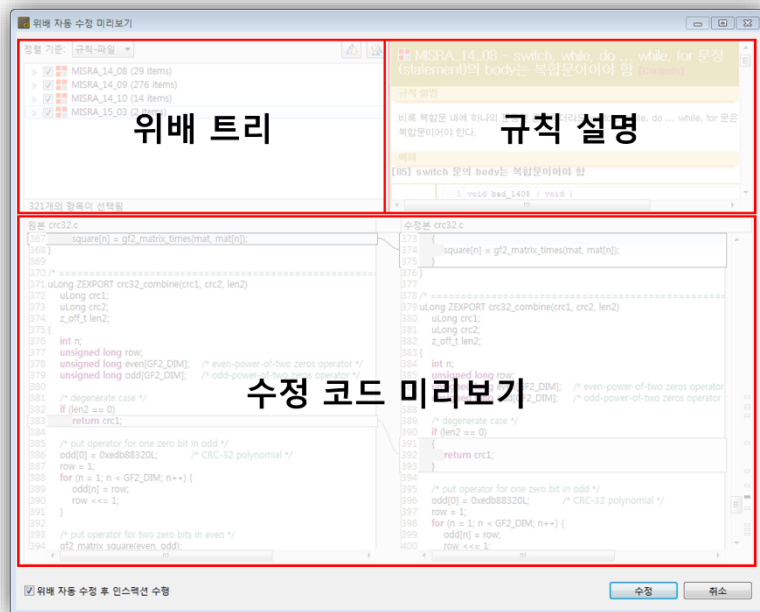


위배 자동 수정 미리보기

위배를 자동으로 수정하기 전에, 수정 결과를 미리 볼 수 있습니다.

미리보기 창은 [테스트 네비게이터] 뷰나 [위배 상세] 뷰의 오른쪽 클릭 메뉴에서 실행할 수 있습니다.

위배 자동 수정 미리보기 창의 화면 구성은 다음과 같습니다.



위배 트리 부분

- 정렬 기준 콤보: 위배 트리를 구성하는 정렬 기준을 선택합니다.
- 다음 / 이전 버튼: 선택된 위배의 다음이나 이전 위배를 선택합니다.
- 위배 항목 트리: 정렬 기준에 맞게 상위 노드가 구성된 위배 트리.

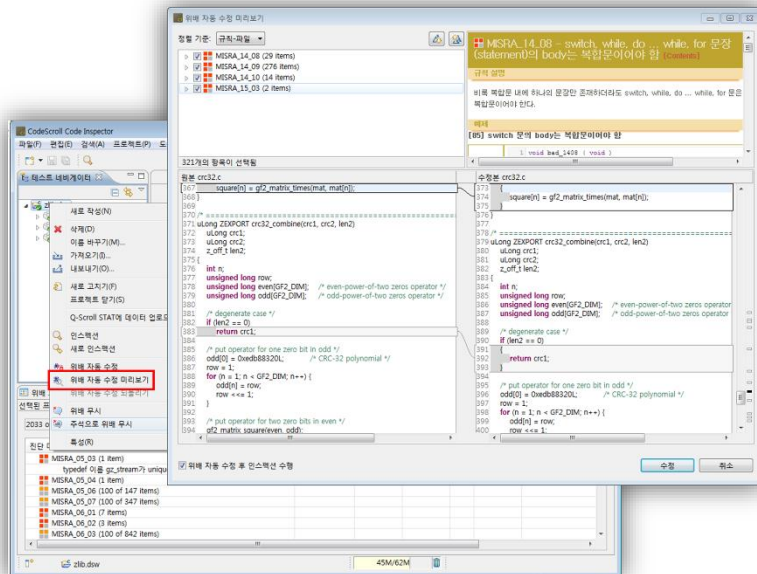
규칙 설명 부분

- 선택된 위배에 관한 규칙 설명을 보여줍니다.

수정 코드 미리보기 부분

- 왼쪽은 수정 전, 오른쪽은 수정 후의 소스 코드를 보여줍니다.

[프로젝트 전체 위배의 자동 수정 결과 미리보기]

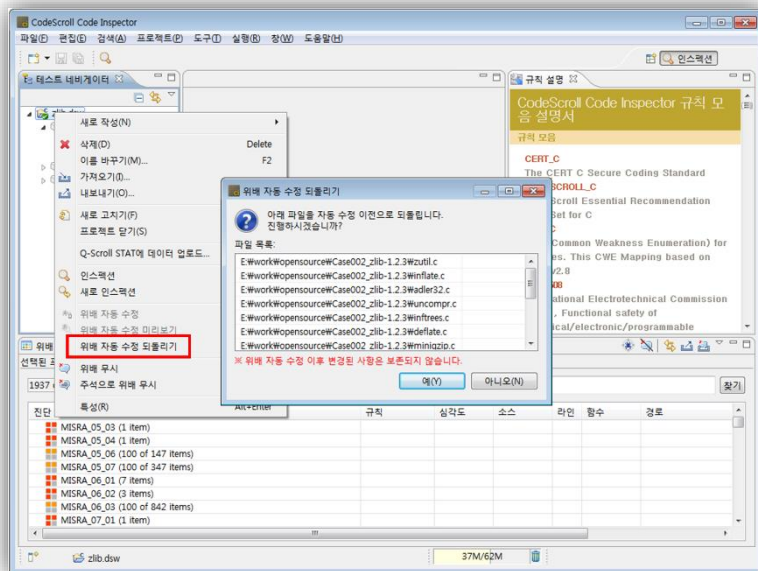


1. [테스트 네비게이터] 뷰에서 분석이 완료된 프로젝트를 선택한 뒤, 오른쪽 클릭 메뉴에서 [위배 자동 수정 미리보기]를 클릭합니다.
2. 좌측 상단의 트리에서 수정 결과를 확인할 위배를 선택합니다.
3. 하단의 미리보기 화면에서 수정 전과 후의 소스 코드를 비교한 뒤에, 좌측 상단의 트리에서 수정할 위배를 체크합니다(기본값은 모두 체크).
4. [수정] 버튼을 클릭하면 위배를 자동으로 수정합니다.
5. 위배를 자동으로 수정한 뒤에 인스펙션이 수행됩니다(이전 단계에서 [위배 자동 수정 후 인스펙션 수행] 버튼을 체크한 경우).
6. 위배 자동 수정 결과 창이 열리고, 결과를 확인할 수 있습니다.

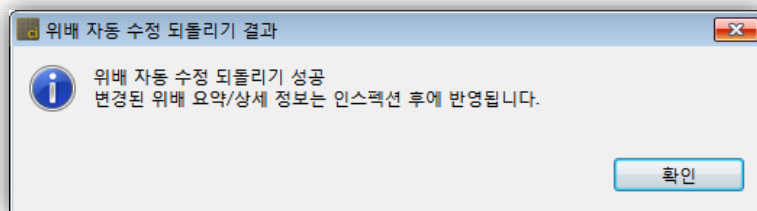


위배 자동 수정 되돌리기

자동 수정한 소스 파일을 마지막 자동 수정 수행 전 상태로 되돌립니다.
[테스트 네비게이터] 뷰의 오른쪽 클릭 메뉴에서 기능을 실행할 수 있습니다.

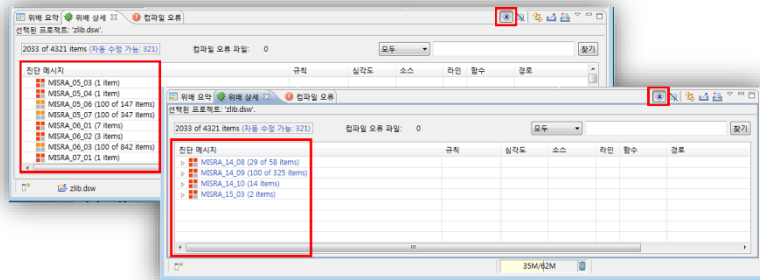


1. [테스트 네비게이터] 뷰에서 자동 수정한 프로젝트를 선택한 뒤, 오른쪽 클릭 메뉴에서 [위배 자동 수정 되돌리기]를 클릭합니다.
2. 마지막으로 백업된 파일 목록을 확인한 뒤, [예] 버튼을 클릭합니다.
※ 마지막으로 위배를 자동 수정한 이후 변경된 사항은 보존되지 않습니다.
3. 되돌리기가 끝나면, 결과 창이 나타납니다.



4. 되돌려진 소스 파일의 위배 정보는 인스펙션 후에 확인할 수 있습니다.

자동 수정 가능한 위배만 보기



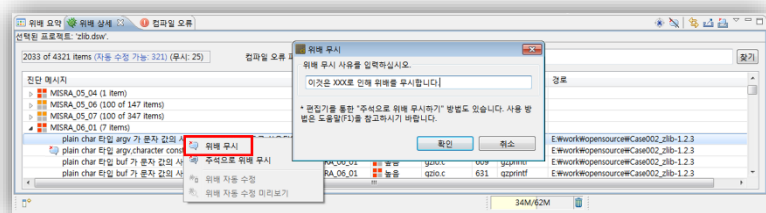
‘자동 수정 가능한 위배만 보기’ 필터 버튼으로 [위배 상세] 뷰에서 파란색으로 표시되는 자동 수정 가능한 위배사항 리스트만 볼 수 있습니다. 필터 버튼의 ON/OFF 상태는 기억되며 자동 수정 가능한 위배가 없을 경우, 위배사항 리스트에는 아무것도 표시되지 않습니다.


12. 위배 무시 기능

‘위배 무시’란 위배 항목을 무시하는 기능입니다.

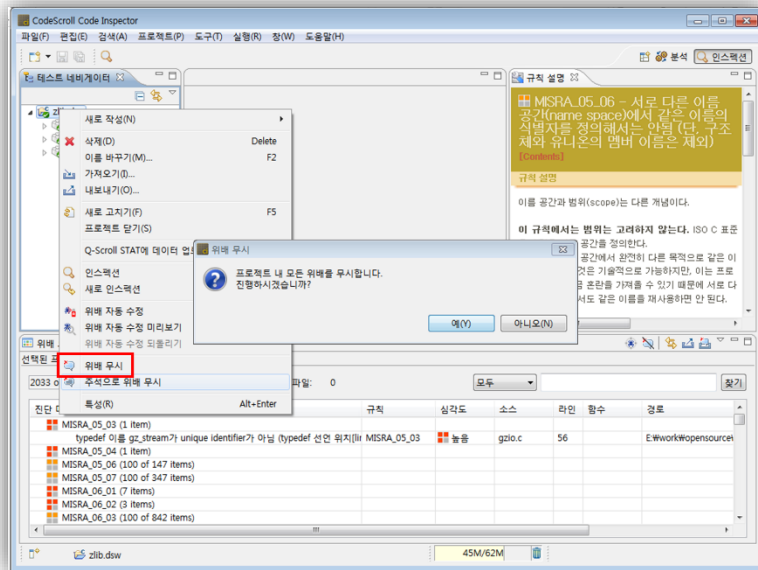
[위배 상세] 뷰와 [테스트 네비게이터] 뷰에서 각각 위배 무시, 주석으로 위배 무시 기능을 실행할 수 있습니다.

[위배 상세] 뷰에서 위배 무시하기

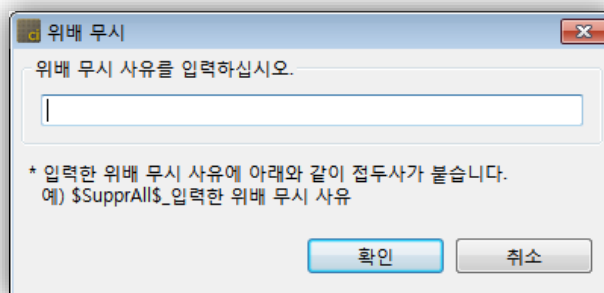



1. [위배 상세] 뷰에서 위배항목을 선택합니다.
2. 오른쪽 클릭 메뉴에서 [위배 무시]를 클릭합니다.
3. 위배 무시 사유를 입력하는 창이 열립니다.
4. 위배 무시 사유를 입력하고 [확인] 버튼을 클릭합니다.
 - ✓ 위배 무시를 적용한 위배 항목에  표시가 생깁니다.

[테스트 네비게이터] 뷰에서 프로젝트 전체 위배 무시하기



1. [테스트 네비게이터] 뷰에서 분석이 완료된 프로젝트를 선택한 뒤, 오른쪽 클릭 메뉴에서 [위배 무시]를 클릭합니다.
2. 프로젝트 내 모든 위배를 무시한다는 확인 창이 열립니다.
3. [예] 버튼을 누르면 위배 무시 사유를 입력하는 창이 열립니다.



4. 무시 사유를 입력한 뒤, [확인] 버튼을 누르면 프로젝트 내 모든 위배가 무시됩니다.
 ✓ 위배 무시를 적용한 위배 항목에  표시가 생깁니다.

위배 무시 유지

위배 무시 유지란 프로젝트의 분석 결과를 검토한 후 무시한 위배들을 재 분석 시에도 유지시켜주는 기능을 말합니다. 이러한 위배 무시 유지는 위배의 종류에 따라 다르게 동작합니다. 전역 위치에서 검출된 위배의 경우 해당 위배가 검출된 파일이 재분석 시에도 변경되지 않았을 때에만 유지됩니다. 이와 달리 어떤 위배가 전역 위치가 아닌 함수 내부에서 검출된 경우 해당 파일이 변경되었더라도 함수의 내용만 변경되지 않았다면 위배 무시가 유지됩니다. 자세한 예제는 아래와 같습니다. 전역 또는 함수의 내용에 변경이 없을 때에만 위배 무시가 유지되는 이유는 부록 A를 참고하십시오.

A. 위배 무시가 유지되는 경우

i. 전역 위치에 있는 위배의 경우

- 이전 분석 이후에 전체 파일의 내용이 변경되지 않았을 때

BEFORE	<pre>int x; /* violation: base type is used */ void function() { int x; /* violation: base type is used */ }</pre>
AFTER	<pre>int x; void function() { int x; }</pre>

ii. 함수 내부에 있는 위배의 경우

- 이전 분석 이후에 해당 위배를 포함하고 있는 함수의 내용이 변경되지 않았을 때
-

BEFORE	<pre> int x; /* violation: base type is used */ void function() { int x; /* violation: base type is used */ } </pre>
AFTER	<pre> int xy; /* violation: base type is used */ void function() { int x; } </pre>

B. 유지되지 않는 케이스들

i. 전역 위치에 있는 위배의 경우

- 이전 분석 이후에 전체 파일의 내용이 변경되었을 때

BEFORE	<pre> int x; /* violation: base type is used */ void function() { int x; /* violation: base type is used */ } </pre>
AFTER	<pre> int xy; /* violation: base type is used */ void function() { int x; } </pre>

ii. 함수 내부에 있는 위배의 경우

-
- 이전 분석 이후에 해당 위배를 포함하고 있는 함수의 내용이 변경되었을 때

BEFORE	<pre>int x; /* violation: base type is used */ void function() { int x; /* violation: base type is used */ }</pre>
AFTER	<pre>int xy; /* violation: base type is used */ void function() { int xy; /* violation: base type is used */ }</pre>

주석으로 위배 무시하기

주석으로 간단하게 파일 전체 또는 특정 범위, 특정 라인의 위배를 무시합니다. 소스 형상이 바뀌더라도 주석이 남아있으면 위배 무시가 유지되는 이점이 있습니다.

<기본 형태>

```
/* [CS_IGNORE=규칙모음 이름.규칙 이름] */
// [CS_IGNORE=규칙모음 이름.규칙 이름]
```

<다른 주석에 추가 형태>

```
/* do something [CS_IGNORE=규칙모음 이름.규칙 이름] */
// do something [CS_IGNORE=규칙모음 이름.규칙 이름]
```

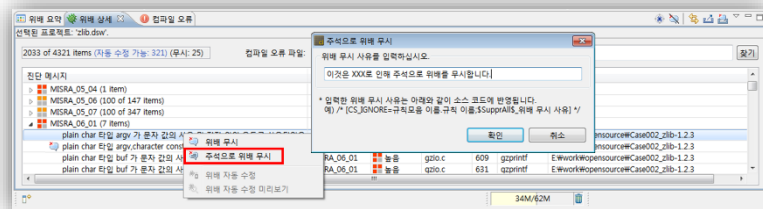
-
- ✓ 위배 무시 사유를 기록하기 위해서는 규칙 이름 뒤에 세미콜론(;)을 넣고 사유를 입력합니다.
ex)
/** [CS_IGNORE=규칙모음 이름.규칙 이름;위배 무시 사유] */
 - ✓ 별표 (*) 사용 가능
 - '*'는 임의 문자가 0번 이상 반복됨을 의미
 - [CS_IGNORE=MISRA_CPP.*] 는 MISRA_CPP 규칙모음의 모든 규칙 위배를 무시한다는 뜻
 - ✓ 주석이 여러 라인인 경우에, 주석이 시작되는 라인의 위배가 무시됩니다.


<파일 또는 특정 범위 위배 무시 형태>

```
/* [CS_FILE_IGNORE=규칙모음 이름.규칙 이름] */  
// [CS_FILE_IGNORE=규칙모음 이름.규칙 이름]  
/* [CS_FILE_IGNORE_END] */  
// [CS_FILE_IGNORE_END]
```

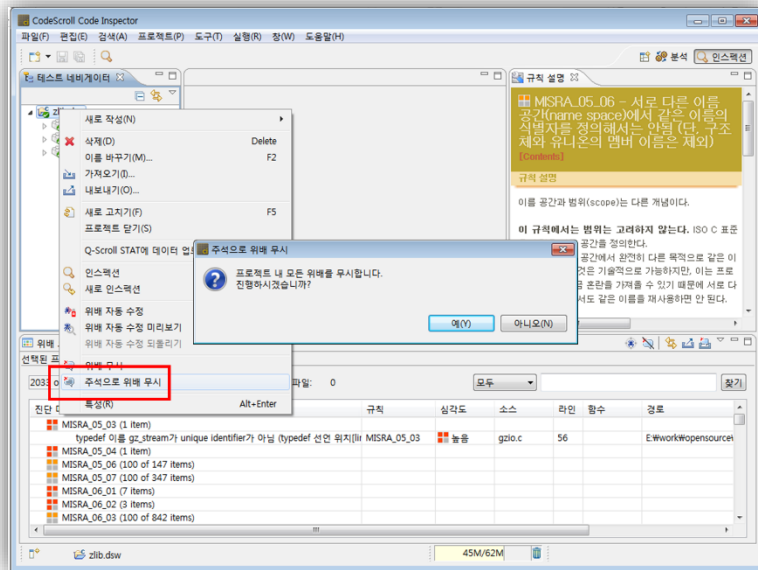
- ✓ 특정 범위를 위배 무시하기 위해서는 해당 범위가 시작되는 부분에 [CS_FILE_IGNORE=규칙모음 이름.규칙 이름]를 입력합니다.
- ✓ 해당 범위가 끝나는 부분에 [CS_FILE_IGNORE_END] 입력합니다.
- ✓ [CS_FILE_IGNORE_END]가 없으면 이 파일 전체의 위배를 무시합니다.

[위배 상세] 뷰에서 주석으로 위배 무시하기

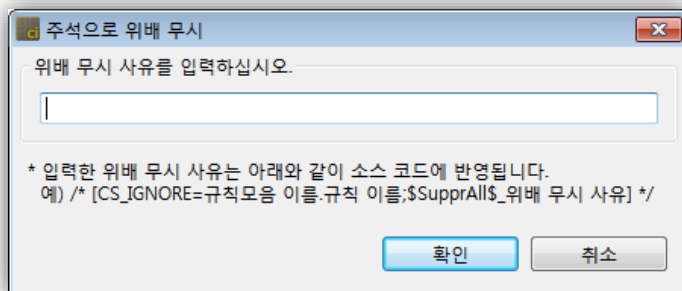



1. [위배 상세] 뷰에서 위배항목을 선택합니다.
 2. 오른쪽 클릭 메뉴에서 [주석으로 위배 무시]를 클릭합니다.
 3. 위배 무시 사유를 입력하는 창이 열립니다.
 4. 주석에 포함될 위배 무시 사유를 입력하고 [확인] 버튼을 클릭합니다.
 - ✓ 주석으로 위배 무시를 적용한 위배 항목에  표시가 생깁니다.
- ※ 주석으로 위배 무시가 적용된 라인에 동일한 규칙에 대한 위배가 추가 될 경우, 해당 위배도 같이 무시됩니다.
- ※ 위배 무시 주석이 추가되면서 소스 코드의 컬럼 길이가 늘어나 CODESCROLL_C.CSC_ETC_03 규칙에 대한 위배가 검출될 수 있습니다.

[테스트 네비게이터] 뷰에서 프로젝트 전체 주석으로 위배 무시하기



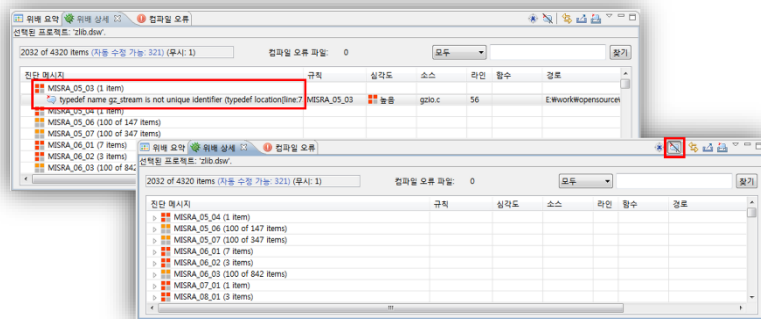
1. [테스트 네비게이터] 뷰에서 분석이 완료된 프로젝트를 선택한 뒤, 오른쪽 클릭 메뉴에서 [주석으로 위배 무시]를 클릭합니다.
2. 프로젝트 내 모든 위배를 무시한다는 확인 창이 열립니다.
3. [예] 버튼을 누르면 위배 무시 사유를 입력하는 창이 열립니다.



4. 주석에 포함될 위배 무시 사유를 입력한 뒤, [확인] 버튼을 누르면 프로젝트 내 모든 위배가 주석으로 무시됩니다.
 - ✓ 주석으로 위배 무시를 적용한 위배 항목에  표시가 생깁니다.

- ※ 주석으로 위배 무시가 적용된 라인에 동일한 규칙에 대한 위배가 추가 될 경우, 해당 위배도 같이 무시됩니다.
- ※ 위배 무시 주석이 추가되면서 소스 코드의 컬럼 길이가 늘어나 CODESCROLL_C.CSC_ETC_03 규칙에 대한 위배가 검출될 수 있습니다.

무시된 위배 제외하고 보기

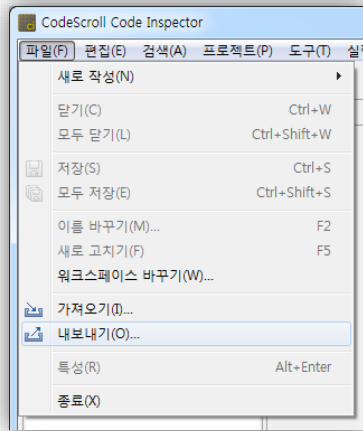


‘무시된 위배 제외하기’ 필터 버튼으로 [위배 상세] 뷰에서 무시되지 않은 위배사항 리스트만 볼 수 있습니다. 필터 버튼의 ON/OFF 상태는 기억됩니다.

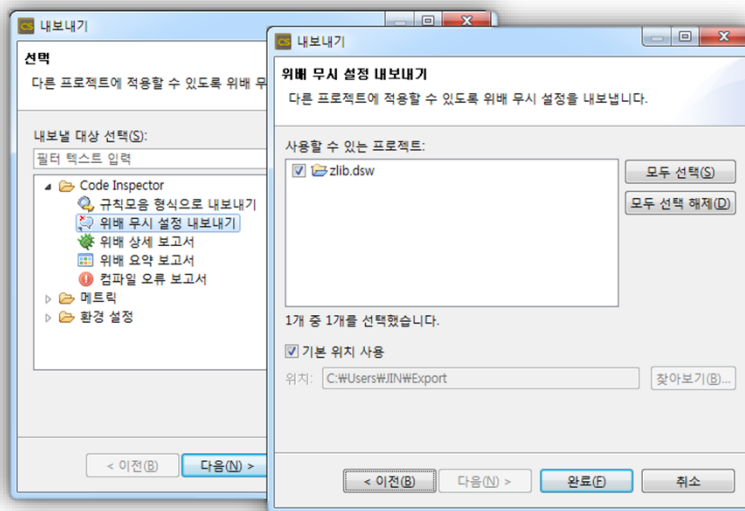
위배 무시 내보내기/가져오기

위배 무시 설정 내보내기/가져오기 기능을 통해 프로젝트 간의 위배 무시 항목들을 유지/관리 할 수 있습니다.

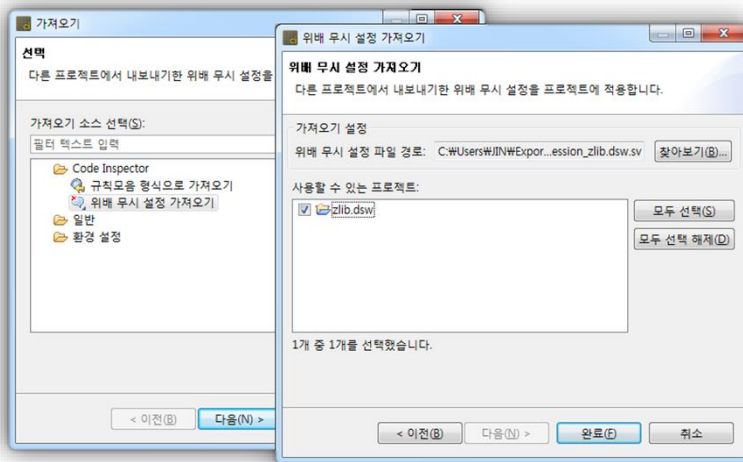
1. 메뉴의 [파일] -> [내보내기] 혹은 [가져오기]를 클릭합니다.



2. [내보내기] 혹은 [가져오기] 대화 상자에서 [위배 무시 설정 내보내기] 혹은 [위배 무시 설정 가져오기]를 선택한 후 [다음>] 버튼을 클릭합니다.
3. [내보내기]에서는 내보낼 프로젝트를 선택하고 내보낼 위치를 지정한 후 [완료] 버튼을 클릭하면 지정한 위치에 위배 무시 설정 파일을 내보냅니다.



-
4. [가져오기]에서는 위배 무시 설정 파일이 있는 경로를 지정하고 적용할 프로젝트를 선택한 후 [완료] 버튼을 클릭하면 위배 무시 설정 파일로부터 지정한 프로젝트에 위배 무시 설정을 적용합니다.

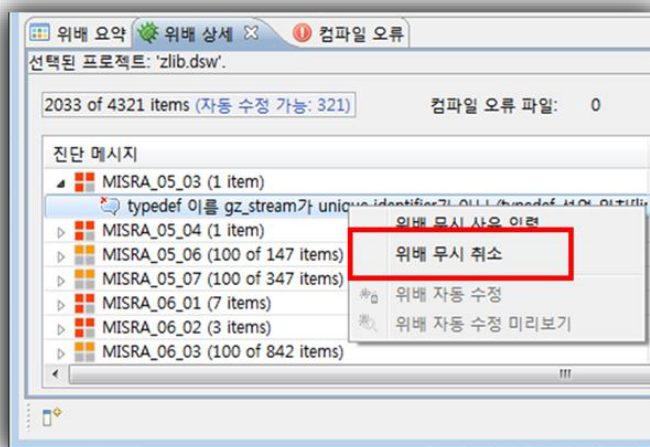


위배 무시 초기화

‘위배 무시 취소’ 또는 ‘위배 무시 초기화’ 기능을 통해 무시된 위배의 상태를 무시되지 않은 상태로 초기화할 수 있습니다.

다만 소스 코드의 위배 무시 주석은 제거되지 않습니다.

개별적으로 위배 초기화하기

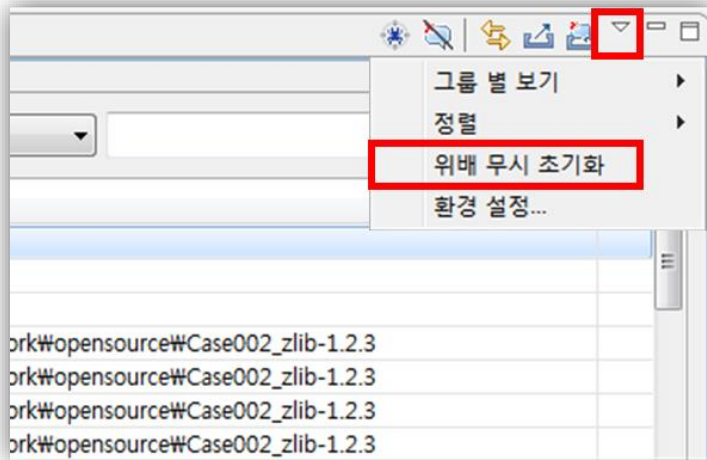



1. [위배 상세] 뷰에서 초기화할 위배항목을 선택합니다.
2. 오른쪽 클릭 메뉴에서 [위배 무시 취소]를 클릭합니다.

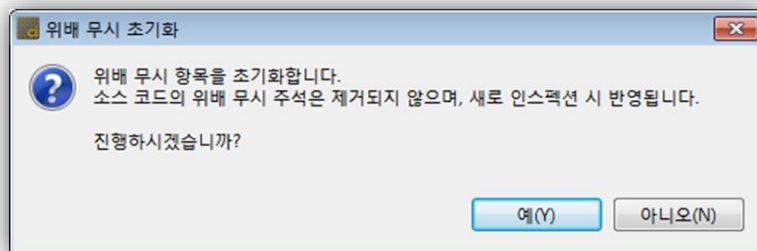
✓ 위배 항목의  표시가 사라집니다.

※ 소스 코드의 위배 무시 주석은 제거되지 않습니다.

프로젝트 전체 위배를 초기화하기



1. [위배 상세] 뷰 오른쪽 위에 위치한 [보기 메뉴]  를 클릭합니다.
2. [위배 무시 초기화]를 클릭합니다.
3. 위배 무시 항목을 초기화한다는 확인 창이 열립니다.



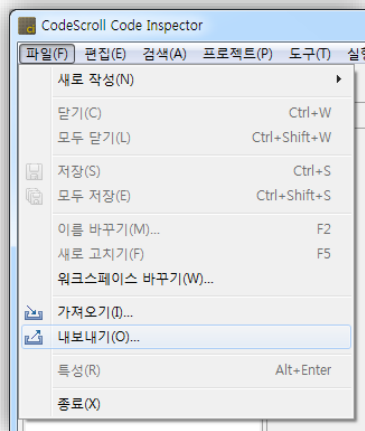
4. [예] 버튼을 누르면 프로젝트 내 모든 위배의 상태가 초기화됩니다.
※ 소스 코드의 위배 무시 주석은 제거되지 않습니다.

13. 인스펙션 결과의 문서화

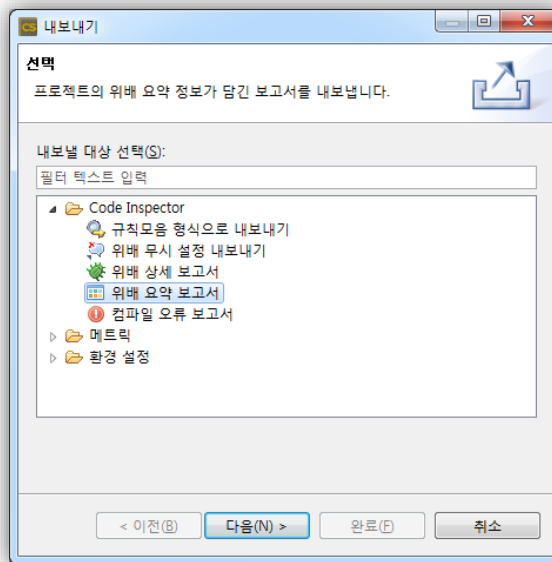
내보내기 기능을 통한 보고서 생성

인스펙션을 수행한 결과를 보고서 생성 기능을 통하여 다양한 형태의 문서로 생성할 수 있습니다.

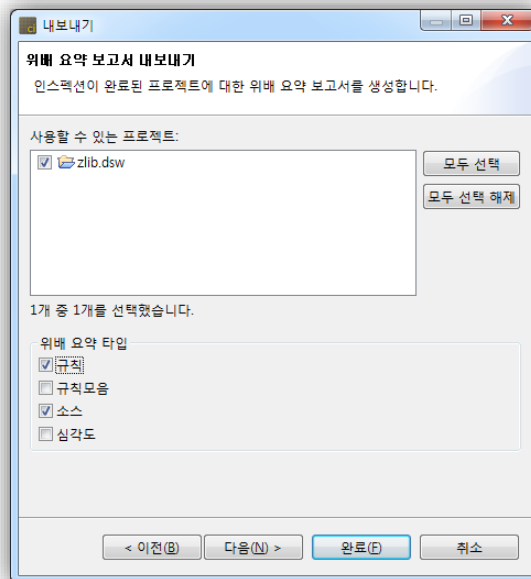
1. 메뉴의 [파일] -> [내보내기]를 클릭합니다.



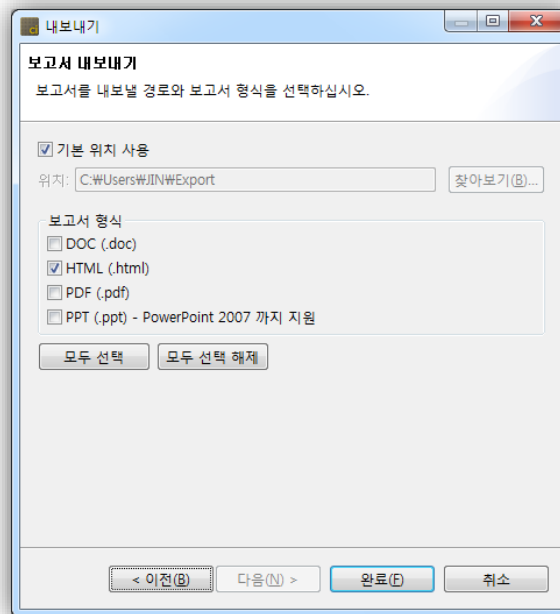
2. [내보내기] 대화 상자에서 생성할 보고서를 선택한 후 [다음>] 버튼을 클릭합니다.




3. 보고서를 생성할 프로젝트를 선택하고 타입이 다양한 보고서일 경우는 원하는 타입을 선택합니다.

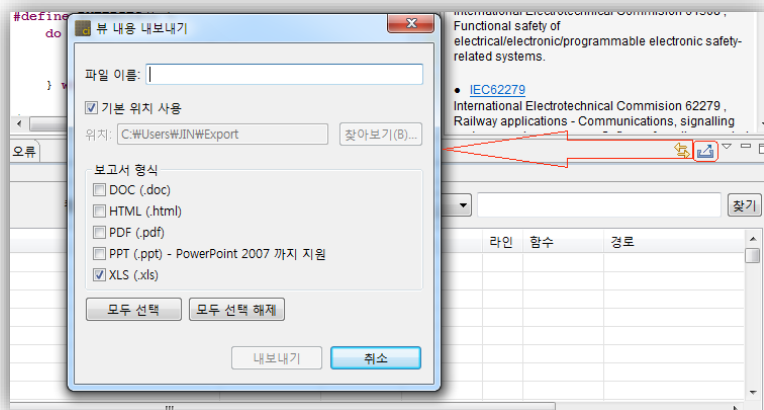


-
4. 생성할 문서 타입을 선택하여 [완료] 버튼을 클릭하면 지정한 위치에 보고서를 생성합니다.



뷰에서 내보내기

뷰에 있는  아이콘을 통해 현재 보이는 내용을 보고서로 생성할 수 있습니다.

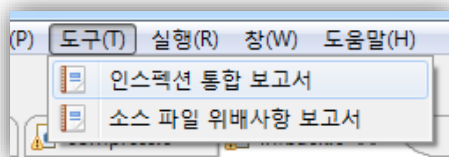


1. [뷰 내용 내보내기] 아이콘을 클릭합니다
2. [뷰 내용 내보내기] 대화 상자에서 생성할 보고서 형식을 선택하여 [내보내기] 버튼을 클릭하면 지정한 위치에 보고서가 생성됩니다.

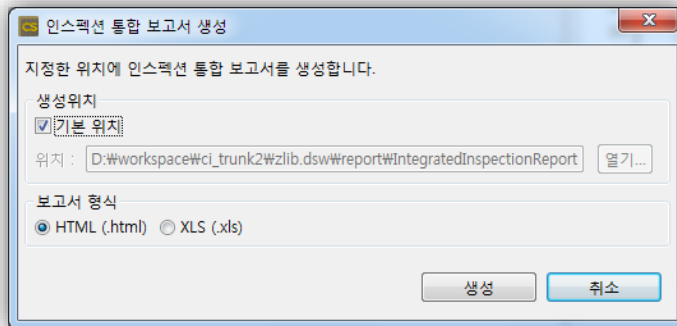
인스펙션 통합 보고서

인스펙션 요약 정보와 위배 상세 내용과 메트릭 정보를 HTML 또는 엑셀 (.xls) 형식으로 한번에 생성합니다.

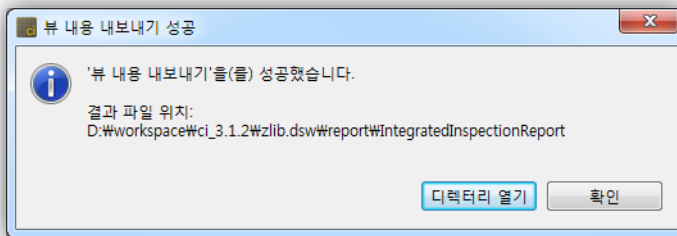
1. 메뉴의 [도구] -> [인스펙션 통합 보고서]를 선택합니다.



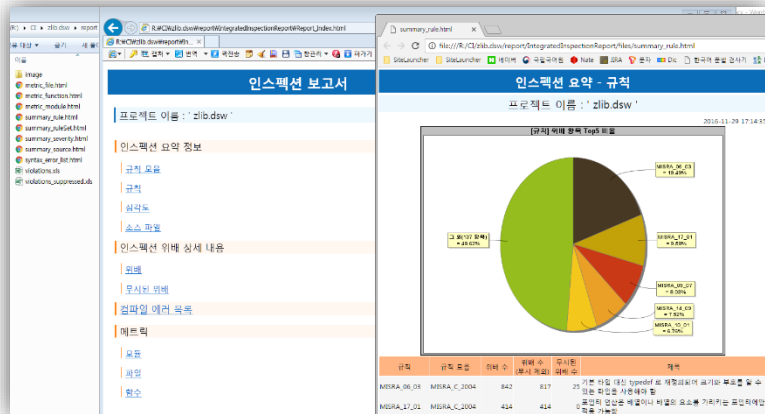
2. 생성 위치와 보고서 형식을 지정하고 [생성]을 클릭하면 지정한 위치에 통합 보고서를 생성합니다.



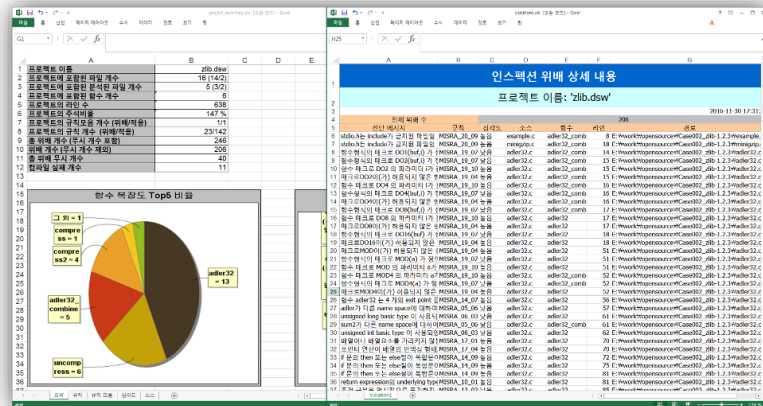
3. 생성이 완료되면 [디렉터리 열기]를 클릭하여 보고서가 생성된 디렉터리를 열어볼 수 있습니다.



- A. HTML 형식으로 생성하면 메인 페이지인 “Report_Index.html”을 통해 다음과 같이 인스펙션 결과를 확인할 수 있습니다.



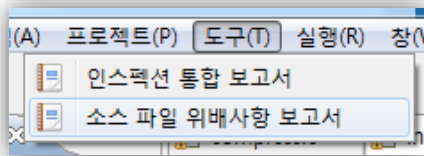
B. XLS 형식으로 생성하면 다음과 같이 인스펙션 결과를 확인할 수 있습니다.



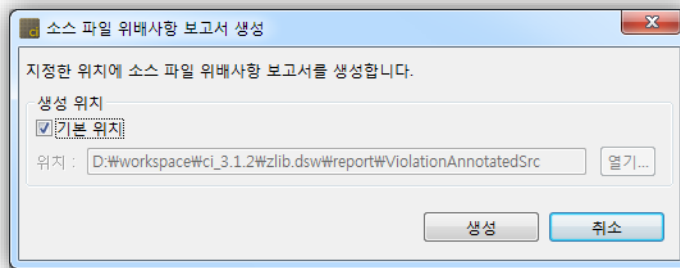
소스 파일 위배사항 보고서

소스 파일에서 위배가 발생한 위치에 위배 메시지가 삽입된 형태의 HTML 보고서를 제공합니다.

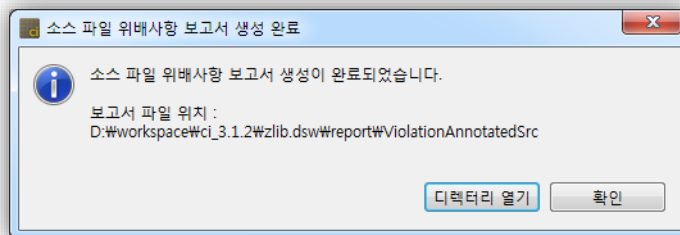
1. 메뉴의 [도구] -> [소스 파일 위배사항 보고서]를 선택합니다.



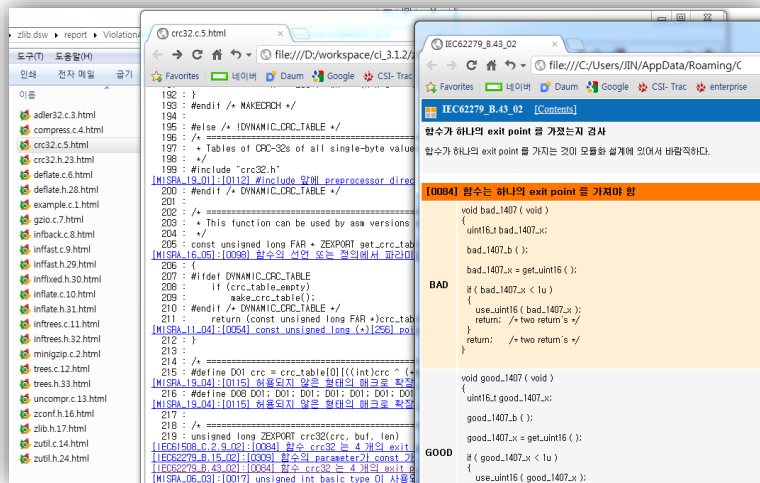
2. 생성 위치를 지정하고 [생성]을 클릭하면 지정한 위치에 소스 파일 위배사항 보고서를 생성합니다.



3. 생성이 완료되면 [디렉터리 열기]를 클릭하여 보고서 문서가 생성된 디렉터리를 열어볼 수 있습니다.



4. 소스 파일에 삽입된 위배 메시지를 클릭하면, 위배에 대한 규칙의 설명을 보여줍니다.

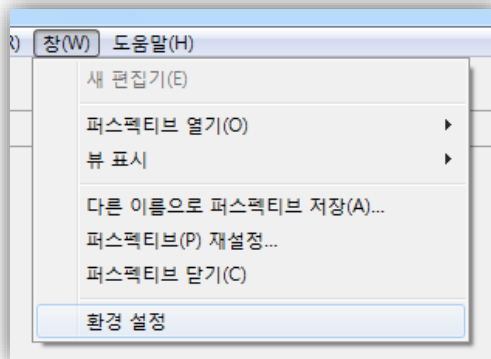


14. 규칙모음 설정

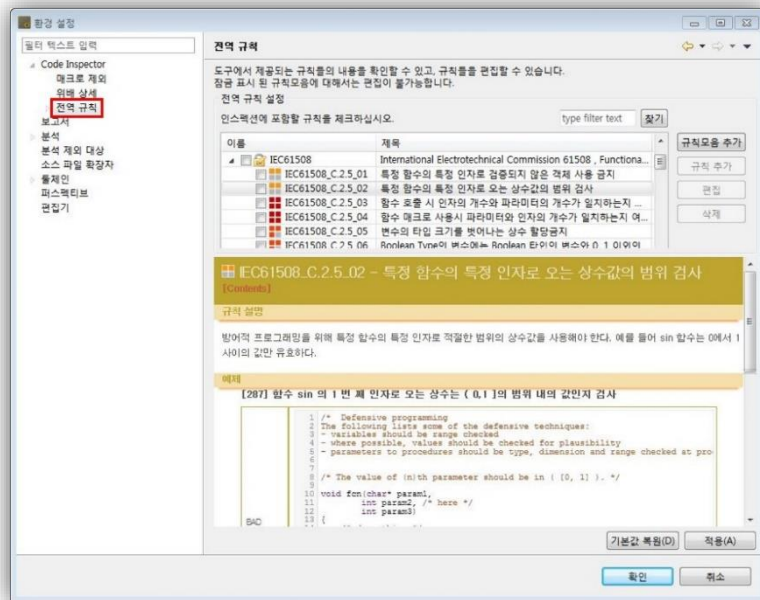
전역 규칙 설정


인스펙션에 사용되는 규칙모음과 규칙들을 확인하고, 기존 내용을 편집하고 삭제하거나 규칙을 새롭게 생성할 수 있습니다.

1. 메뉴의 [창] -> [환경 설정]을 클릭합니다.



2. [Code Inspector] -> [전역 규칙]을 선택하면 전역 규칙 설정 페이지가 나타납니다.

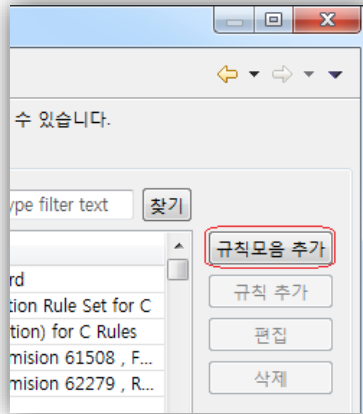


3. 규칙모음의 체크박스를 체크하면 앞으로 생성하는 프로젝트의 기본 규칙으로 지정합니다.
4. 규칙 이름 옆의 아이콘은 편집 가능 여부를 표시합니다. 자물쇠 표시 가 있으면 편집이 불가능합니다.
5. 규칙모음을 클릭하면 해당 규칙모음에 포함된 규칙들을 확인할 수 있고 각 규칙의 설명을 볼 수 있습니다.
6. [규칙모음 추가], [규칙 추가], [편집], [삭제] 버튼을 클릭하여 새로운 규칙모음을 추가하거나 선택한 규칙모음과 규칙을 추가/삭제/편집할 수 있습니다.
7. [기본값 복원] 버튼을 클릭하면 제품 기본 규칙들로 초기화됩니다. 초기화 과정 시, 사용자가 추가/편집한 규칙이 삭제되므로 사용자 규칙의 백업을 권장합니다. [규칙모음 내보내기] 기능을 통해 규칙을 백업할 수 있습니다.

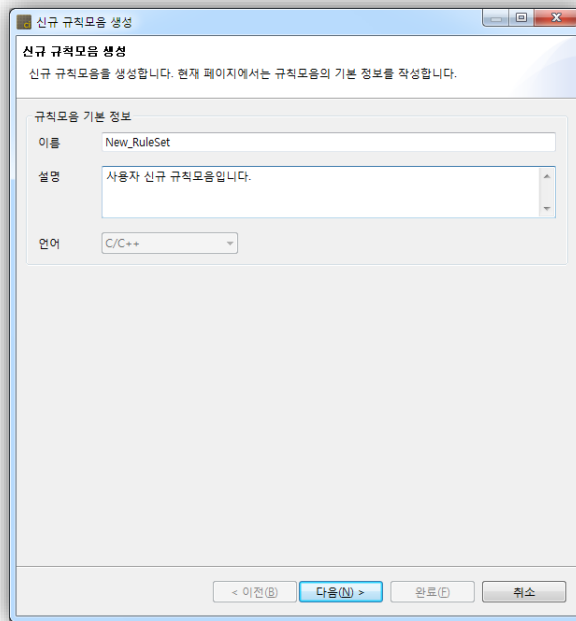
신규 규칙모음 생성

사용자가 원하는 특정 규칙만으로 새로운 규칙모음을 생성할 수 있습니다.

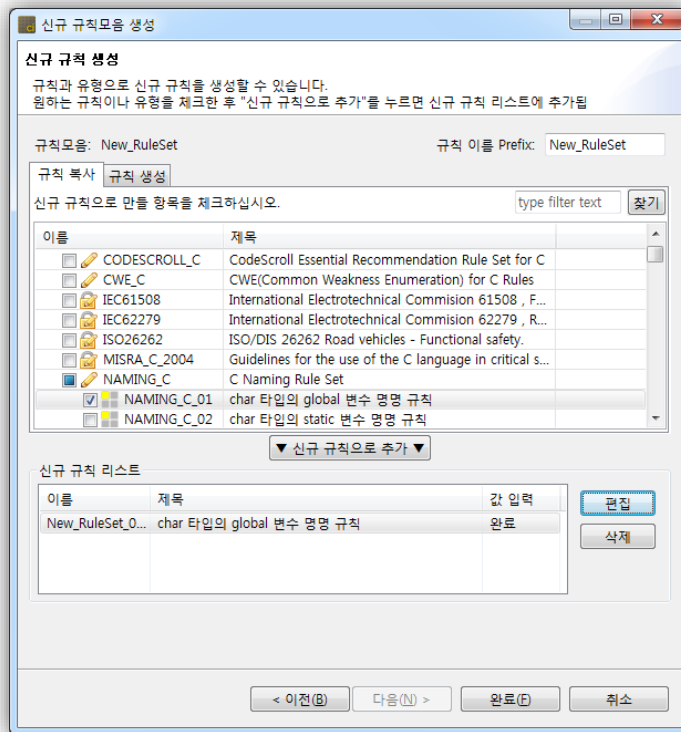
1. [전역 규칙] 페이지의 [규칙모음 추가] 버튼을 클릭합니다.



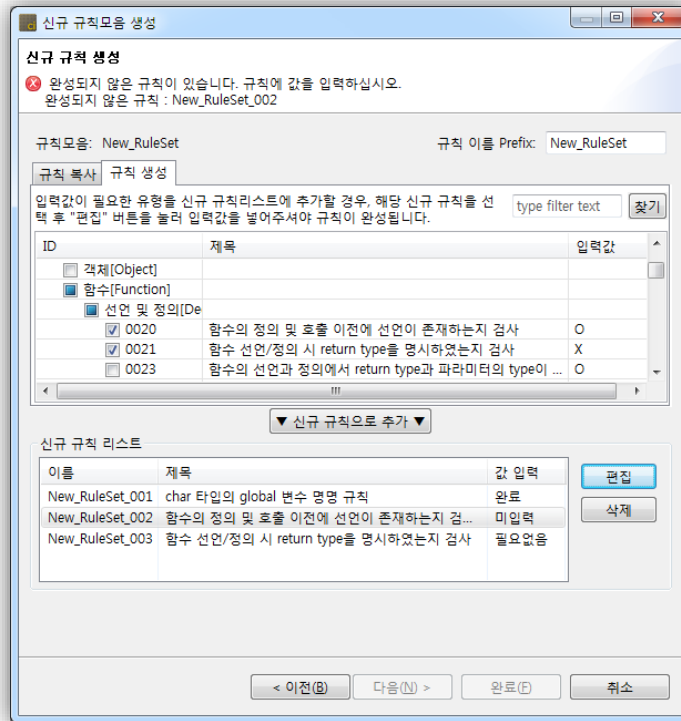
2. [이름], [설명], [언어]를 입력하고 [다음>] 버튼을 클릭합니다.



3. [규칙 복사] 탭을 선택하면 기존 규칙모음으로부터 복사하여 생성할 수 있습니다. 기존의 규칙모음에서 구체적인 내용을 변경하지 않고 그대로 추가합니다.



- A. 기존 규칙모음이나 규칙을 체크하고, 대화 상자 가운데에 위치한 [▼신규 규칙으로 추가▼] 버튼을 클릭합니다.
 - B. 대화 상자 하단에 위치한 [신규 규칙 리스트]에 체크한 규칙이 추가됩니다. [값 입력] 항목에 값이 필요한 규칙은 값을 수정할 수 있습니다.
 - C. 모든 입력이 완료 되면 [완료]를 클릭하여 규칙을 생성합니다.
4. [규칙 생성] 탭을 선택하면 Code Inspector에서 제공하는 템플릿을 이용하여 새롭게 규칙을 생성할 수 있습니다.



- A. 유형에 맞는 규칙을 체크하고 [▼신규 규칙으로 추가▼] 버튼을 클릭합니다.
- B. 입력 값이 필요한 규칙에 값을 입력할 때는 [편집] 버튼을 클릭하여 값을 넣어줍니다.
 - ✓ 값이 필요한 규칙에 값이 입력되지 않으면 [값 입력]란에 “미입력”이라는 표시가 나타나고 규칙 생성을 완료할 수 없습니다.
- C. 모든 입력이 완료 되면 [완료]를 클릭하여 규칙을 생성합니다.

- [규칙 복사] 탭을 통해 신규 규칙 리스트에 규칙이 추가될 때는 다음과 같은 방식으로 규칙 이름이 만들어집니다.

신규 규칙 = 규칙 이름 Prefix + 기존 규칙 이름

“규칙 이름 Prefix + 001” 과 같은 형태로 규칙 이름을 만들고 싶을 경우, 환경설정 값을 입력합니다.

<환경설정 값 입력 방법>

₩[workspace]₩.metadata₩.plugins₩org.eclipse.core.runtime₩.settings
에서 com.codescroll.ci.prefs 파일을 엽니다. (파일이 존재하지 않을 경우, 텍
스트 파일을 생성하고 이름을 com.codescroll.ci.prefs로 변경합니다.)

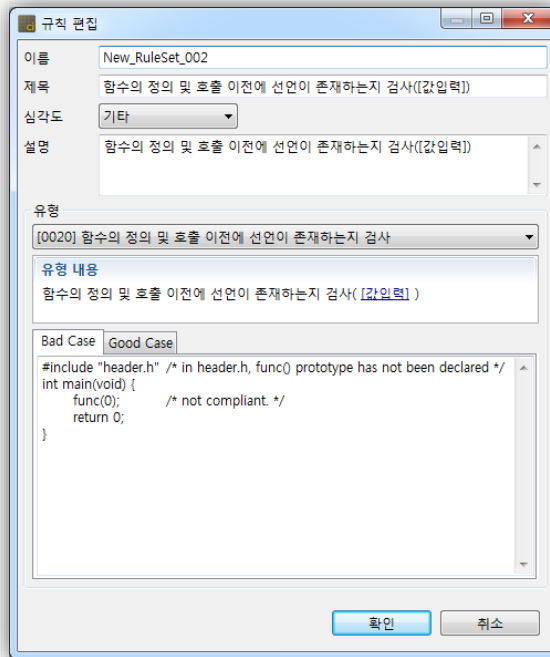
위 파일에 아래 값을 입력합니다.

```
rule.name.create.type.target.name=false
```

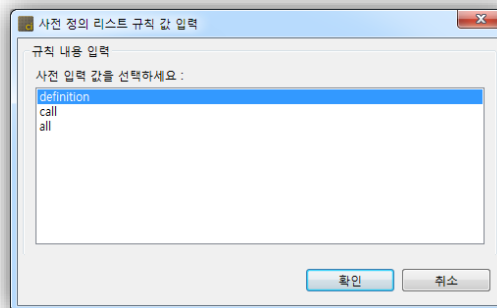
규칙 편집

규칙 편집 기능으로 선택한 규칙의 내용을 변경하거나 값의 입력이 필요한
규칙일 경우는 값을 입력/변경할 수 있습니다.

1. 규칙을 선택하고 [편집]을 클릭합니다.
2. [규칙 편집] 대화 상자에서 내용을 편집합니다. 값의 입력이 필요한 유
형을 갖는 경우에는 [유형내용]란의 “값입력”을 클릭합니다.



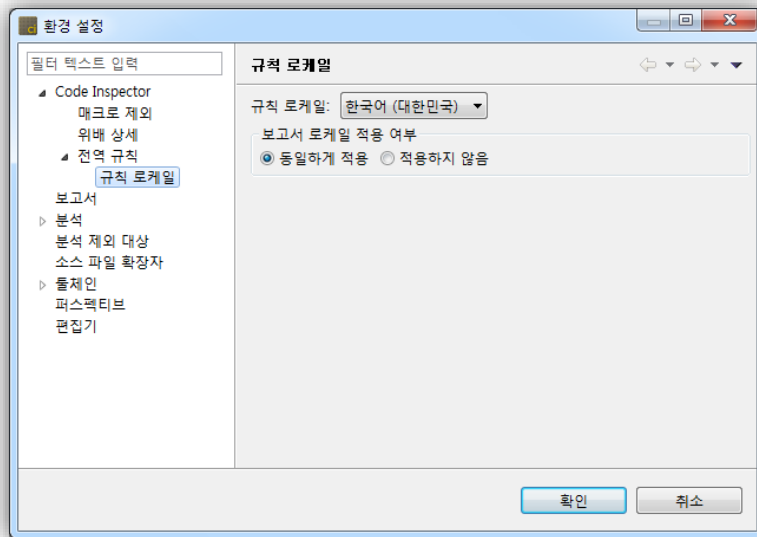
3. [값입력]을 클릭하여 나타나는 대화 상자에서 구체적인 값을 지정합니다.



규칙 로케일 변경

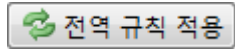
규칙 내용과 위배 결과물을 다국어로 표현하고 싶을 경우, 규칙 로케일 변경 기능을 통해 전역 규칙의 로케일을 변경합니다.

1. 메뉴의 [창] -> [환경 설정] -> [Code Inspector] -> [전역 규칙] -> [규칙 로케일]을 선택하면 규칙 로케일 설정 페이지가 나타납니다.



2. [규칙 로케일] 드롭다운 목록을 이용하여 언어를 변경하면 선택한 언어에 맞게 전역 규칙 로케일이 변경됩니다.
 3. 변경 상태에 따라 보고서 로케일도 동일한 로케일로 적용하고 싶을 경우, [보고서 로케일 적용 여부]를 [동일하게 적용]으로 설정합니다.
- 전역 규칙 데이터는 로케일 별로 관리가 됩니다. 한국어 로케일 상태에서 편집된 규칙은 한국어 규칙 데이터에만 반영됩니다.
 - 로케일 변경 후에 생성되는 프로젝트에 대해 로케일이 적용됩니다.
 - 기존 프로젝트에 적용하기 위해서는 테스트 네비게이터 뷰에서 프로젝트의 오른쪽 클릭 메뉴에서 [특성]을 선택, [Code Inspector] -> [규칙

설정] 페이지가 나오면, [전역 규칙 적용]



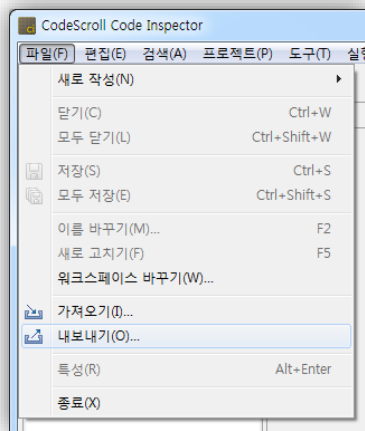
버튼을

클릭하여 프로젝트에 로케일이 변경된 전역 규칙을 적용합니다.

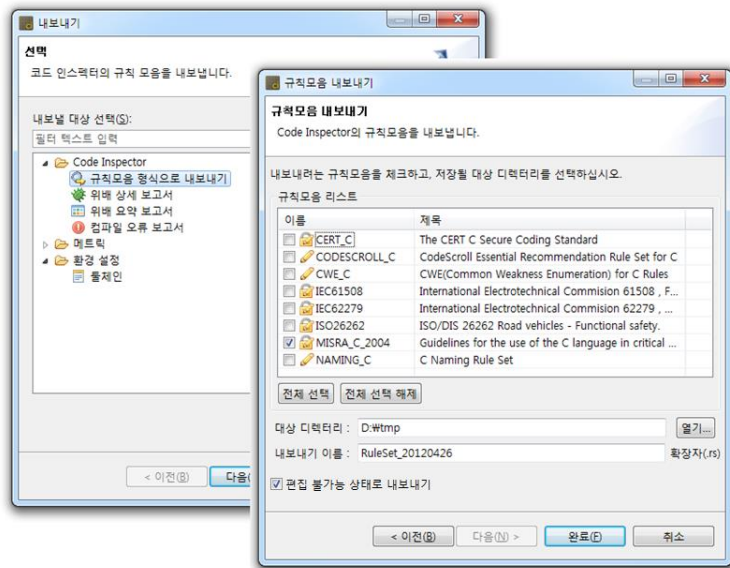
규칙모음 내보내기

‘내보내기’ 기능을 통해 규칙모음을 내보낼 수 있습니다.

1. 메뉴의 [파일] -> [내보내기]를 선택합니다.



2. [Code Inspector] -> [규칙모음 형식으로 내보내기]를 선택하고 [다음>]을 클릭합니다.

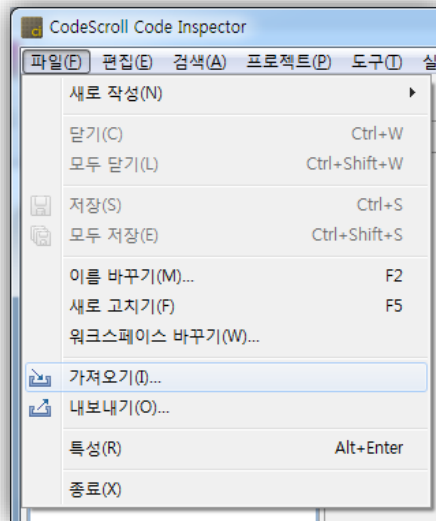


3. 내보낼 규칙모음을 체크하고 내보낼 경로를 지정합니다.
 - A. 편집 여부 상태를 유지하려면 [편집 불가능 상태로 내보내기] 체크박스를 해제하기 바랍니다.
4. 입력을 완료하고 [완료] 버튼을 클릭하면 규칙 파일을 생성합니다.

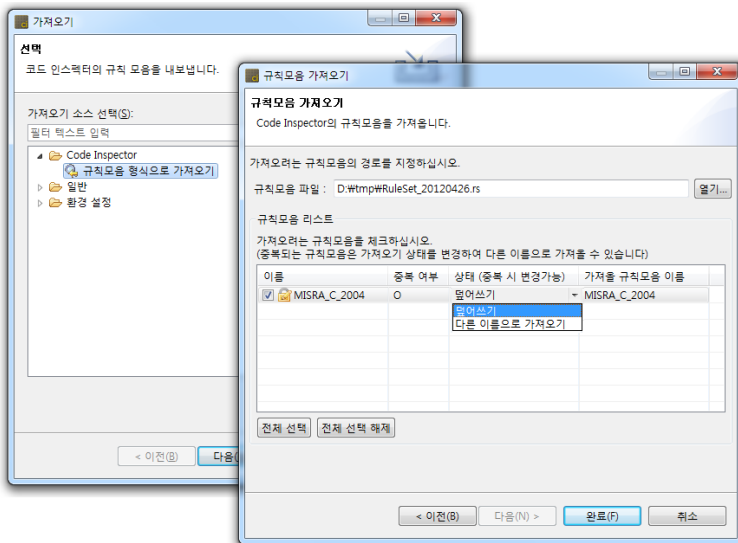
규칙모음 가져오기

‘가져오기’ 기능을 통해 내보내기 한 규칙모음을 가져올 수 있습니다.

1. 메뉴의 [파일] -> [가져오기]를 선택합니다.



2. [Code Inspector] -> [규칙모음 형식으로 가져오기]를 선택하고 [다음]을 클릭합니다.



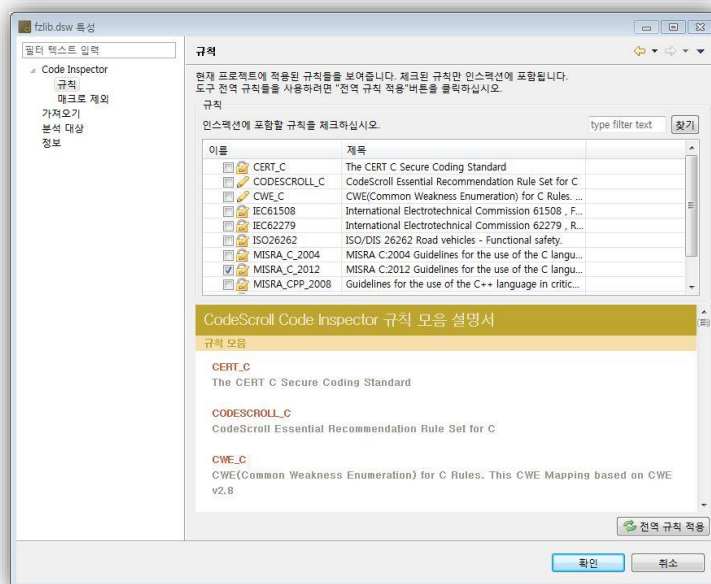
3. 가져올 규칙모음 파일을 지정하고 가져올 규칙모음을 체크합니다.
4. 기존 규칙모음과 중복되면 덮어쓰거나 다른 이름으로 가져올 수 있습니다.


-
5. 입력을 완료하고 [완료] 버튼을 클릭하면 규칙 파일을 가져옵니다.

프로젝트에 도구 전역 규칙 적용하기

기존에 생성했던 프로젝트에 도구 설정에서 추가/편집한 규칙을 적용하기 위해서는 [전역 규칙 적용] 기능을 사용해야 합니다.

1. 프로젝트를 선택하고 [파일] -> [특성]을 선택하거나, 선택한 프로젝트의 오른쪽 클릭 메뉴에서 [특성]을 선택합니다.
2. 프로젝트 특성 창의 [Code Inspector] -> [규칙]을 선택하면 규칙 설정 페이지가 나타납니다.



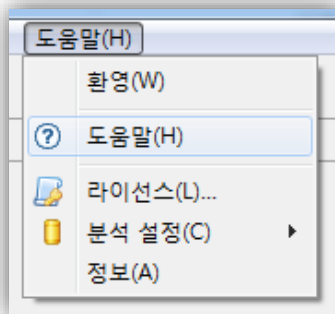
3. [전역 규칙 적용]  **전역 규칙 적용** 버튼을 클릭하여 프로젝트에 전역 규칙을 적용합니다.

15. 도움말

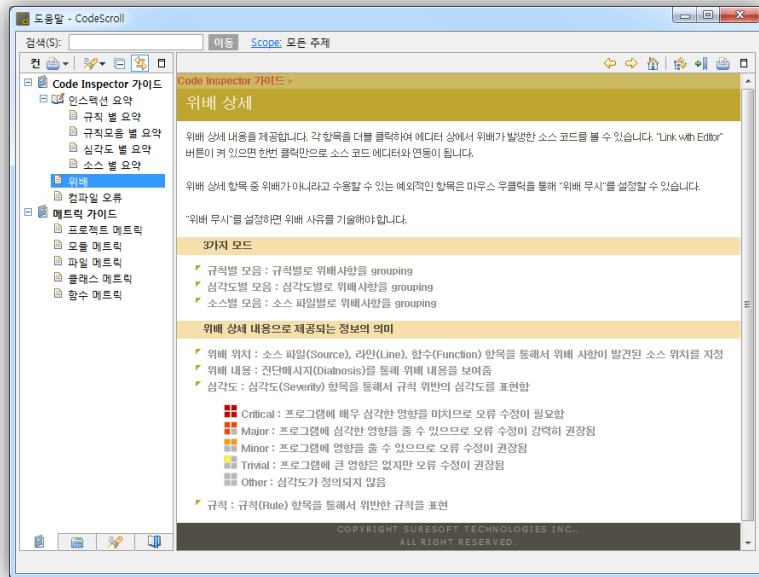
도움말 메뉴

도움말에는 각 뷰의 구성요소들이 설명되어 있습니다.

1. 메뉴의 [도움말] -> [도움말]을 선택합니다.



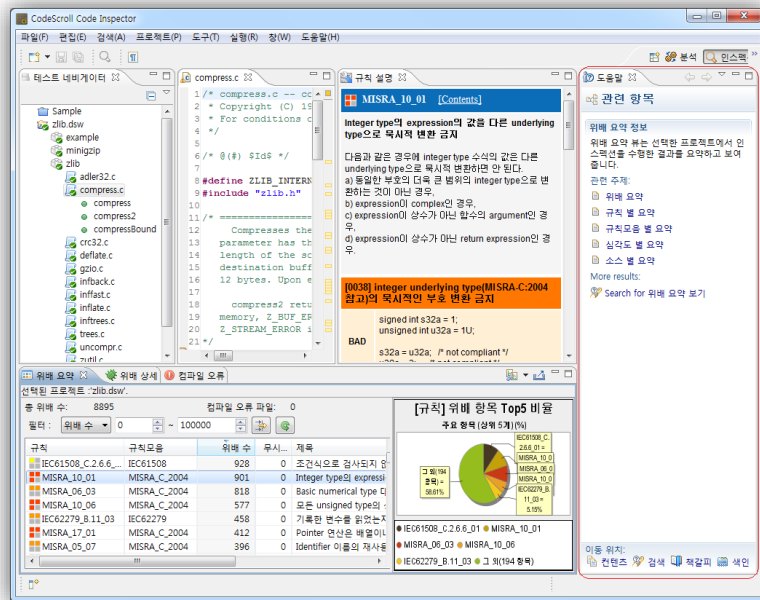
2. 원하는 도움말 페이지를 확인합니다.



뷰 도움말

도움말 버튼(F1)을 통해 각 뷰에서 해당 도움말로 쉽게 접근할 수 있습니다.

1. 뷰를 선택한 상태에서 F1 키를 누릅니다.



2. 원하는 도움말 페이지를 확인합니다.

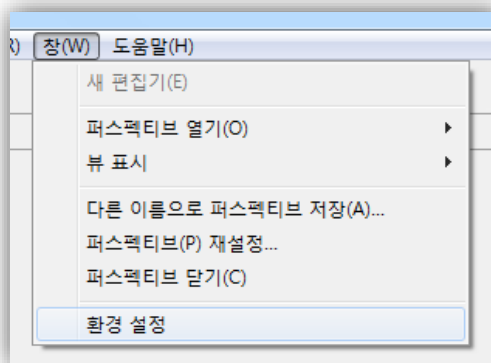


16. 환경 설정

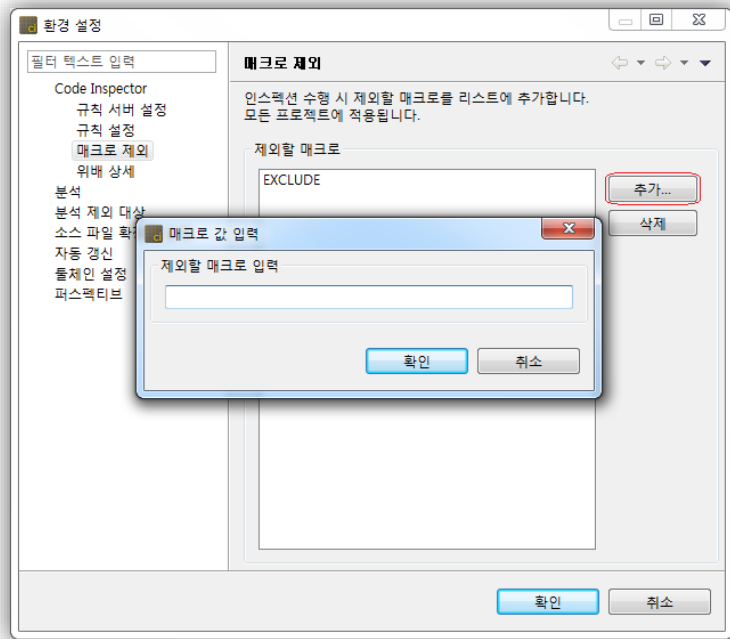
매크로 제외

인스펙션 수행 시 위배 사항에서 제외할 매크로를 설정할 수 있습니다.
해당 기능은 매크로 제외 기능이 포함된 규칙이 지원되어야 사용이 가능합니다. 사용을 원하시는 경우, 본 문서 2페이지에 있는 기술지원 연락처를 통해 문의해주시기 바랍니다.

1. 메뉴의 [창] -> [환경 설정]을 클릭합니다.



2. [Code Inspector] -> [매크로 제외]를 선택하면 매크로 제외 페이지가 나타납니다.



프로젝트 별로 설정하기 위해서는 프로젝트의 [특성]에서 설정할 수 있습니다.

17. 문제해결

설치 시 보안프로그램과의 충돌

설치 과정에서 오류가 발생한다면, 일시적으로 보안프로그램 동작을 중지하고 설치를 다시 진행해주시오.

제품 실행 시 작업공간 지정 후 제품이 구동되지 않는 경우

제품 최초 실행 시 CodeScroll의 전역 데이터가 %APPDATA% 경로에 저장되고 있습니다. 이 경로에 대한 접근에 제약이 있을 경우 제품이 구동되지 않을 수 있습니다.

이를 해결하기 위한 방법으로 전역 데이터 경로를 설정할 수 있는 기능을 사용하시기 바랍니다.

설정 방법: CodeScroll.ini 파일의 -g default(기본값) 대신에 설정할 전역 데이터 경로를 입력합니다(-g 또는 -global, csc.ini 파일도 동일)

-g 옵션이 주어지지 않으면, 제품 최초 실행 시에 임의의 전역 데이터 경로를 설정하는 창이 나타납니다. 이때 취소하면 기본 경로로 설정됩니다.

규칙 설정 변경 내용이 적용되지 않을 경우

전역 규칙 설정에서 규칙을 추가/변경할 경우, 이미 생성된 프로젝트에는 변경 내용이 바로 적용되지 않습니다.

“규칙모음 설정” 창의 “프로젝트에 도구 전역 규칙 적용하기”를 통해 프로젝트에 변경된 규칙들을 적용할 수 있습니다.

일부 규칙 설명이 보이지 않을 경우

Code Inspector의 “규칙 설명 뷰”는 사용자의 Internet Explorer를 이용하여 규칙설명을 보여주고 있습니다.

Internet Explorer의 인코딩 자동 선택 기능이 불안정할 경우, 일부 규칙 설명이 보이지 않을 수 있습니다.

보이지 않는 규칙 설명을 보기 위해선 Internet Explorer를 실행하여 [보기] -> [인코딩]에서 UTF-8을 선택하고, “자동 선택” 옵션을 꺼주신 후에 제품을 다시 실행하시면 규칙 설명을 보실 수 있습니다.

규칙 유형 290번을 갖는 규칙을 사용하는 방법

규칙 유형 290번을 갖는 규칙을 사용하기 위해서는 사전에 사용자가 설정할 내용이 있습니다.

1. 메뉴의 [창] -> [환경 설정]을 클릭하여 [툴체인 설정] 페이지로 이동합니다.
2. 사용하는 툴체인을 선택하고 [편집] 버튼을 클릭하여 [툴체인 편집] 대화 상자로 이동합니다.
3. [다음] 버튼을 클릭하여 툴체인 세부사항 설정 단계로 넘어갑니다.
4. [기타] 탭을 선택하고 [Effectively boolean으로 사용하는 타입]에 [추가] 버튼을 클릭합니다.
5. [Effectively boolean으로 사용하는 타입] 항목 입력란에 인스펙션 대상 소스 코드에서 사용하는 Boolean 타입의 이름을 입력합니다.
Boolean 데이터 타입을 사용하기 위해, 타입 정의를 별도로 한 이름을 입력합니다.

예) `typedef unsigned char BOOL;`
‘BOOL’을 입력합니다.

- ✓ Code Inspector에서 기본으로 제공하는 규칙들 중에 290번을 갖는 규칙은 CODESCROLL_C 규칙모음의 CSC_TYPE_06 규칙과

위배 요약 파이차트에서 한글이 깨질 경우

특정 환경에서 자바 기본 폰트인 Sanserif를 지원하지 않을 경우, 파이 차트의 한글이 깨집니다. 다음과 같이 파이차트 폰트를 변경하십시오.

〈파이차트 폰트 변경 방법〉

1. 프로그램 설치경로에 있는 “plugin_customization.ini” 파일을 엽니다.
2. “com.codescroll.ci/chart.font.name” 키를 찾아서 지원 가능한 폰트를 값으로 입력합니다.

ex) com.codescroll.ci/chart.font.name=Dotum

〈CLI 환경에서 적용하는 방법〉

1. `%[workspace]%\metadata%\plugins\org.eclipse.core.runtime\%settings` 에서 com.codescroll.ci.prefs 파일을 엽니다.
2. “chart.font.name” 키를 찾아서 지원 가능한 폰트를 값으로 입력합니다.

ex) chart.font.name=Dotum

Windows Vista/7 에서 설치 및 실행

설치 및 실행 Windows 계정이 동일해야 하며, 반드시 관리자 계정으로 설치/실행해야 합니다.

UAC를 비활성화하지 않고 사용하려면, Program Files 이하에 설치하지 않도록 설치 과정에서 디렉터리를 변경하십시오

Command Line Interface 기능 사용 시

- ✓ 사용자 계정 컨트롤 설정(UAC)이 켜져 있는 상태에서 csc.exe를 수행하려면, 명령어를 입력할 Windows 명령 처리기(cmd.exe)도 관리자 권한으로 실행해야 합니다.

-
- ✓ 운영체제의 사용자 계정 컨트롤 설정(UAC)에서 사용자 계정 컨트롤을 끄고 나서 운영체제를 다시 시작하면, Windows 명령 처리기(cmd.exe)가 항상 관리자 권한으로 실행됩니다.

프로그램 분석 실패 또는 Compile(전처리) 오류

인스펙션 수행이 불가능한 소스 코드 상태입니다. 컴파일 플래그와 헤더파일(Include) 경로 설정을 확인해주시오.

설정 내용 확인 방법은 메뉴의 [환경 설정] -> [툴체인 설정]에서 가능합니다.

또한, 프로젝트 생성 시에 지정한 분석 설정이 현재 인스펙션 대상 소스 코드의 개발 환경과 일치하는지 확인해주시오.

만약, 모든 설정 내용이 정상일 경우(실제 개발 환경(IDE 등)에서 문제 없이 Compile/Build가 가능할 경우)에도 오류가 발생하면, 슈어소프트테크(주)로 문의바랍니다.

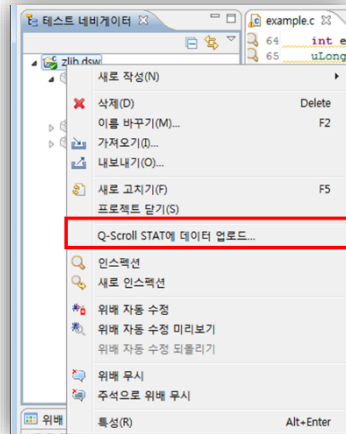
프로젝트 초기화 시 유지되는 정보

프로젝트 초기화가 수행되어도 사용자 입력에 의한 프로젝트 설정은 유지됩니다.

사용자 입력에 의한 프로젝트 설정들은 다음과 같습니다.

- 컴파일 플래그 설정
- 분석 제외 설정
- 규칙 설정
- 위배 무시 설정

Q-Scroll STAT에 데이터 업로드



프로젝트에 선택 후 오른쪽 클릭 메뉴에 “Q-Scroll STAT에 데이터 업로드” 기능이 있습니다. 이 기능에 대해서는 기술지원 연락처를 통해 문의해주시기 바랍니다.

기술지원

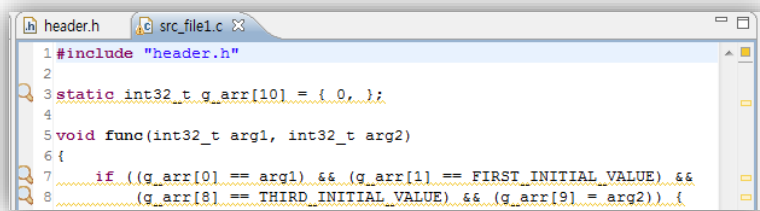
문제 발견 시 본 문서 2페이지에 있는 기술지원 연락처를 통해 문의해주시기 바랍니다.

18. 부록 A - 위배 무시 유지 기능

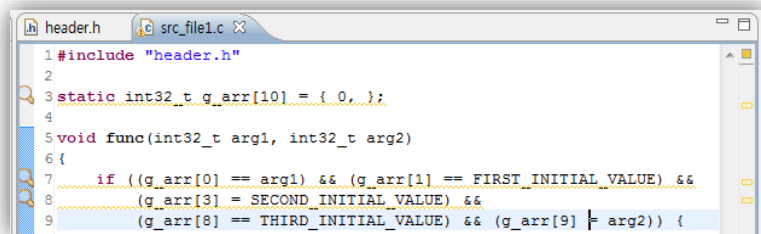
이 장에서는 함수 내에서 검출된 위배는 함수가 변경되지 않았을 때, 전역 위치에서 검출된 위배는 소스파일의 내용에 변경이 없을 때만 위배 무시가 유지되는 이유에 대해서 설명합니다. 여타 다른 도구와 같이 함수/소스 내용의 변경에도 위배 무시를 유지하는 경우 아래와 같은 문제가 발생할 수 있습니다.

소스 수정으로 인해 의도하지 않았던 위배를 무시하는 문제

타 도구의 위배 무시 유지 기능은 단순히 위배가 발생한 라인만을 이용하여 동작합니다. 따라서 소스 내용이 변경되어 원래 검토자가 의도하였던 위배가 아닌 새로운 위배가 발생하였으나 기존에 작업하였던 위배 무시 기능으로 해당 위배를 볼 수 없는 상황이 발생합니다. 아래의 예제를 참조해 주십시오.



처음 분석 시 8번 라인에서 발생한 위배를 무시 처리 하였다고 가정합니다. 그 이후 수정된 소스에 대해 다시 검사를 수행하면,



이렇듯 새로 추가된 8번 라인에 대한 새로운 위배가 검출되어 사용자에게 보고되어야 합니다. 그러나 단순히 라인 정보만을 이용하여 위배 무시를 유지하는 경우 이런 위배는 사용자의 확인 없이 무시됩니다. 저희 제품에서는 이러한 문제점을 막기 위하여 위배 무시 유지 시 사용자의 소스에 변경 여부를 확인합니다.

소스 수정으로 인해 유효하지 않은 위배 무시를 유지하는 문제

마찬가지로 기존에 발생한 위배를 검토하고, 무시한 이후에 소스가 변경되었을 때, 해당 변경 내역 때문에 기존에 무시한 위배를 재검토해야 하는 경우가 있습니다. 아래의 예제를 참조해 주십시오.

```
void check_status(enum CD_PLAYER_STATUS status)
{
    switch (status) {
        case CD_RESUME:
            check_current_cd_player_status();
            /* MISRA_15_02 switch case가 break로 끝나지 않음 위배 발생 */
            /* 의도한 fall through, 위배 무시 처리함 */
        case CD_PLAY:
            play();
            break;
        default:
            /* do nothing */
    }
}
```

```
        break;
    }
}
```

위의 코드에서는 CD_RESUME case문에서 fall-through를 의도하고 구현하였기 때문에 검출된 위배를 무시하였습니다.

```
void check_status(enum CD_PLAYER_STATUS status)
{
    switch (status) {
        case CD_PAUSE:
            /* pause 일 때도 resume일 때처럼 현재 상태 점검 */
        case CD_RESUME:
            check_current_cd_player_status();
            /* 기존 MISRA_15_02 위배 무시는 CD_PAUSE가 고려되지 않았음. */
            /* pause일 때도 play()가 호출됨, 의도하지 않은 동작일 수 있음. */
        case CD_PLAY:
            play();
            break;
        default:
            /* do nothing */
            break;
    }
}
```

그러나 수정된 코드에서는 현재 상태가 CD_PAUSE일 때에도 fall-through가 발생합니다. 이 경우 기존에 무시한 위배를 다시 검토해야 합니다. 따라서 저희 제품에서는 사용자 소스에 변경이 있는 경우 기존의 위배 무시를 더 이상 유지하지 않습니다.

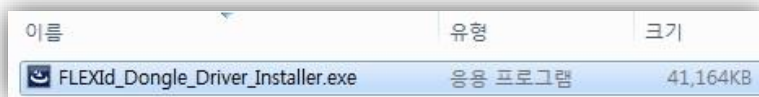
19. 부록 B - 동글 라이선스 기능

이 장에서는 Code Inspector에서 제공하는 라이선스 방식 중 하나인 동글 라이선스 기능에 관한 설명을 제공합니다.

동글 드라이버 설치

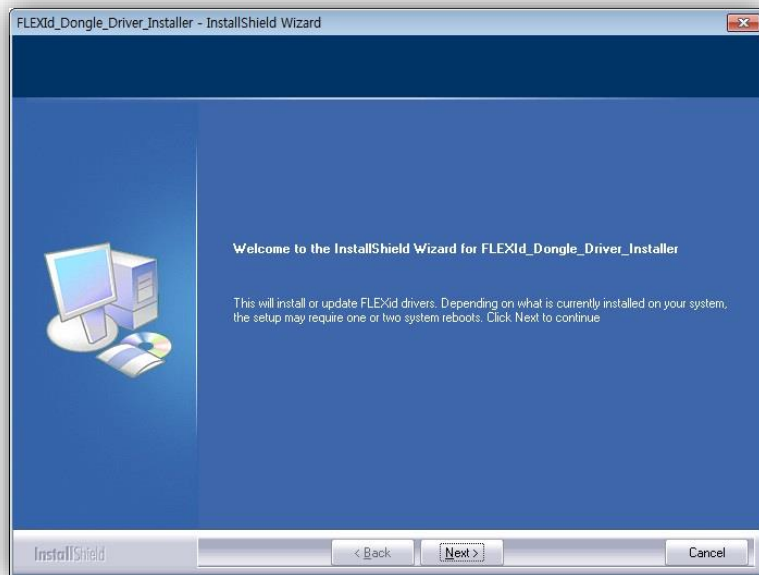
동글 라이선스 기능을 사용하기 위해서 제공받은 동글 드라이버 설치 파일로 동글 드라이버를 사전에 컴퓨터에 설치합니다. 설치 방법은 다음과 같습니다.

1. FLEXId_Dongle_Driver_Installer.exe 파일을 실행합니다.

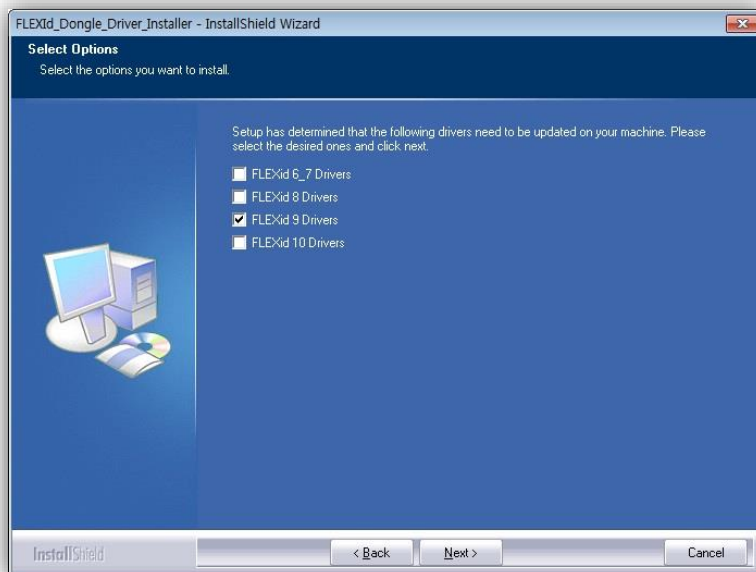


이름	유형	크기
FLEXId_Dongle_Driver_Installer.exe	응용 프로그램	41,164KB

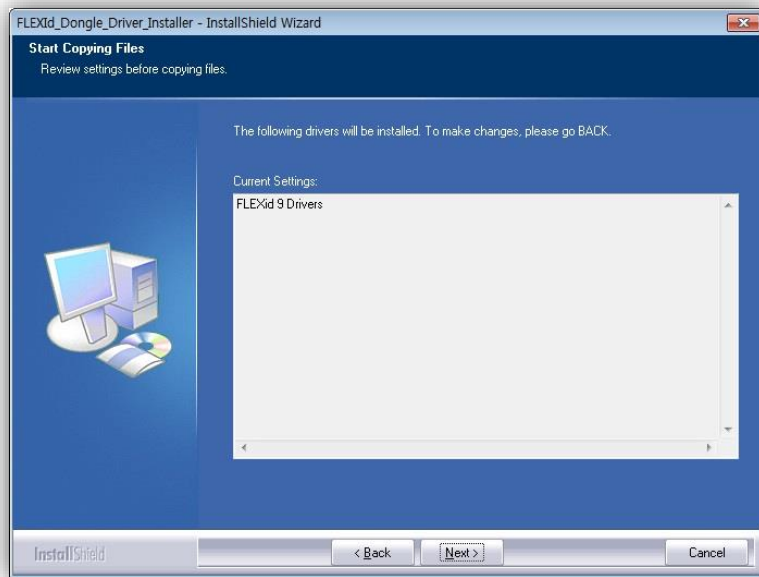
2. 설치 마법사가 실행되면 [Next]를 클릭합니다.



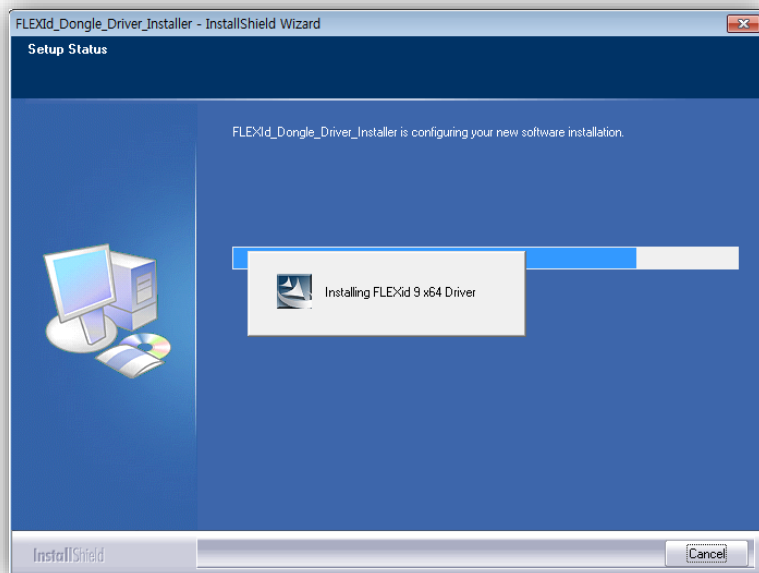
3. [FLEXid 9 Drivers] 선택 후 [Next]를 클릭합니다.



4. 설치하려는 드라이버가 맞는지 확인 후 [Next]를 클릭합니다.

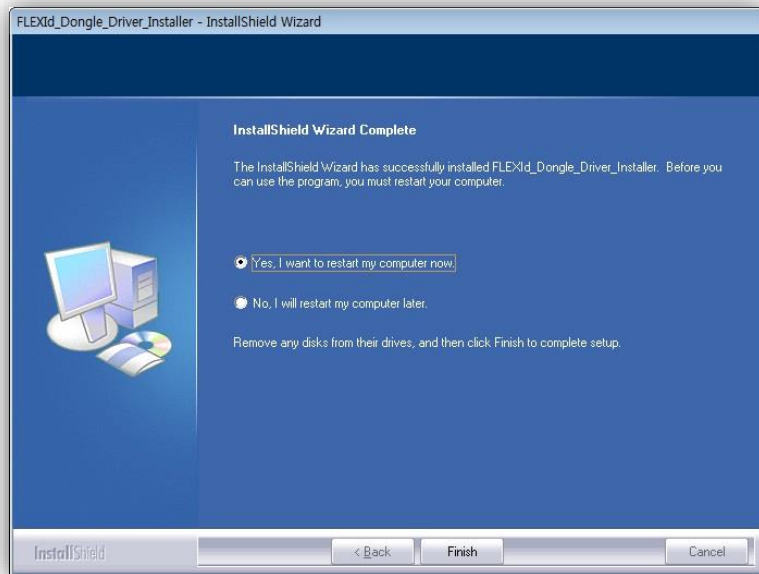


5. 동글 드라이버가 설치됩니다.

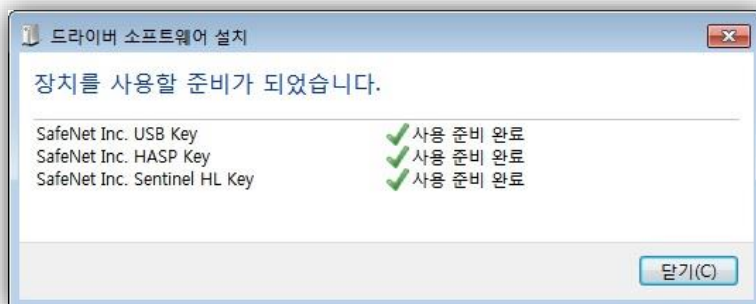


6. 설치가 완료되었습니다. 컴퓨터 재부팅 여부를 선택 후 [Finish]를 클릭

하면 드라이버 설치가 완료됩니다.



7. 드라이버 설치 완료 후, 동글을 컴퓨터에 연결하면 다음과 같이 동글 인식을 위한 설치가 진행됩니다.



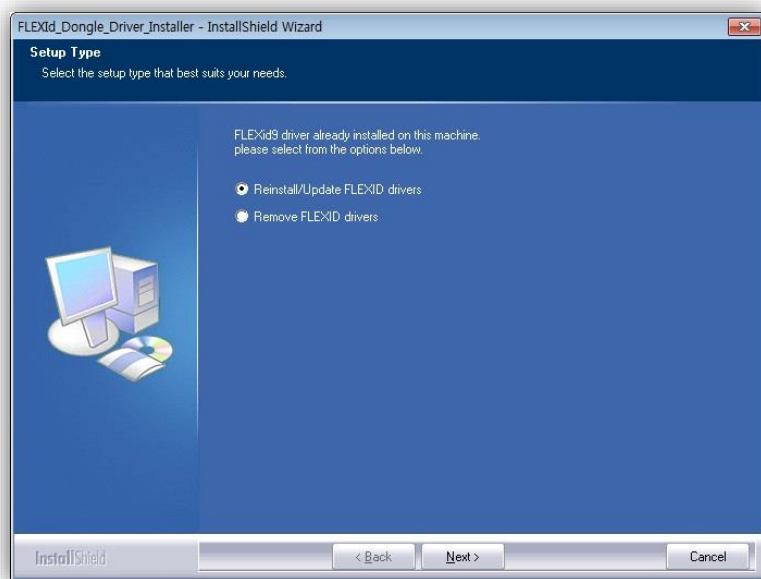
동글 드라이버 제거

동글 드라이버를 제거하기 위해서 제공받은 동글 드라이버 설치 파일을 실행합니다. 제거 방법은 다음과 같습니다.

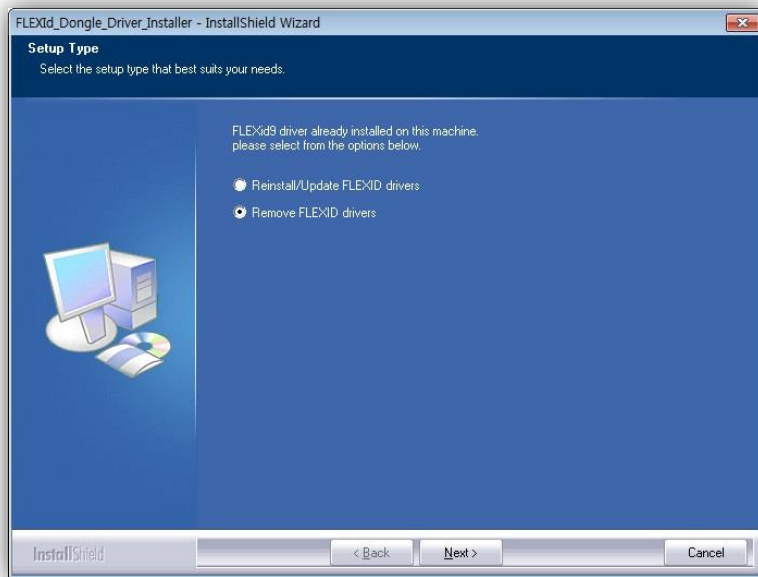
-
1. FLEXid_Dongle_Driver_Installer.exe 파일을 실행합니다.

이름	유형	크기
FLEXid_Dongle_Driver_Installer.exe	응용 프로그램	41,164KB

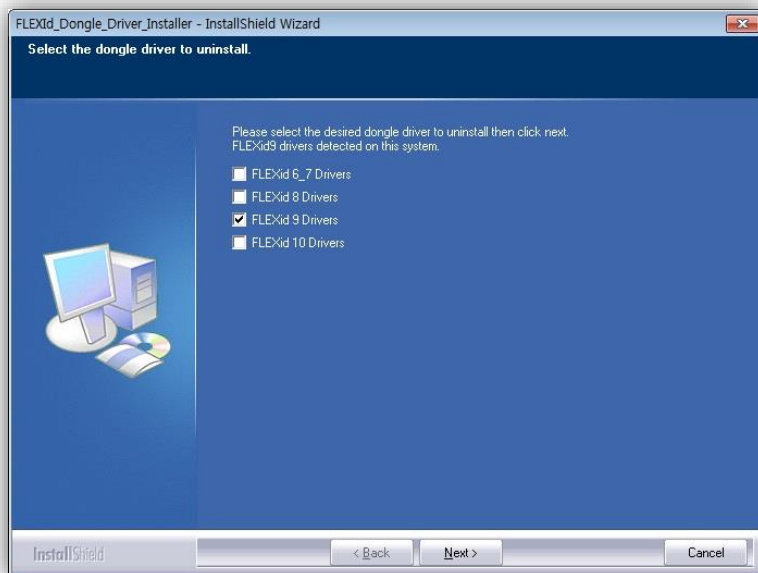
2. 동글 드라이버가 설치된 상태라면 제거 모드로 실행됩니다.



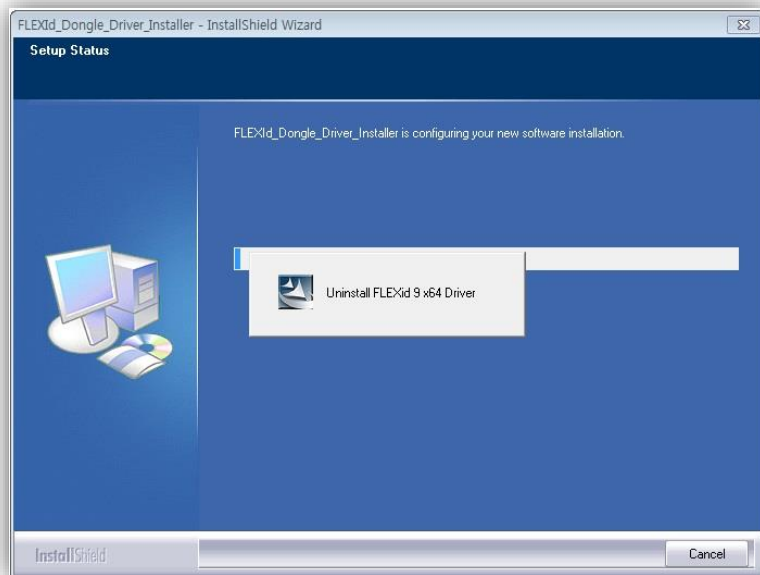
3. 드라이버 제거를 위해 [Remove FLEXID drivers] 체크 후 [Next]를 클릭합니다.



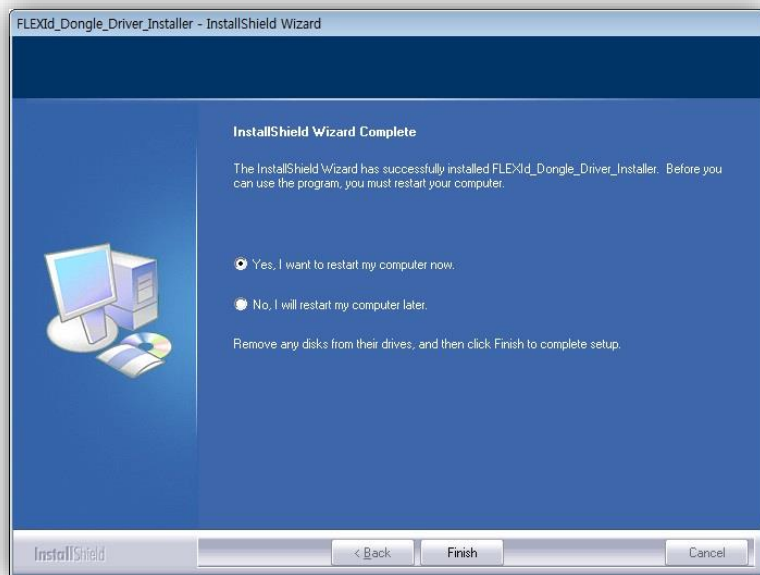
4. 제거하려는 [FLEXid 9 Drivers] 선택 후 [Next]를 클릭합니다.



5. 동글 드라이버가 제거됩니다.



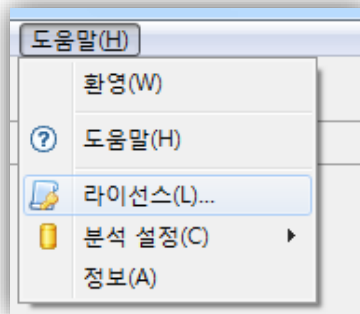
6. 제거가 완료됐습니다. 컴퓨터 재부팅 여부를 선택 후 [Finish]를 클릭하면 드라이버 제거가 완료됩니다.



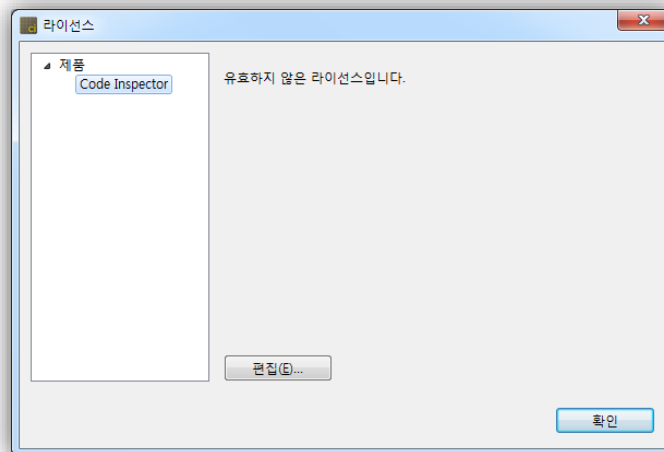
동글 라이선스 설정

동글 라이선스를 사용하려면 먼저 동글을 컴퓨터에 연결한 뒤, 라이선스를 등록해야 합니다. 라이선스 등록 과정은 다음과 같습니다.

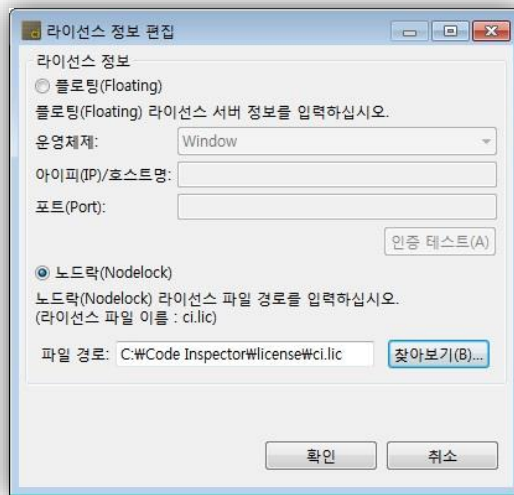
1. 메뉴의 [도움말] -> [라이선스]를 선택합니다.



2. [제품] -> [Code Inspector]를 선택하고 [편집]을 클릭합니다.



3. 라이선스 정보 편집 창에서 노트락 방식으로 라이선스를 입력하고 [확인] 버튼을 클릭하여 라이선스를 등록합니다.



4. 동글 라이선스가 등록이 완료되면 동글을 컴퓨터에 연결한 상태에서 Code Inspector를 사용할 수 있습니다.

※ 동글 라이선스 방식으로 Code Inspector 사용 중, 동글 연결 해제 등으로 동글 인식이 불가능 할 경우 제품 사용이 불가능해지고, 진행 중이던 작업이 강제 종료되므로 주의하시기 바랍니다.