

Code Inspection

Summary

This guide discusses the code inspection feature based on defined rules, notifies identified error and dangerous factors

Description

Using Code Inspection Implementation tool, developer's source code can be inspected when applying eGovFrame. Source code inspection can increase developers' convenience and efficiency by identifying errors and dangerous factors.

Code Inspection provides following features

Main Fuction	Description
Syntax Error Inspection	It provides the function to inspect syntax errors in written source codes.
Logical Error Inspection	It provides the function to inspect possible errors in written source codes.
Reference Inspection	It provides the function to inspect unused parts in written source codes.
Reporting	It provides the function to provide result from inspections as Excel and HTML document format.
Rule C ustomizing	It provides the function to add newly defined rules using the API that defines the rule.

PMD[<http://pmd.sourceforge.net/>] was chosen as the Code Inspection tool for eGovFrame out of other various open source inspection tools. This is distributed as PMD Plugin for eGovFrame IDE.

eGovFrame Standard Inspection Rule Set

Rule is basis for Code Inspection using PMD, eGovFrame summarizes eGovFrame Standard Inspection Rule Set out of rules provided by PMD.

eGovFrame Standard Inspection Rule consists of following 39 rules. Refer to eGovFrame Standard Inspection Rule Set for detailed information.

No.	Rule name	Description
01	EmptyC atchBlock	Empty Catch Block found
02	EmptyIfStmnt	Empty if Statement found
03	EmptyWhileStmnt	Empty while Statement found
04	EmptyTryBlock	Empty try Block found
05	EmptyFinallyBlock	Empty finally Block found
06	UnnecessaryC onversionTemporary	Unnecessary Temporary Conversion found when default data type is changed to String
07	EmptyStatementNotInLoop	Empty Statement (;) found
08	WhileLoopsMustUseBraces	Use of while loops without braces is discouraged
09	AssignmentInOperand	Assignment in Operand found. It increases code complexity and decreases readability.
10	UnnecessaryParentheses	Unnecessary Parentheses found. It decreases readability.
11	SimplifyBooleanExpressions	Avoid unnecessary boolean compare statement
12	SwitchStmntsShouldHaveDefault	Default label is required in Switch statements
13	AvoidReassigningParameters	Reassignment on method parameters found.
14	FinalFieldC ouldBeStatic	Set final field to static to reduce overhead
15	EqualsNull	Equals method used with null
16	SimpleDateFormatNeedsLocale	Use Locale creating SimpleDateFormat instances
17	ImmutableField	Set Final to allocated variable created by constructor
18	AssignmentToNonFinalStatic	Static field can be used in unsafe manner
19	AvoidSynchronizedAtMethodLevel	Use of block level synchronization is encouraged than that of mothod level synchronization
20	<u>AbstractClassWithoutAbstractMethod</u>	No Abstract Method found in Abstract Class
21	UncommentedEmptyMethod	Add Comment to Empty methods
22	AvoidC onstantsInterface	Use Interface to implement Class behavior only
23	DuplicateImports	Duplicate Import found
24	ImportFromSamePackage	Unnecessary import statement found within the same package
25	SystemPrintln	System.out.print is used. Exclusive logger is recommended.
26	VariableNamingC onventions	Only final variables are allowed to have underscore
27	MisleadingVariableName	non-field name begins with m_ ' found
28	AvoidArrayLoops	<u>Use System.arraycopy() method instead of looping to copy array values, it's more efficient and fast.</u>
29	<u>UnnecessaryWrapperObjectC reation</u>	Unnecessary Wrapper Object found. Delete specified code, and use of special parse method is recommended
30	AvoidThrowingRawExceptionTypes	Throwing raw Exception is discouraged
31	AvoidThrowingNullPointerException	NullPointerException throwing is discouraged
32	StringInstantiation	Unnecessary String Instance Creation code found. Make the code simpler
33	StringToString	toString() method used to String object. Remove specified code.
34	InefficientStringBuffering	Direct join by non-string operation within StringBuffer function found. Use append method instead.
35	InefficientEmptyStringC heck	Avoid using String.trim().length() to check empty string. Implement special logic tocheck whitespace/Non-whitespace.
36	UselessStringValueOf	String.valueOf function is unnecessary when appending String.
37	UnusedPrivateField	Unused Private Field found

38	UnusedPrivateMethod	Notify Unused Private Method declaration
39	UnusedFormalParameter	Notify Unused parameter in method declaration

Installation

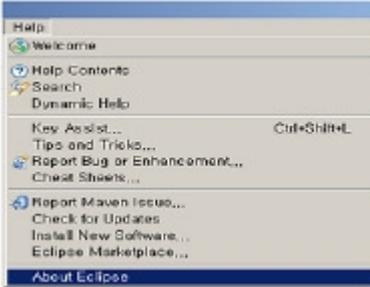
You can install PMD, an eGovFrame inspection tool, via either following two methods.

- Download the Eclipse IDE (eGovFrame Implementation Tool), and check the included PMD
- Using Software Updates feature in the Eclipse IDE

Check PMD included in the eGovFrame implementation tool

You can download the eGovFrame implementation tool from the eGovFrame online support portal [<http://www.egovframe.go.kr>]. Browse to Download > Development Environment > Implementation Tool. Unarchive the downloaded file, run and follow steps below to check a PMD version.

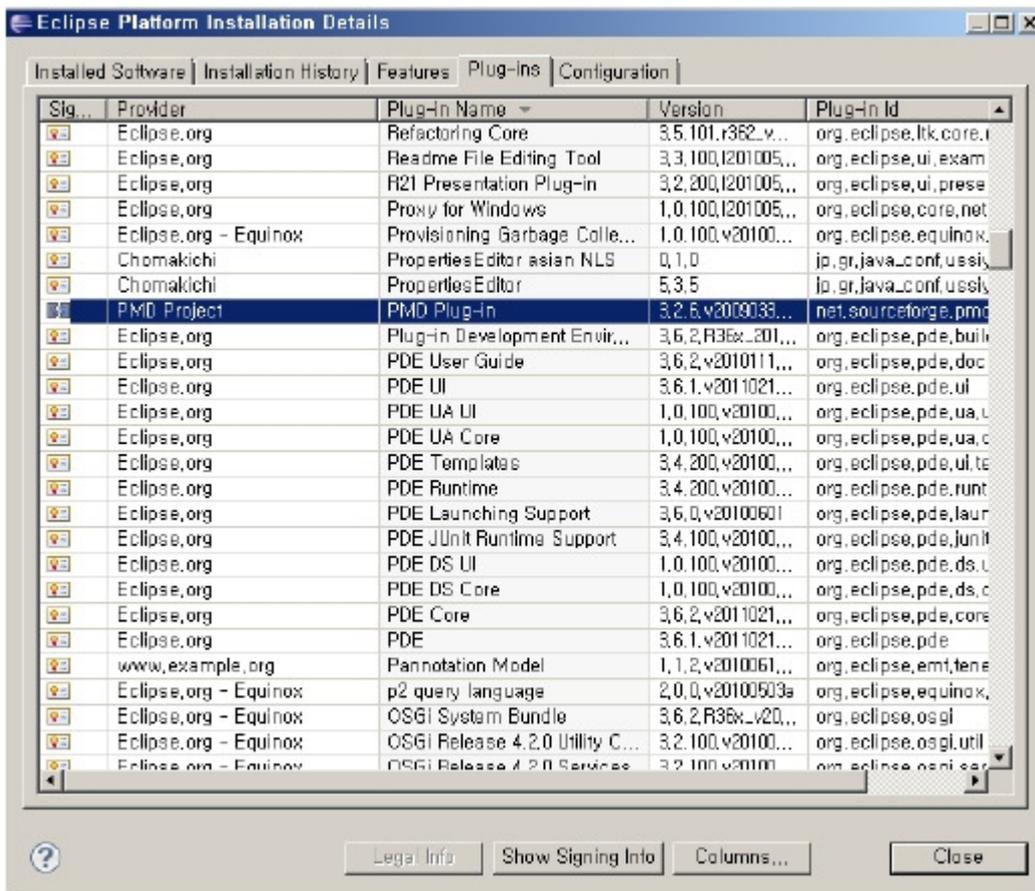
1. Select Help > About Eclipse in Eclipse IDE Menu.



2. Click the Installation Details button in the About Eclipse window.



3. Review the PMD Plug-in item and version in the Plug-in list of About Eclipse Plug-ins window.



Using Software Updates feature in Eclipse IDE

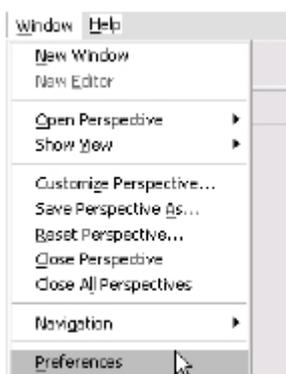
PMD installation using Software Updates feature in Eclipse IDE is the same as Update Section. Please refer to the description and proceed to initial PMD installation.

Preferences

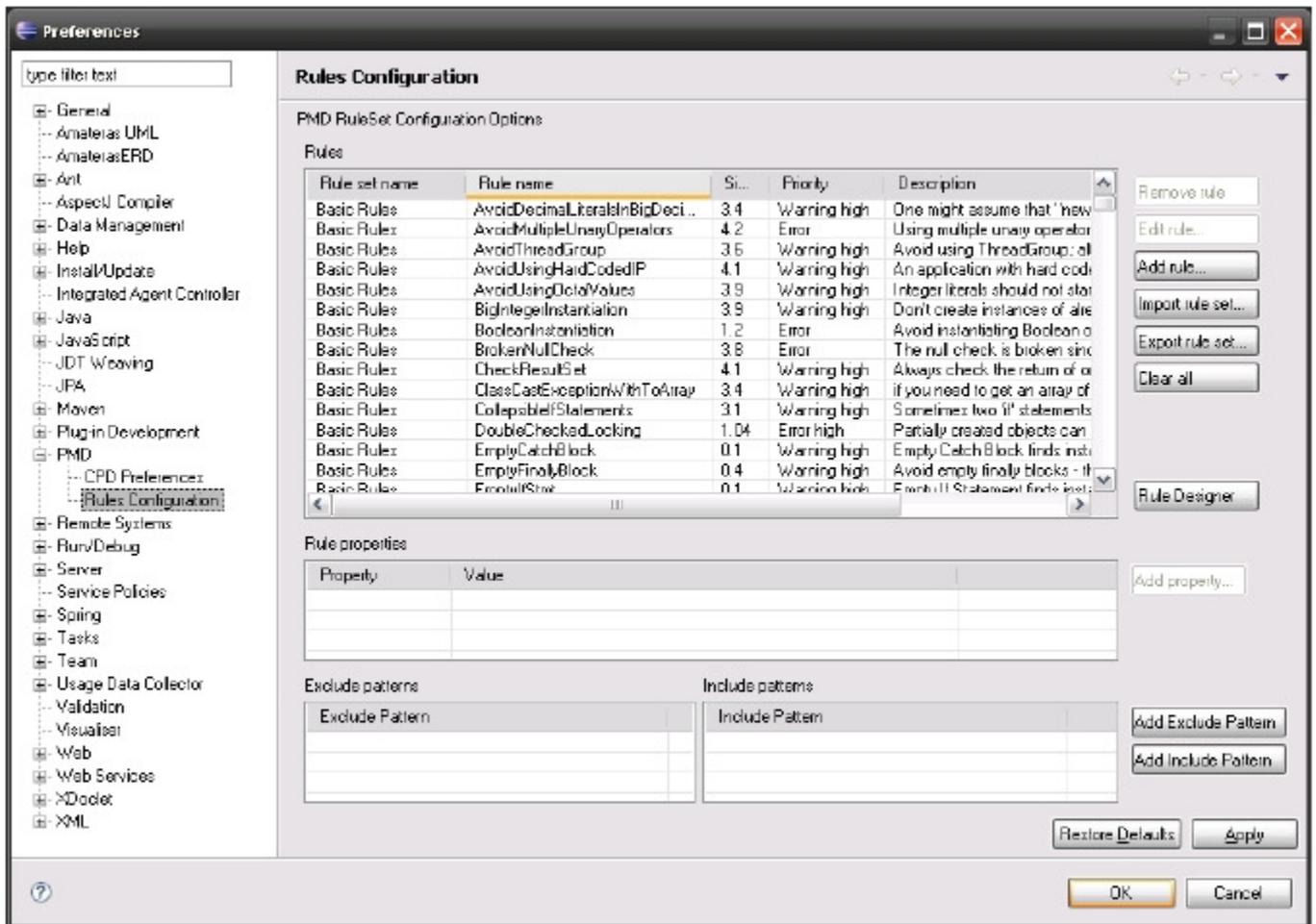
You can apply the eGovFrame standard Inspection RuleSet to PMD Plug-in via the preference in the Eclipse IDE, follow steps below.

Apply the eGovFrame Standard Inspection RuleSet

1. Get the eGovFrame Standard Inspection RuleSet via following 2 ways.
 - Archive file of Standard Inspection RuleSet Korean/English version
 - Browse to eGovFrame online support portal, and download the file from Download > IDE > Implementation Source Code.
2. Unarchive the downloaded file.
3. Select Window > Preferences in menu of the Eclipse IDE



4. Select PMD > Rules Configuration in menu pane at the left side of the Preferences window.



5. Click the "Clear all" button

6. Click OK in the PMD Confirmation dialog to delete all rules.



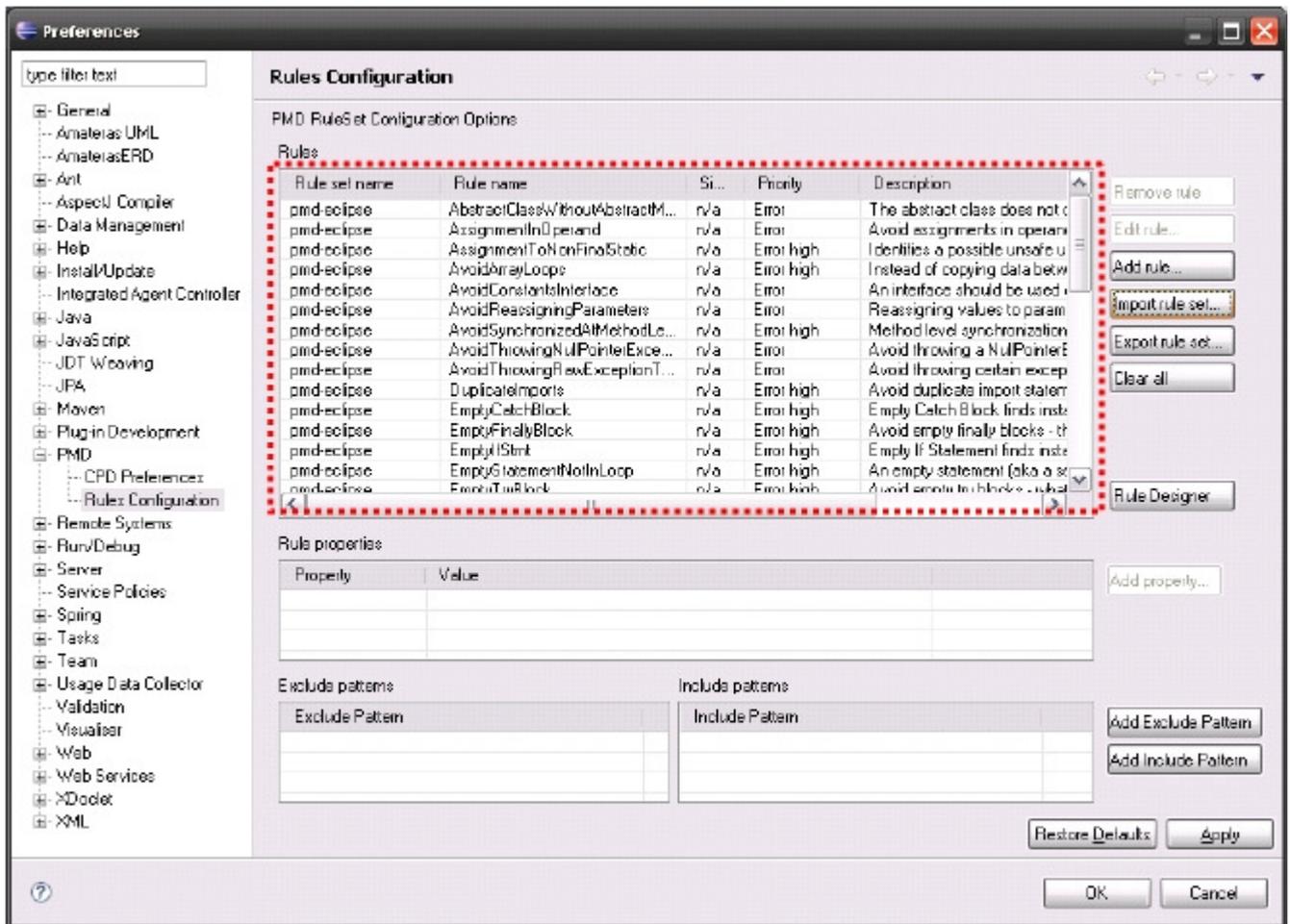
7. Click the "Import rule set..." button

8. In the PMD Plugin dialog, Click the Browse button and select the *EgovInspectionRules_kor.xml* file. (Korean version from the unarchived RuleSet files)

9. Check Import by Copy, then click OK.



10. Review a Rules grid list in the Preferences window for Imported rule.



11. Click OK in the Preferences window to apply newly imported rules.

Rules Configuration Description

Following table lists items that configure Rule Preferences of PMD Plug-in.

Configuration	Type	Description
Rules	grid	Displays a list for all rules composing eGov Standard Inspection RuleSet
Remove rule	button	Delete selected rule from Rules grid
Edit rule	button	Edit selected rule from Rules grid
Add rule	button	Add rule to Rules grid
Import rule set...	button	Import rules from RuleSet file.
Export rule set...	button	Export rules to file.
Clear all	button	Delete all rules in grid
Rule Designer	button	Launch external program for creating new rule in XPath (PMD Plug-in)
Rule properties	grid	Displays Key-Value typed property for specific rule, usually XPath query (XQuery) on Rules
Add property...	button	Add new property to specific rule
Exclude patterns	grid	Displays exclude patterns list when applying whole RuleSet to a project
Include patterns	grid	Displays include patterns list when applying whole RuleSet to a project
Add Exclude Patterns	button	Add exclude patterns
Add Include Patterns	button	Add include patterns

The description for Rules grid is provided in Register and Apply Custom Rules. Exclude/Include pattern is described below.

Exclude patterns

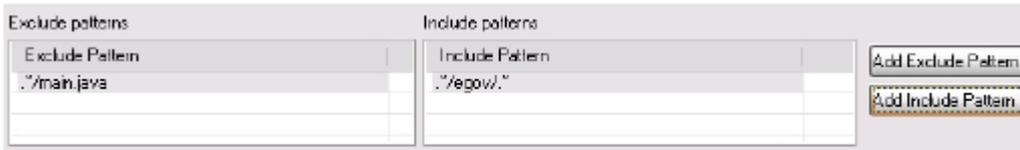
You can add exclude patterns by clicking the Add Include Pattern button as shown below.



A default pattern, '*/PATTERN/*.*', will be entered. You can change the pattern by double clicking it. The pattern in previous image means Inspection rule won't apply to all the files named main.java.

Include patterns

You can add include patterns by clicking Add Include Patterns button as shown below.

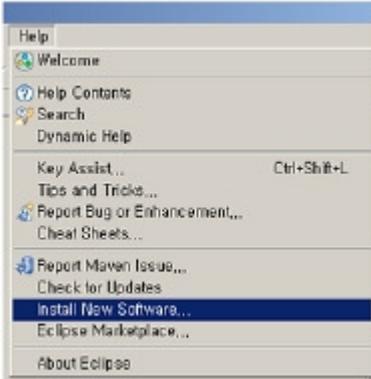


Default pattern, '.*/*PATTERN/*.*', will be entered. You can change the pattern by double clicking it. The pattern in previous image means the inspection rule will apply to all the files containing egov as a parent folder.

Update

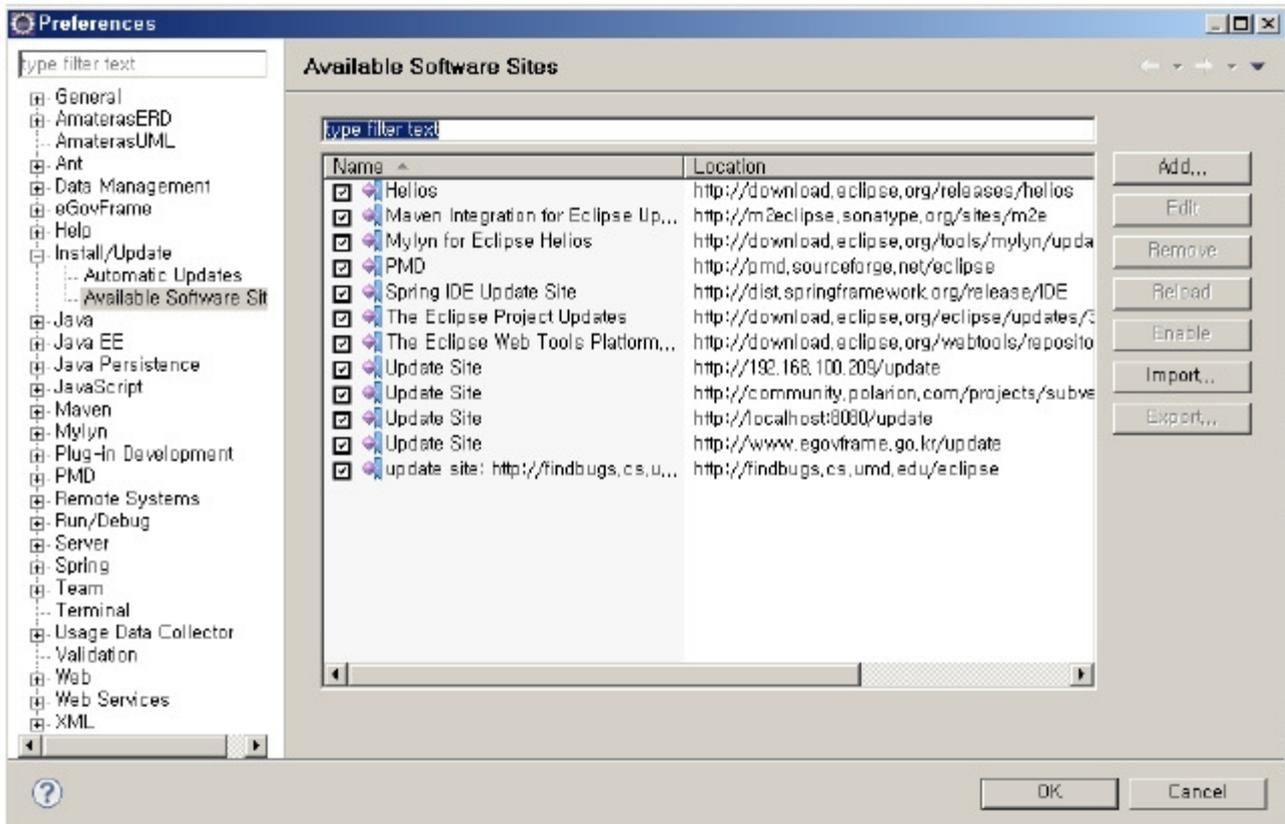
You can update the latest version of PMD in the Eclipse IDE. Following steps describe how to update.

1. Click Help > Install New Software in the Eclipse IDE Menu.

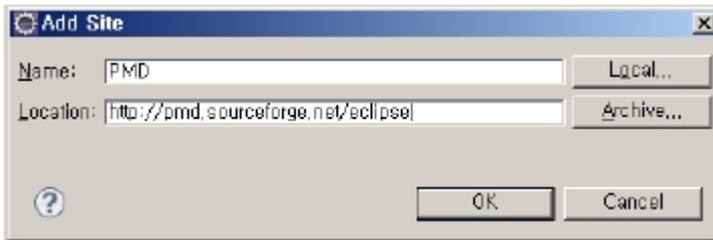


2. Click Available Software Sites link in 'Available Software' window.

3. Click the Add... button.

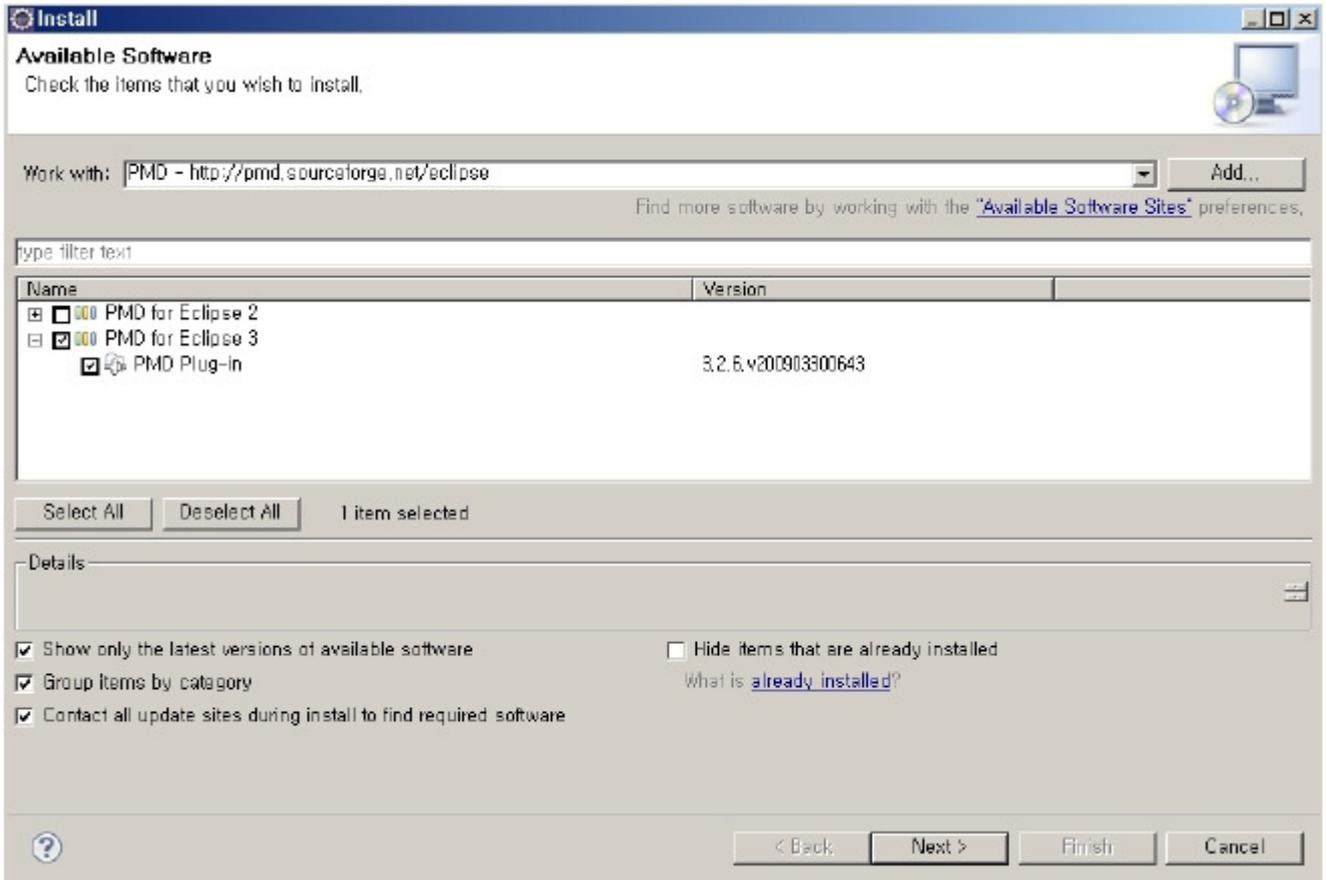


4. Enter <http://pmd.sourceforge.net/eclipse> to Location in 'Add Site' window, then click OK.



5. URL is added to 'Available Software' list.

6. Select PMD for Eclipse 3 from the tree.



7. Click the Next button to start up the updating procedure.

Limitations of the "Update" installation

If you don't use eGovFrame Development Tool (downloaded from eGov standard framework online support portal) or updated PMD plug-in using Software Updates directly, following limitations may occur.

Limitations

- Due to message hardcode in the source code on VariableNamingConventions rule of PMD, messages are displayed in English, even if the Korean version of standard Inspection RuleSet is installed.

Solution

1. Download a modified file from following path
 - Korean display fixed library file: Modified library file
2. Unarchive the downloaded file.
3. Replace the pmd14-4.2.5.jar file to the original library at the following
 - Original file path: %Eclipse directory%\plugins\net.sourceforge.pmd.eclipse.plugin_xxx\lib\pmdxx-x.x.x.jar (x denotes version)
4. Relaunch the Eclipse to apply a changed library file.