

Table of Contents

- Code Inspection
 - 개요
 - 설명
 - 전자정부 표준프레임워크 표준 Inspection 룰셋
 - 설치
 - 전자정부 표준프레임워크 개발도구 다운로드 후 포함된 PMD 확인
 - Eclipse IDE의 Software Updates 기능을 사용
 - 환경설정
 - 전자정부 표준프레임워크 표준 Inspection 룰셋 적용하기
 - 룰 환경설정(Rules Configuration) 항목 설명
 - 업데이트
 - 업데이트 설치 시의 제약 사항

개요


정의된 규칙(Rule)을 기반으로 개발자가 작성한 소스 코드를 검사하여, 오류 및 위험 요인을 식별하여 알려 주는 기능을 제공한다.

설명

전자정부 표준프레임워크 적용 시 Code Inspection 구현도구를 통해 개발자가 작성한 소스 코드를 검사할 수 있다. 소스 코드 검사(Inspection) 작업을 통해 오류 및 위험 요인을 식별하여 개발자에게 편의성 및 효율성을 향상시킬 수 있다.

Code Inspection에서 제공하는 주요기능은 다음과 같다.

주요기능	설명
Syntax Error Inspection	작성한 소스 코드의 Syntax 오류를 검사하는 기능을 제공한다.
Logical Error Inspection	작성한 소스 코드에서 실행 시 발생 가능한 오류를 찾아내는 기능을 제공한다.
Reference Inspection	작성한 소스 코드에서 실행 시 사용되지 않는 부분을 찾아내는 기능을 제공한다.
Reporting	인스펙션을 수행한 결과를 Excel, HTML등이 문서 형식으로 제공하는 기능을 제공한다.
Rule Customizing	Rule을 정의하는 API를 사용하여 새로운 Rule을 정의하여 추가할 수 있는 기능을 제공한다.

전자정부 표준프레임워크의 Code Inspection 도구는 여러 오픈소스 Inspection 도구들 중에서 PMD가 선정되었으며, 이는 전자정부 표준프레임워크 개발도구에 PMD Plugin으로 포함되어 배포되고 있다.

전자정부 표준프레임워크 표준 Inspection 룰셋

PMD를 이용한 Code Inspection 시 기준이 되는 요소는 룰이며, 전자정부 표준프레임워크에서는 PMD가 제공하는 수 많은 룰 중에서 기본이 될 만한 것들을 간추려 전자정부 표준프레임워크 표준 Inspection 룰셋이라는 이름으로 구성하였다. 전자정부 표준프레임워크의 표준 Inspection 룰셋은 다음의 표와 같은 39개의 룰들로 구성된다. 개별 룰에 대한 상세한 설명은 [전자정부 표준 inspection 룰셋](#)을 참고한다.

번호	룰이름	설명
01	EmptyCatchBlock	내용이 없는 Catch Block이 존재
02	EmptyIfStmt	빈 if 구문의 사용을 피하도록 함
03	EmptyWhileStmt	빈 while 구문이 사용되었음
04	EmptyTryBlock	내용이 없는 try 블록이 존재함
05	EmptyFinallyBlock	내용이 없는 finally 블록이 존재함
06	UnnecessaryConversionTemporary	기본 데이터 타입을 String으로 변환할 때 불필요한 임시 변환 작업을 피하도록 함
07	EmptyStatementNotInLoop	필요없는 문장 (;)이 있음
08	WhileLoopsMustUseBraces	중괄호없이 사용된 while문의 사용은 바람직하지 못한 코딩 습관임
09	AssignmentInOperand	피연산자내에 할당문이 사용됨. 해당 코드를 복잡하고 가독성이 떨어지게 만듦
10	UnnecessaryParentheses	괄호가 없어도 되는 상황에서 불필요한 괄호를 사용할 경우 마치 메소드 호출처럼 보여서 소스 코드의 가독성을 떨어뜨릴 수 있음
11	SimplifyBooleanExpressions	boolean 사용 시 불필요한 비교 연산을 피하도록 함
12	SwitchStmtsShouldHaveDefault	Switch구문에는 반드시 default label이 있어야 함
13	AvoidReassigningParameters	넘겨받는 메소드 parameter 값을 직접 변경하는 코드 탐지.
14	FinalFieldCouldBeStatic	final field를 static으로 변경하면 overhead를 줄일 수 있음
15	EqualsNull	null 값과 비교하기 위해 equals 메소드를 사용하였음
16	SimpleDateFormatNeedsLocale	SimpleDateFormat 인스턴스를 생성할때 Locale 을 지정하는 것이 바람직함
17	ImmutableField	생성자를 통해 할당된 변수를 Final로 선언하지 않았음
18	AssignmentToNonFinalStatic	static 필드의 안전하지않은 사용 가능성이 존재
19	AvoidSynchronizedAtMethodLevel	mothod 레벨의 synchronization 보다 block 레벨 synchronization 을 사용하는 것이 바람직함
20	AbstractClassWithoutAbstractMethod	Abstract Class내에 Abstract Method가 존재하지 않음
21	UncommentedEmptyMethod	빈 메소드에 빈메소드임을 나타내는 주석을 추가할 것
22	AvoidConstantsInterface	Interface는 클래스의 behavior 을 구현하는 데에만 사용해야 함
23	DuplicateImports	import문이 중복 선언 되었음
24	ImportFromSamePackage	동일 패키지에 있을 때는 import문을 사용할 필요가 없음
25	SystemPrintln	System.out.print 가 사용됨. 전용 로거를 사용할 것을 권장
26	VariableNamingConventions	final이 아닌 변수는 밑줄을 포함할 수 없음
27	MisleadingVariableName	non-field 이름이 m_ 으로 시작함'

28	AvoidArrayLoops	배열의 값을 루프문을 이용하여 복사하는 것 보다, System.arraycopy() 메소드를 이용하여 복사하는 것이 효율적이며 수행 속도가 빠름
29	UnnecessaryWrapperObjectCreation	불필요한 Wrapper Object가 생성되었음. 탐지된 코드는 삭제하고, 별도의 parse관련 전용 메소드 사용을 권장
30	AvoidThrowingRawExceptionTypes	가공되지 않은 Exception을 throw하는 것은 비추천
31	AvoidThrowingNullPointerException	NullPointerException을 throw하는 것은 비추천
32	StringInstantiation	불필요한 String Instance를 생성하는 코드를 탐지. 간단한 형태의 코드로 변경 필요
33	StringToString	String 객체에서 toString()함수를 사용하는 것은 불필요함. 해당 코드 제거 필요
34	InefficientStringBuffering	StringBuffer 함수내에서 비문자열 연산 이용하여 직접 결합하는 코드 사용을 탐지. append 메소드 사용을 권장
35	InefficientEmptyStringCheck	빈 문자열 확인을 위해 String.trim().length() 을 사용하는 것은 피하도록 함. whitespace/Non-whitespace 확인을 위한 별도의 로직 구현을 권장
36	UselessStringValueOf	String 을 append 할 경우, String.valueOf 함수를 사용할 필요 없음
37	UnusedPrivateField	사용되지 않는 Private field의 탐지
38	UnusedPrivateMethod	사용되지 않는 Private Method 선언을 탐지
39	UnusedFormalParameter	메소드 선언 내에서 사용되지 않는 파라미터를 탐지

설치

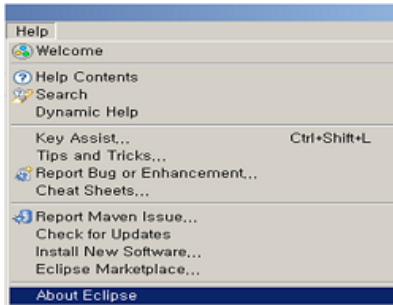
전자정부 표준프레임워크 Inspection 도구인 PMD는 다음의 두 가지 방법으로 적용/설치할 수 있다.

- 전자정부 표준프레임워크 개발도구의 구현도구(Eclipse IDE)를 다운로드 후 포함된 PMD 확인
- Eclipse IDE의 Software Updates 기능을 사용

전자정부 표준프레임워크 개발도구 다운로드 후 포함된 PMD 확인

전자정부 표준프레임워크 개발도구는 [전자정부 표준프레임워크 온라인 지원 포탈사이트](#)의 다운로드 > 개발환경 > **구현도구**에서 다운로드 받을 수 있다. 다운로드 받은 구현도구 파일의 압축을 풀고, 실행시키면 다음의 작업으로 PMD의 설치내역과 버전을 확인할 수 있다.

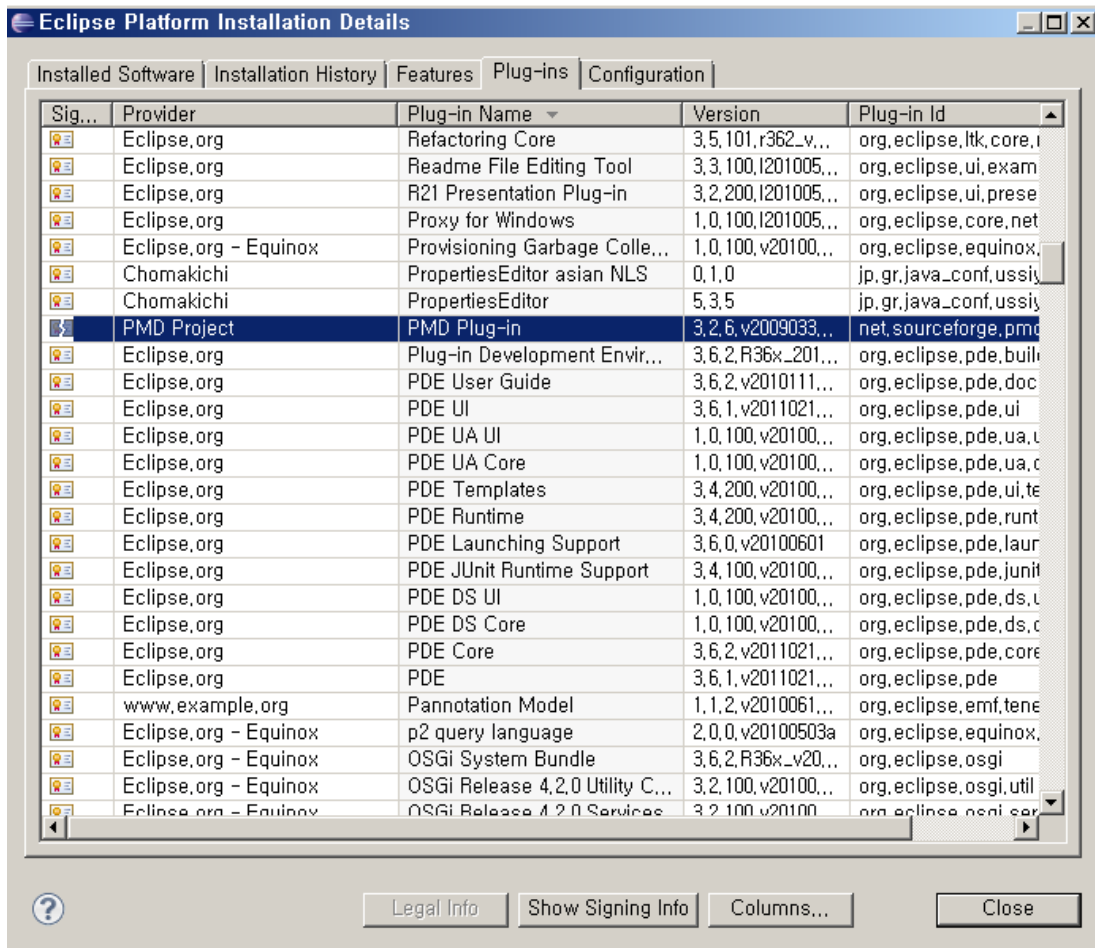
1. Eclipse IDE의 메뉴에서, Help > **About Eclipse** 선택



2. About Eclipse 창에서, 하단 중앙의 **Installation Details** 버튼을 클릭



3. About Eclipse Plug-ins 창의 플러그인 목록에서 PMD Plug-in 항목과 버전을 확인



Eclipse IDE의 Software Updates 기능을 사용

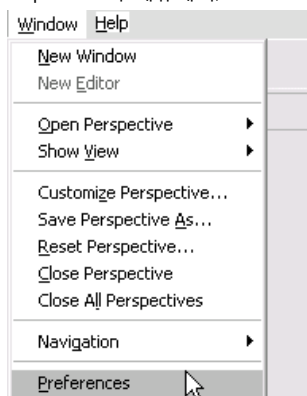
Eclipse IDE의 Software Updates 기능을 사용한 PMD 설치 방법은 업데이트 항목에 기술된 방법과 동일하므로 해당 설명을 참조하여 최초 PMD 설치 작업을 진행한다.

환경설정

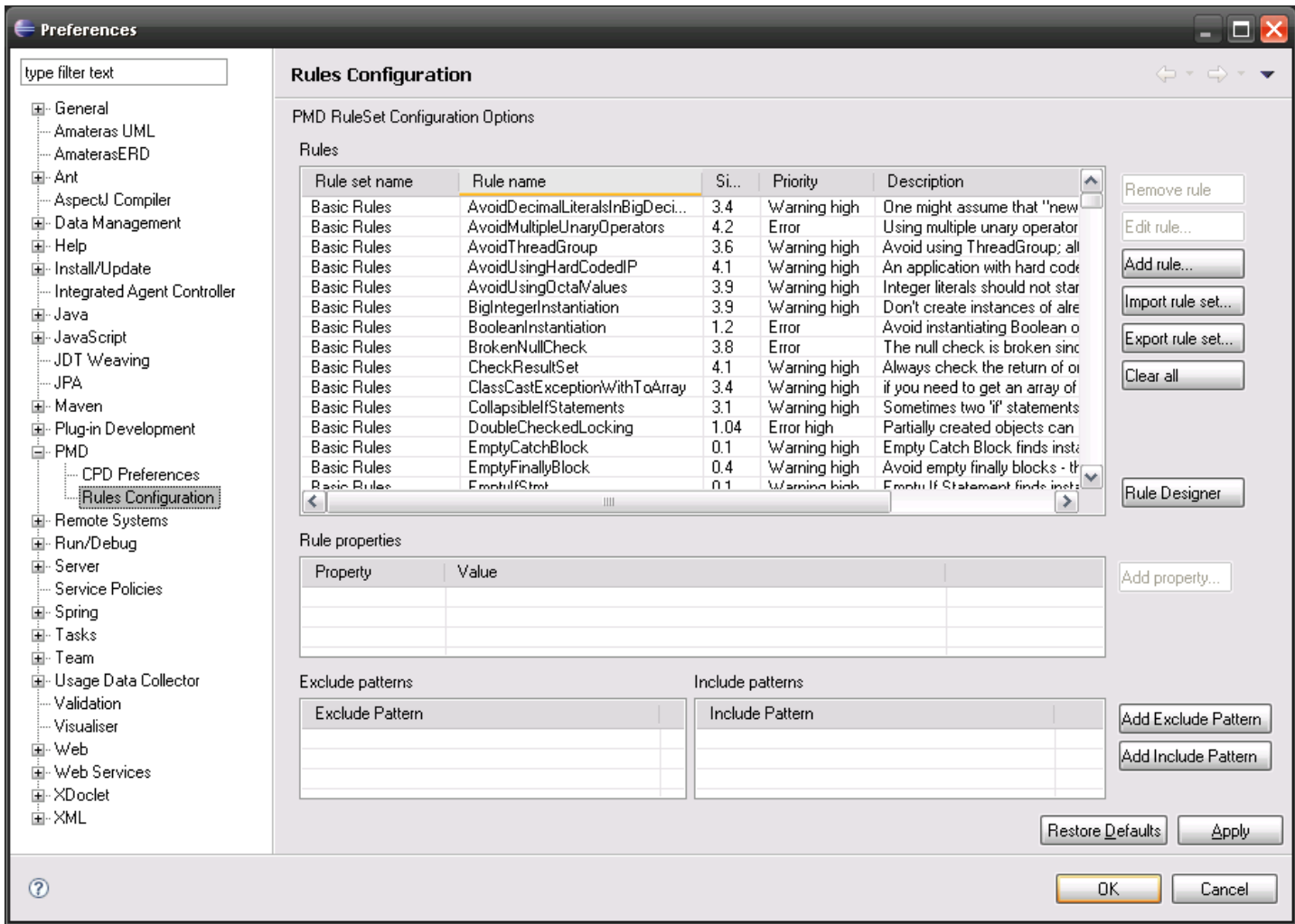
Eclipse IDE의 환경설정을 통해, 전자정부 표준프레임워크 표준 Inspection 룰셋을 PMD Plugin에 적용할 수 있으며, 다음과 같은 순서로 수행한다.

전자정부 표준프레임워크 표준 Inspection 룰셋 적용하기

- 다음 두가지 경로 중, 하나를 선택해서 전자정부 표준프레임워크 표준 Inspection 룰셋을 내려받기
 - 표준 Inspection 룰셋 한글/영문판의 압축파일
 - 전자정부 표준프레임워크 온라인 지원 포털사이트의 다운로드 > 개발환경 > 구현도구 소스코드에서 내려받기
- 내려받은 파일을 임의의 디렉터리에 압축 해제
- Eclipse IDE의 메뉴에서, Window > **Preferences** 선택

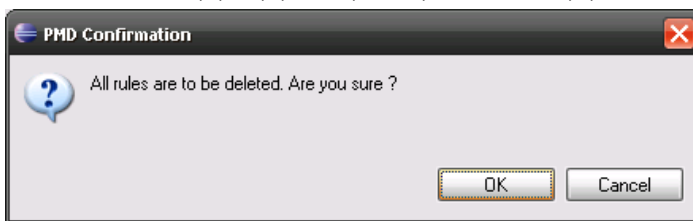


4. Preferences 창의 왼쪽 메뉴 구조에서, PMD > **Rules Configuration** 선택



5. Preferences 창에서, 오른쪽 중앙의 **Clear all** 버튼 클릭

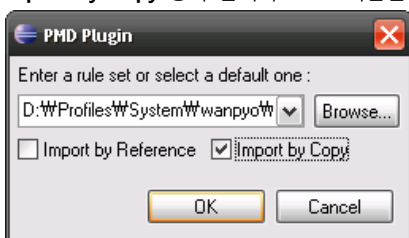
6. PMD Confirmation 대화창에서, **OK** 버튼 클릭으로 모든 룰 삭제



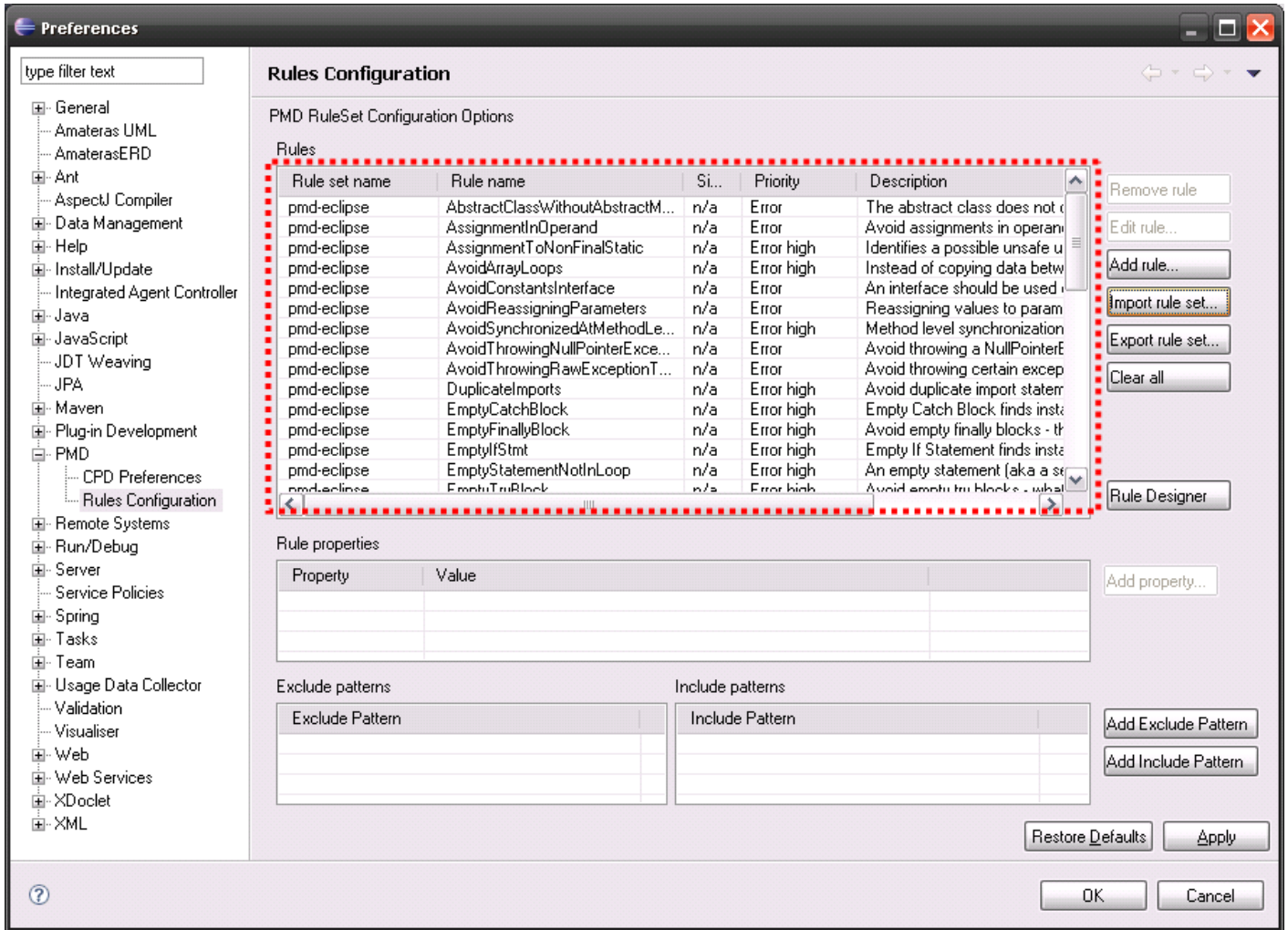
7. Preferences 창의 오른쪽의 **Import rule set...** 버튼 클릭

8. PMD Plugin 대화창에서, **Browse...** 버튼을 클릭하여 앞서 압축해제한 룰셋 파일 중 한글판인 **EgovInspectionRules_kor.xml** 파일을 선택

9. **Import by Copy** 항목 선택하고 **OK** 버튼을 클릭



10. Preferences 창의 Rules 그리드 목록에서 방금 들어온 룰 확인



11. Preferences 창의 OK 버튼을 클릭하여, 새로이 들어온 룰을 Eclipse IDE에 반영

룰 환경설정(Rules Configuration) 항목 설명

PMD 플러그인의 룰 환경설정을 구성하고 있는 항목은 다음과 같다.

구성 항목	유형	설명
Rules	그리드	전자정부 표준 Inspection 룰셋을 구성하고 있는 전체 룰의 목록을 표시하는 그리드
Remove rule	버튼	Rules 그리드에서 선택한 항목의 룰을 삭제
Edit rule	버튼	Rules 그리드에서 선택한 항목의 룰을 수정
Add rule	버튼	Rules 그리드의 룰셋에 새로운 룰을 추가
Import rule set...	버튼	하나 이상의 룰들을 파일단위로 룰셋에 추가
Export rule set...	버튼	하나 이상의 룰들을 추출하여 외부 파일로 저장
Clear all	버튼	Rules 그리드에 나타난 모든 룰들을 삭제
Rule Designer	버튼	XPath 기반의 새로운 룰을 만들기위한 외부 프로그램 실행(PMD 플러그인 제공)
Rule properties	그리드	개별 룰에 대한 Key-Value 형태의 속성 목록을 표시, 주로 룰에 대한 XPath 쿼리(XQuery)를 표시
Add property...	버튼	개별 룰에 새로운 property 추가
Exclude patterns	그리드	전체 룰셋을 프로젝트에 적용할 때, 예외 대상 패턴을 정의한 목록을 표시
Include patterns	그리드	전체 룰셋을 프로젝트에 적용할 때, 포함 대상 패턴을 정의한 목록을 표시
Add Exclude Patterns	버튼	룰셋 적용 예외 대상 패턴을 새로이 추가
Add Include Patterns	버튼	룰셋 적용 포함 대상 패턴을 새로이 추가

위의 항목 중, Rules 그리드와 관련된 기능에 대한 상세설명은 [사용자정의 룰 등록,반영](#) 항목을 참조하면 되고, 예외/포함 패턴과 관련된 설명은 다음과 같다.

Exclude patterns

Rule Configuration 오른쪽 하단의 **Add Exclude Patterns** 버튼을 클릭하면, 룰셋 적용의 예외항목에 대한 패턴을 다음 그림과 추가할 수 있다.

Exclude patterns	Include patterns																
<table><thead><tr><th>Exclude Pattern</th><th></th></tr></thead><tbody><tr><td>*/main.java</td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></tbody></table>	Exclude Pattern		*/main.java						<table><thead><tr><th>Include Pattern</th><th></th></tr></thead><tbody><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></tbody></table>	Include Pattern							
Exclude Pattern																	
*/main.java																	
Include Pattern																	

Add Exclude Pattern

Add Include Pattern

기본적으로 '*/PATTERN/*'와 같은 형태의 패턴이 기본으로 입력되며, 이를 더블 클릭해서 해당 항목을 수정할 수 있다. 위 그림에 기술된 패턴 예는 'main.java'라는 이름을 가진 파일에 대해서는 Inspection 룰을 적용하지 않음을 의미한다.

Include patterns

Rule Configuration 오른쪽 하단의 **Add Include Patterns** 버튼을 클릭하면, 룰셋 적용의 포함항목에 대한 패턴을 다음 그림과 추가할 수 있다.

Exclude patterns	Include patterns																
<table><thead><tr><th>Exclude Pattern</th><th></th></tr></thead><tbody><tr><td>*/main.java</td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></tbody></table>	Exclude Pattern		*/main.java						<table><thead><tr><th>Include Pattern</th><th></th></tr></thead><tbody><tr><td>*/egov/*</td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></tbody></table>	Include Pattern		*/egov/*					
Exclude Pattern																	
*/main.java																	
Include Pattern																	
/egov/																	

Add Exclude Pattern

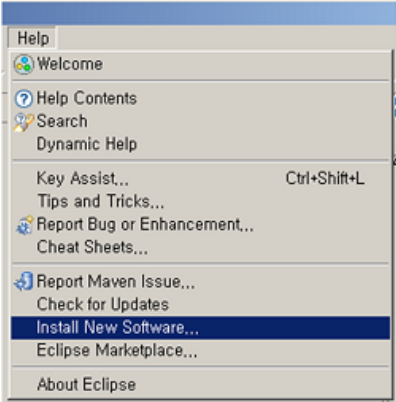
Add Include Pattern

기본적으로 '*/PATTERN/*'와 같은 형태의 패턴이 기본으로 입력되며, 이를 더블 클릭해서 해당 항목을 수정할 수 있다. 위 그림에 기술된 패턴 예는 'egov' 라는 경로를 포함한 모든 파일에 대해서 Inspection 룰을 적용함을 의미한다.

업데이트

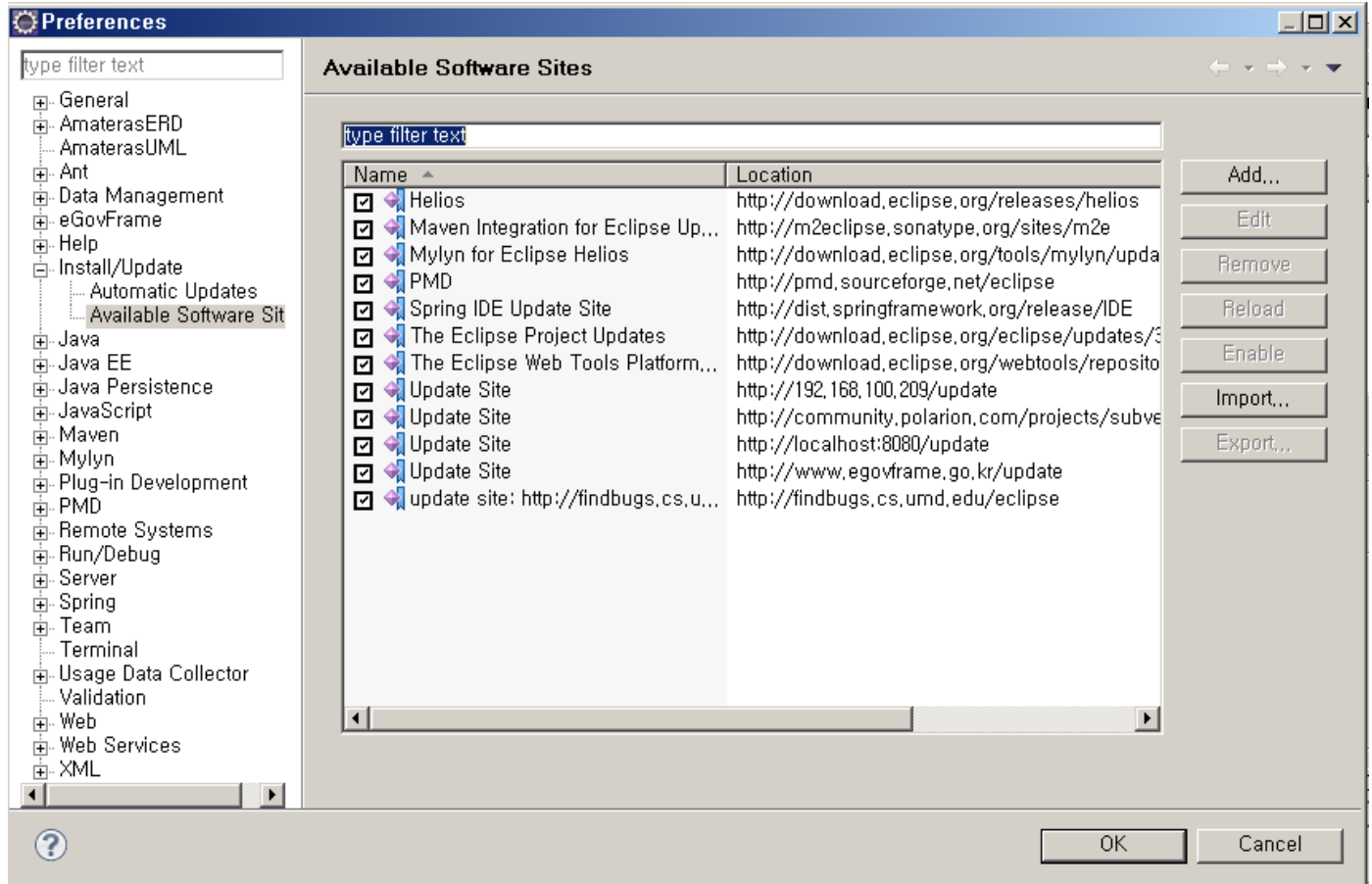
사용자는 Eclipse IDE에서 PMD를 최신 버전으로 업데이트할 수 있다. 업데이트하는 방법은 다음과 같다.

- 1. Eclipse IDE의 메뉴에서, Help > **Install New Software** 를 클릭한다.

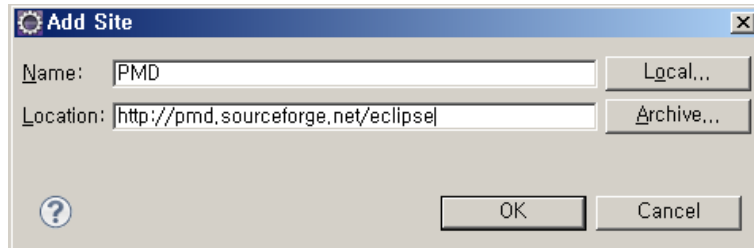


- 2. 'Available Software' 창에서 **Available Software Sites** 링크를 클릭한다.

3. 'Available Software Sites' 목록 우측에 있는 **Add** 버튼을 클릭한다.

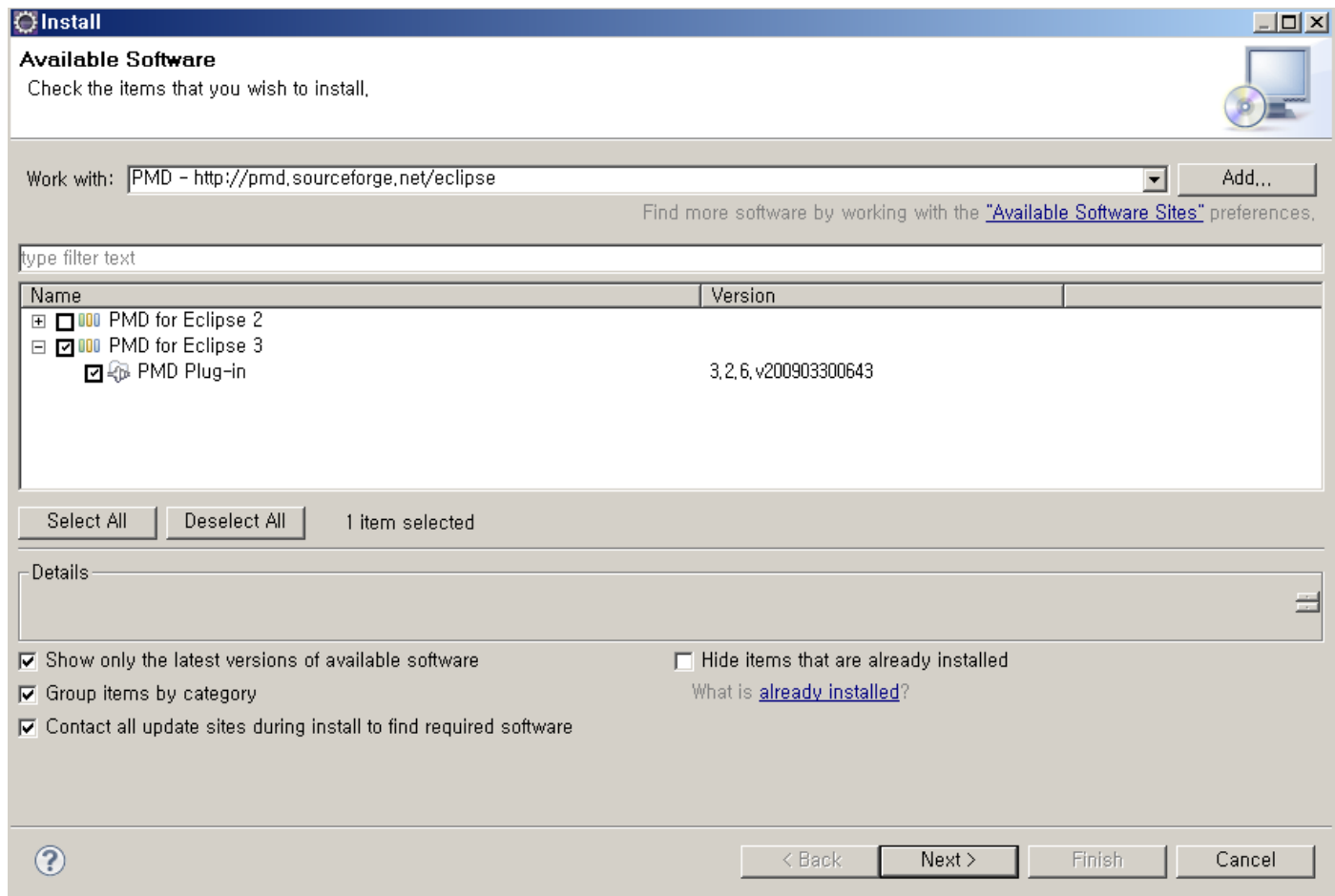


4. 'Add Site' 창에서, **Location** 입력 항목에 **http://pmd.sourceforge.net/eclipse** 를 입력하고 **OK** 버튼을 클릭한다.



5. 'Available Software' 목록에 입력한 URL 이 추가된다.

6. 입력한 URL의 트리에서 **PMD for Eclipse 3**을 선택한다.



7. 하단에 있는 **Next** 버튼을 클릭하여 업데이트를 시작한다.

업데이트 설치 시의 제약 사항

전자정부 표준프레임워크의 개발도구(전자정부 표준프레임워크 온라인 지원 포털사이트에서 다운로드 받은)를 사용하지 않거나, PMD 플러그인을 Software Updates를 통해서 직접 업데이트하였을 경우 다음과 같은 제약 사항이 있으므로 조치 후 사용한다.

제약 사항

- PMD의 VariableNamingConventions 룰에 대한 소스 코드내 메시지 하드코딩으로 인하여, 한글판 표준 Inspection 룰셋을 설치하여도 영문 메시지가 출력됨

해결 방법

1. 다음 경로를 통해 수정된 파일을 내려받기
 - 한글표기 해결 라이브러리 파일: 수정된 라이브러리 파일
2. 내려받은 파일을 임의의 디렉터리에 압축 해제
3. **pmd14-4.2.5.jar** 파일을 다음 위치의 원본 라이브러리와 교체함
 - 교체할 파일의 경로: %Eclipse 설치경로%\plugins\wnet.sourceforge.pmd.eclipse.plugin_xxx\lib\pmdxx-x.x.x.jar (x는 버전정보)
4. Eclipse를 재기동하여 변경된 라이브러리 파일 반영