FUJITSU

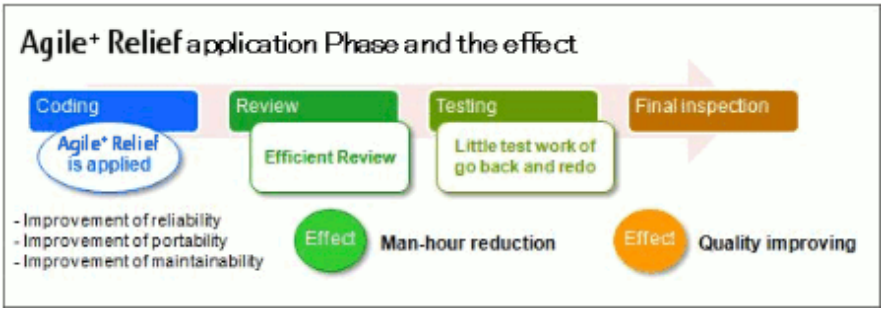
FUJITSU Software
Agile+ Relief J V1.1.1

Customizer Operation Guide

B1WD-3616-01ENZ0(00)
January 2023

Preface

During software development, improving quality at early stage of development process can reduce subsequent revision of the post-process and cut development cost. Agile+ Relief J is a product that provides support for improving the quality of coding process in development process. Through static analysis of the program described by Java language, the problems that are ignored during visual review can be detected. As a result, the product quality can be improved, and the development costs can be reduced.



What Is Agile+ Relief J?

Agile+ Relief J contains Agile+ Relief J Customizer (called customizer hereinafter) and Agile+ Relief J Code check (called Code check hereinafter).

- Customizer

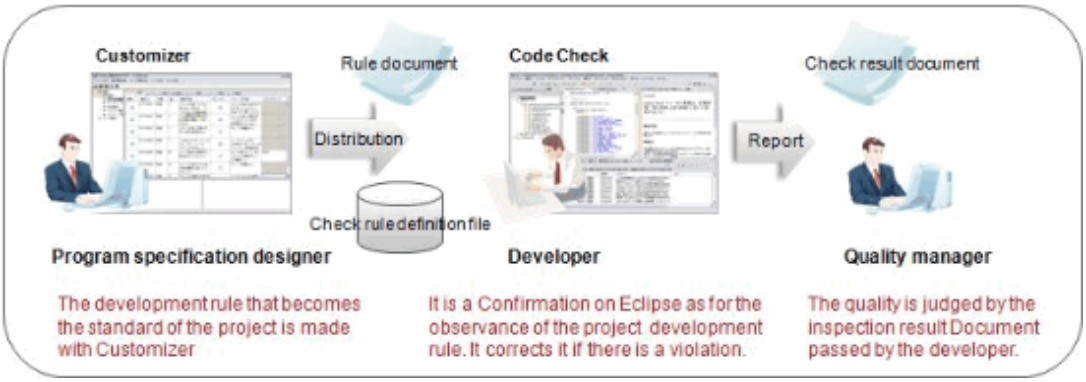
  Create coding rules as the development standard of development project.

- Code check

  Analyze the source codes written by Java language statically to detect violation of coding rules created by customizer.

Users and Roles of Agile+ Relief J

The following figure shows the expected users and their respective roles of Agile+ Relief J.



- Program specification designer

  The program specification designer determines the coding rules as the standard of development project. By understanding the check rules of Agile+ Relief J, the program specification designer creates coding rules by using customizer, and delivers the created coding rules to developer and quality manager.

- Developer

  The developer performs coding according to the coding rules. In addition, the developer checks whether there are rule violations using code check, saves the result in a file, and passes this file to the quality manager.

- Quality manager

  The quality manager finally judges whether the codes comply with the coding rules. The quality manager determines whether the codes are abnormal according to the code check result passed from the developer and confirms validity of coding rules.

The users do not need to be of different roles. The method that "The developer can create coding rules and check whether the written codes comply with the coding rules" is also allowed.

Especially when the developer expands coding rules, full negotiation between users is required.

About This Manual

This manual describes the overview and operation methods of customizer.

For details of code check, please refer to "Agile* Relief J Code check Operation Guide".

Trademarks

- Microsoft, Windows, Windows Server are the registered trademarks or trademarks of Microsoft Corporation in the United States, and other countries
- Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.
- Interstage is the registered trademark of Fujitsu.
- Interstage Studio is a product of Fujitsu.
- The name FindBugs™ and the FindBugs logo are trademarked by The University of Maryland
- Android is the registered trademarks or trademarks of Google Inc.
- The names of other products and services referred to in this document are trademarks of their respective developers and manufacturers.

Contents

Chapter 1 Overview

1.1 What is Customizer Tool

The customizer is a tool for creating the coding rules that are set as the standard for development project. It is assumed to be used by program specification designers.

Through customizer, the selection of apply/not to apply the check rules offered by standard by Agile+ Relief J and modification of the check specification in code check can be performed. In addition, the independent rule of development project that is not provided by Agile+ Relief J can be created.

The coding rules created through customizer are called check rule definition. The check rules are distributed to developers. The developers can check rule violations through code check.

1.2 Files Created by Customizer

The following files can be created through customizer.

- Check rule definition file

  It is a file that defines the coding rules of the development project, and it becomes the input file of code check. For how to specify the check rule definition file during code check, please refer to "Set Check Rule Definition File" of "Code check Operation Guide".

  Two sample files of Check rule definition file are saved under the " Agile+ Relief install folder\PGReliefJava" folder.

  - rule.pgrj

    It is the check rule definition file for performing the general code check of Java application. The rules other than maintainability/ readability are defined as the major check target.

  - android.pgrj

    It is the check rule definition file for performing the code check of Android application. The valid Java general rules and specific rules of Android in Android application are defined as the check target.

  The above-mentioned two check rule definition files are samples only. Please use and edit them appropriately so that the rule definition becomes valid for the development project. In addition, if Agile+ Relief was installed for the first time, the above-mentioned "rule.pgrj" is set to the default check rule definition file for code check.

- Rule document

  The contents of the check rule definition file are output as an html file. The contents can be output to any folder.

  By configuring the shared environment, the coding rules of document base is likely to become well-known in the entire development project.

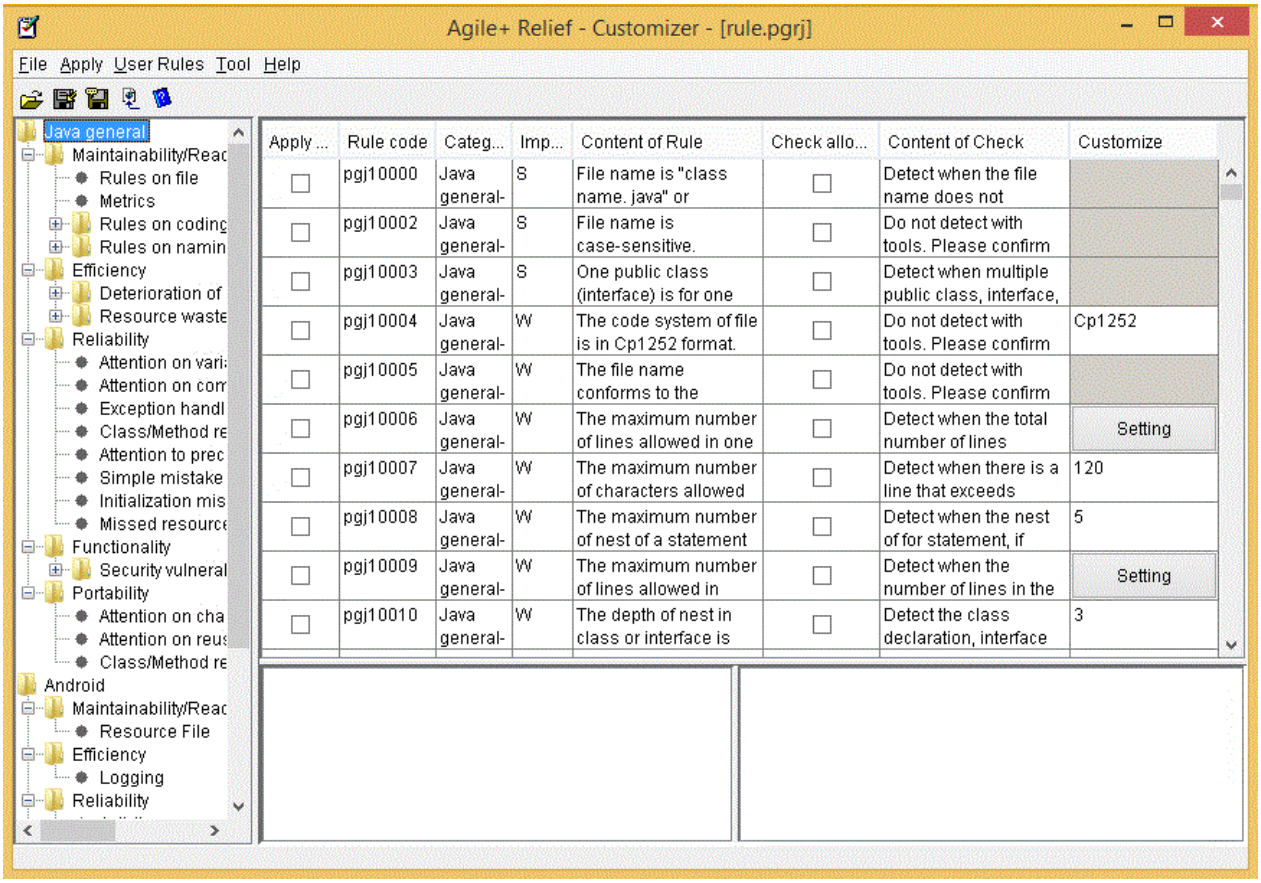
1.3 Cautions

The cautions for using the customizer are as follows.

- Please set the necessary access privilege in the folder and file, such as rule check definition file and document output file that uses tools.
- The environment variable **JAVA_HOME** must be registered. Please register the path of JDK being used to **JAVA_HOME**.

Chapter 2 Screen Component

This chapter describes the screen components of Agile+ Relief J Customizer.



- Title bar

  The names of tools and displayed check rule definition files are displayed.

```
Agile+ Relief J - Customizer - [check rule definition file name]
```

- Menu bar

Menu	Functionality
File	Read/save the check rule definition tool, and terminate the tool.
Apply	Apply and cancel a rule.
User Rules	Add, modify, or delete a user rule, and read/output a template.
Tool	Output the rule document and set the project information.
Help	View the customizer operation manual, description of rule detail, and version information.

- Tool bar

Button	Functionality	Corresponding Menu Operation
	Read rule definition file.	[File]-[Open Rule file]

Button	Functionality	Corresponding Menu Operation
	Save rule definition file.	[File]-[Save]
	Save rule definition file as.	[File]-[Save As]
	Output rule document according to the rule content.	[Tool]-[Output Rule Document]
	Display a manual.	[Help]-[Customizer Operation Guide]

- Rule category tree display column

  The rule category is displayed in the tree structure.

  After a rule category is selected, the rule list that becomes the target will be displayed.

- Rule list display column

  The rules corresponding to the selected rule categories in the rule category tree is displayed.

  The rule application, check application, and modification of importance degree and customization of checking specification are enabled in the rule list.

  For detail of rules, please refer to " Agile+ Relief J Rule Detail Manual".

- Rule content display column

  The content of the rule selected from the rule list is displayed

- Check content display column

  The check content of the rule selected from the rule list is displayed.

Chapter 3 Operation Methods

3.1 Start or Terminate Customizer

This chapter describes the methods of starting and terminating the customizer.

3.1.1 Start Customizer

The customizer can be started through Window menu and Eclipse.

3.1.1.1 Start Customizer through Windows Menu

Start customizer through Windows menu.

<Operations>

1. Select [Agile+ Relief] > [Agile+ Relief J] > [Customizer] from the Start menu.

When the customizer is initially started after installation, the check rule definition file stored under " Agile+ Relief installation folder \PGReliefJava" is automatically loaded.

Later, the check rule definition file that is displayed previously will become the target to be read.

3.1.1.2 Start Customizer through Eclipse

Start customization through the Eclipse.

<Operations>

1. Select [Window] > [Preferences].
2. Select [Agile+ Relief J] from the setting dialog.
3. Click the [Customizer is starting] button of [Setting of check rule definition file].

The file specified in [Check rule definition file] can be read during customizer startup.

3.1.2 Terminate Customizer

Terminate customizer.

If modification is not saved during termination, it is necessary to confirm whether it is saved.

<Operations>

1. Select [File] > [Exit].
2. The dialog box for the confirmation of saving is displayed, the processing is selected.
   - Once the [Yes] button is clicked, the modification will be saved when the customizer is terminated.
   - Once the [No] button is clicked, the modifications will not be saved when the customizer is terminated.
   - Once the [Cancel] button is clicked, the termination operation will be cancelled.

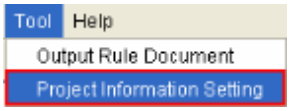
3.2 Set Project Information

This section describes how to set project information and parsing options.

The setting of project name and development tool to be used should be performed in project information setting.

<Operations>

1. Select [Tool] > [Project Information Setting].

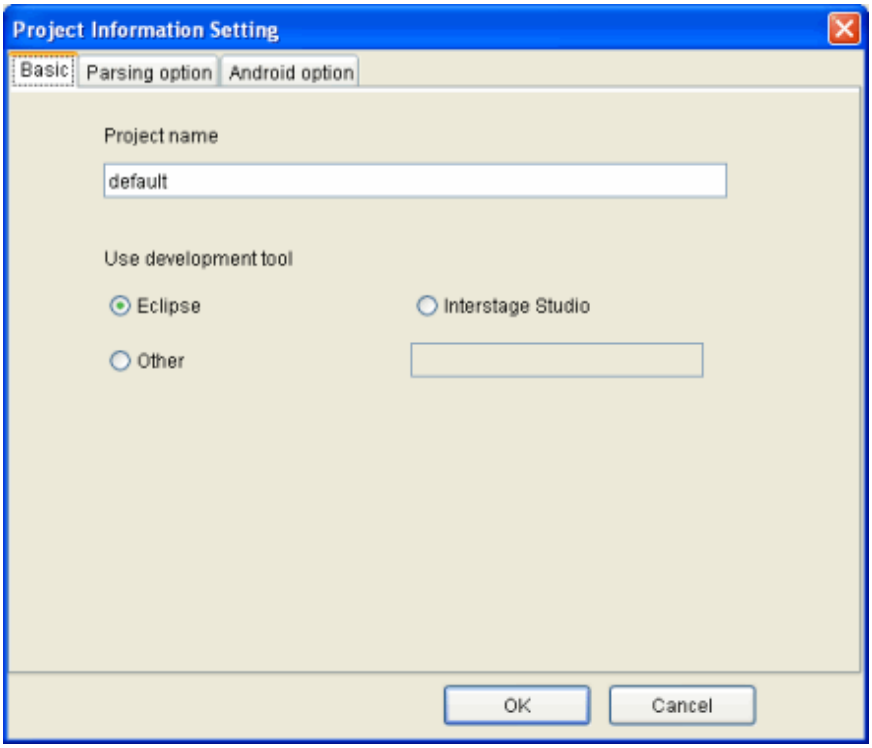


The following sections describe the Configuration Items for each tab.

Click the [OK] button to reflect the settings. Click the [Cancel] button to discard the settings.

3.2.1 Set Project Information

Set the basic information about the project.



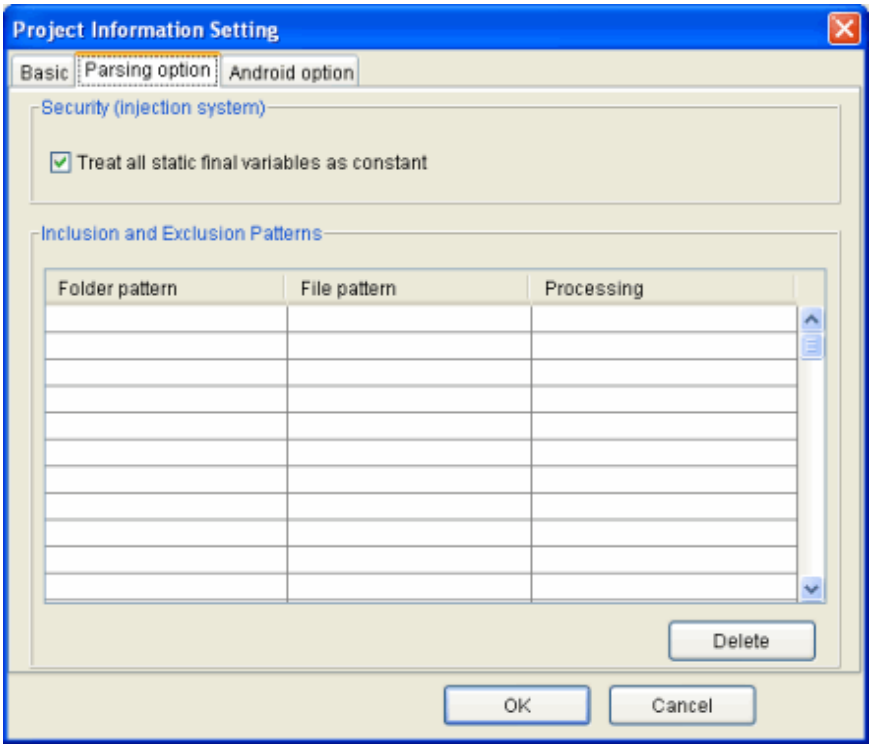
List of Configuration Items

Item	Settings
Project name	Input the project name registered in the check rule definition file. It is used for document output.
Use development tool	Select the development tool being used. The tool name is used for rule document output. - Eclipse - Interstage Studio - Others (Specify the tool names except the above)

Item	Settings
	If "Interstage Studio" is selected, set the rule information used by "Interstage Studio". Please refer to the cautions for relevant rules.

3.2.2 Set Parsing Options

The following items are set as parsing options when code check is executed.



[Security (injection system)]

The security-related options (injection system vulnerability) can be specified.

When this option is being set, please make sure to read "3.6 About Detection of Security vulnerability of Agile* Relief J" of " Agile* Relief J Rule Detail Manual".

- All static final variables are set as constant for processing

  After the option is enabled, all static final variables can be set as constant and are not as detected object.

  <Applicable rules>

  pgj10402, pgj10403, pgj10404, pgj10405, pgj10406, pgj10407, pgj10408, pgj10409, pgj10410, pgj10413, pgj10414

Example

Describe by the detection example of file injection.

In the Example 1, the constructor invocation of the java.io.File class in the sixth line is detected as file injection. The STR2 in this invocation is recognized as a non-constant.

However, after this option is enabled, all the variables declared in static final can be processed as constant. Therefore, the STR2 is also recognized as a constant, as described in example 2, where the constructor invocation of java.io.File class in the sixth line cannot be detected.

Example 1) When the option "Process all static final variables as constant" is disabled

```
1   static final String STR1 = "abc";                          // constant
2   static final String STR2 = System.getProperty( "abc" );    // non-constant
3       :
4   void proc() {
5       File f1 = new File(STR1);  // obey
6       File f2 = new File(STR2);  // violation (file injection)
7           :
8   }
```

Example 2) When the option "Process all static final variables as constant" is enabled

```
1   static final String STR1 = "abc";                          // constant
2   static final String STR2 = System.getProperty( "abc" );    // processed as a constant
3       :
4   void proc() {
5       File f1 = new File(STR1);  // obey
6       File f2 = new File(STR2);  // obey => STR2 that is processed as a constant is detected.
7           :
8   }
```

[Inclusion and Exclusion Patterns]

The specific folders and files can be excluded from target for code check. Part of them is contained in the target for code check.

Please use the asset that is generated automatically through the framework when it is excluded from target for code check.

The condition combining with the folder pattern, file pattern and processing can be specified. The wildcard can be used in the folder pattern and file pattern.

List of Configuration Items

Item	Settings
Folder pattern	Specify the folder for processing. The following wildcard can be used for specifying the folder pattern. *: indicates the character string of any length. The path separator is not included. **: indicates any folder more than 0. When it is ended with \, it is considered to be equal.
File pattern	Specify the file for processing. The following wildcard can be used for specifying the file pattern. *: indicates the character string of any length.
Processing	Specify the processing of the situation when the folder pattern and file pattern are conformed. - Excluded: Exclude the case of conformance - Included: Include the case of conformance

Note

The errors in the specification of folder pattern and file pattern may occur in following cases.

- Folder pattern
  - "\\" or "//" is contained in other positions except the initial.
  - The folder name is composed of " .." or "." only.
  - The prohibited characters ('?', '"', '<', '>', or '|') are contained.
  - ":" is contained in other places except the drive name.
- File pattern
  - The prohibited characters ('¥', '/', ':', '?', '"', '<', '>', or '|') are contained.

Example

The example of excluding folder/file of the asset with the following structure is shown.

```
C:\
+-- workspace
    +-- project1
        +-- gen
        |   +-- src
        |       +-- pk1
        |           +-- Clazz01.java
        |           +-- XXX01Msg.java
        +-- src
            +-- pk1
                +-- Clazz02.java
                +-- XXXM02Msg.java
```

Example 1) Excluding the file under the specific folder

Settings

Folder Pattern	File Pattern	Processing
\gen\src	(not specified)	Excluded

According to the above specification, all the files under the gen\src folder will be the excluded object.

```
C:\workspace\project1\gen\src\pk1\Clazz01.java       exclude
C:\workspace\project1\gen\src\pk1\XXX01Msg.java      exclude
C:\workspace\project1\src\pk1\Clazz02.java           include
C:\workspace\project1\src\pk1\XXXM02Msg.java         include
```

Example 2) Excluding specific files

Settings

Folder Pattern	File Pattern	Processing
(not specified)	*Msg.java	Excluded

According to the above specification, all the files whose suffix is Msg.java will be the excluded object.

```
C:\workspace\project1\gen\src\pk1\Clazz01.java       include
C:\workspace\project1\gen\src\pk1\XXX01Msg.java      exclude
C:\workspace\project1\src\pk1\Clazz02.java           include
C:\workspace\project1\src\pk1\XXXM02Msg.java         exclude
```

Example 3) Excluding the specific files under specific folder

Settings

Folder Pattern	File Pattern	Processing
\gen\src	*Msg.java	Excluded

According to the above specification, in the file under the folder with the name of gen\src, all the files whose suffix is Msg.java will be the excluded object.

```
C:\workspace\project1\gen\src\pk1\Clazz01.java       include
C:\workspace\project1\gen\src\pk1\XXX01Msg.java      exclude
C:\workspace\project1\src\pk1\Clazz02.java           include
C:\workspace\project1\src\pk1\XXXM02Msg.java         include
```

Example 4) Excluding the files in the specified folder but including the specified files

Settings

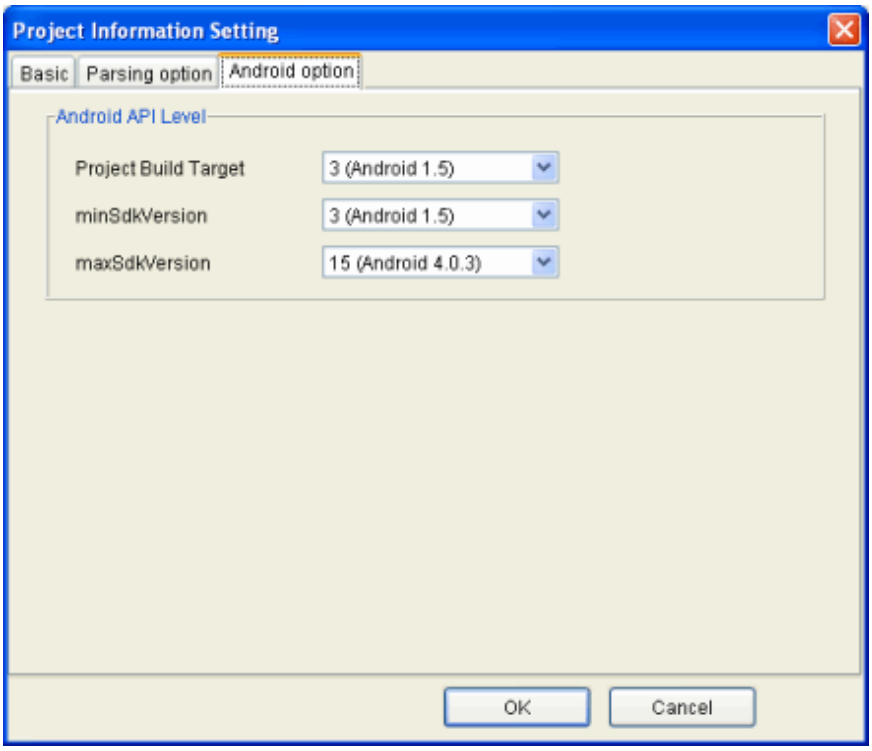
Folder Pattern	File Pattern	Processing
\gen\src		Excluded
\gen\src	*Msg.java	Included

According to the above specification, the file under the folder with the name of gen\src will be the excluded object, and all the files whose suffix is Msg.java will be the included object.

```
C:\workspace\project1\gen\src\pk1\Clazz01.java       exclude
C:\workspace\project1\gen\src\pk1\XXX01Msg.java      include
C:\workspace\project1\src\pk1\Clazz02.java           include
C:\workspace\project1\src\pk1\XXXM02Msg.java         include
```

3.2.3 Set Android Options

Set options for parsing Android program.



[Android API Level]

Specify the API Level to limit the scope of code check for Android program. Unless an appropriate API Level is specified, checking operation could be time consuming, possibly detecting an unnecessary pgr item.

List of Configuration Items

Item	Settings
Project Build Target	Specify the API Level that is used to build Android application as the check target.
minSdkVersion	Specify the lowest API Level for Android application operation as the check target.
maxSdkVersion	Specify the highest API Level for Android application operation as the check target.

3.3 Read/Save Rules

This section describes how to read and save the check rule definition files and how to output the rule document.

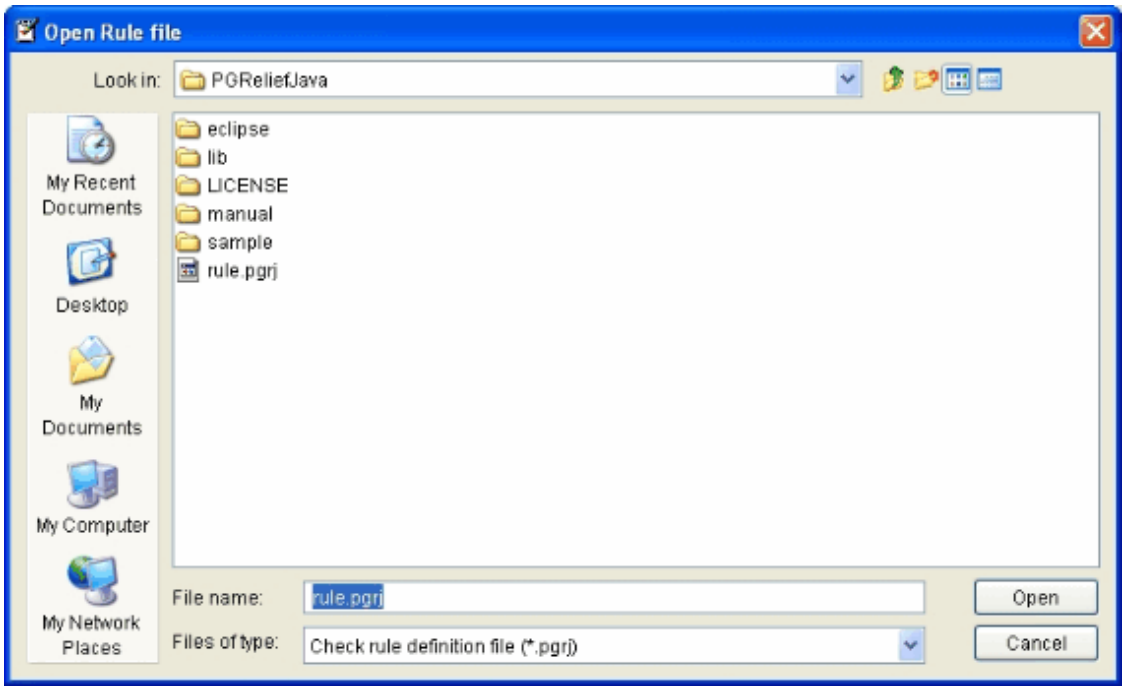
3.3.1 Read Rules

Read the check rule definition files, and display the content of check rule definition in the tree or list format.

<Operations>

1. Select [File] - [Open Rule file]

2. Specify the check rule definition file to be read.



3. Select the reading process.
   - Click the [Open] button to read the specified files.
   - Click the [Cancel] button to cancel file reading.

3.3.2 Overwrite Rules

Overwrite the displaying rule content.

<Operations>

1. Select [File] > [Save].

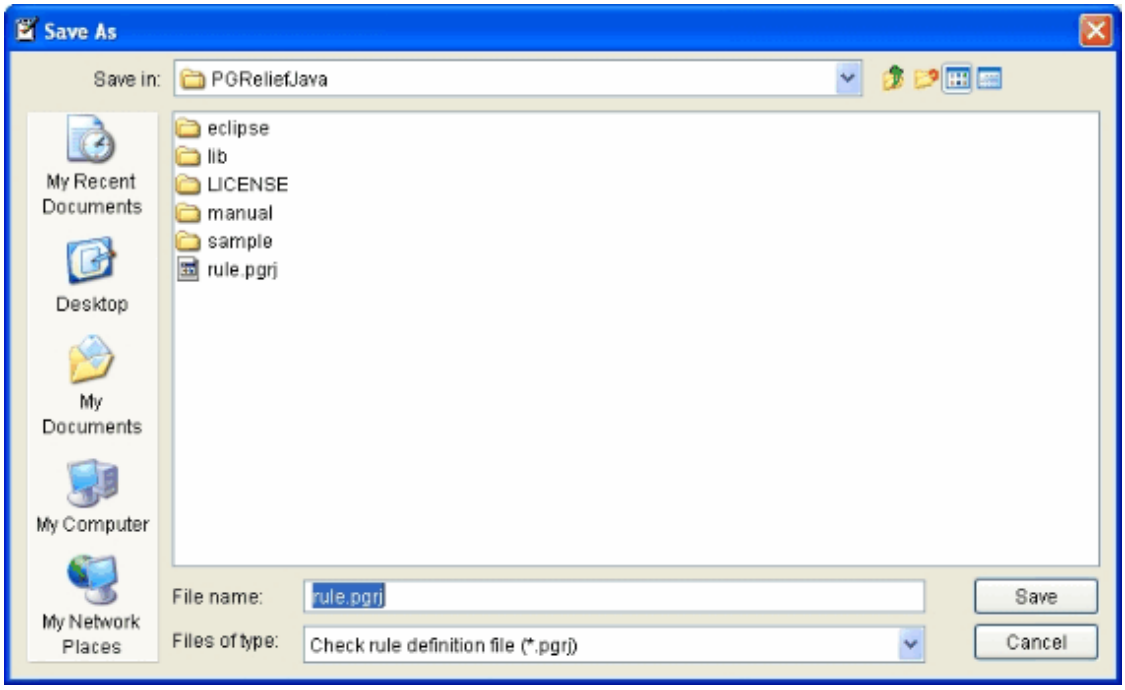
3.3.3 Save Rule As

Save the displaying rule definition as a file with another name.

<Operations>

1. Select [File] > [Save As].

2. As the [Save As] dialog box is displayed, set the file name and select [Save].



- Click the [Save] button to save the customized data as a check rule definition file.
- Click the [Cancel] button to cancel the saving.

3.4 Create Coding Rules

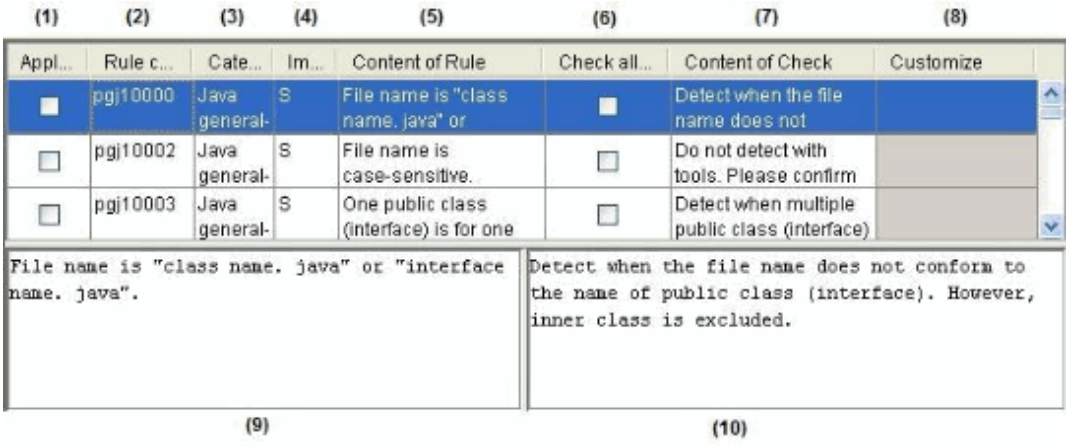
This section describes how to create the coding rules as the standard of development project based on the rules that are offered by standard in Agile+ Relief J.

3.4.1 Display Rule List

Select the rule category from the rule category tree to display the rule list.

The rule category tree is in hierarchical structure. Therefore, as the category of each hierarchy is selected, it can be limited.

The rule list is displayed as the following image.



The items to be displayed for the rule list are as follows.

The item whose Changeable(Yes/No) column is set to "Yes" can be modified.

Display item of rules list

Item No.	Item	Description	Changeable (Yes/No)
(1)	Apply Rule	It is a checkbox for specifying whether to apply the rules. Check the [Apply Rule] checkbox of the rule code to be applied. The checked rules are output as a rule document.	Yes
(2)	Rule code	Display the rule code.	No
(3)	Category	Display the rule category.	No
(4)	Importance degree	Display the importance degree of the rule. S: rules required for correction of program. W: rules recommended for correction of program.	Yes
(5)	Content of Rule	Display the rule contents.	No
(6)	Check allowed	It is a checkbox for specifying whether the applied rules are checked by code check tool. Select the [Apply Rule] checkbox of the rule code being checked. The rules that cannot be detected by the tool cannot be selected.	Yes
(7)	Content of Check	Display the check contents.	No
(8)	Customize	Detailed setting of rule. Select the customization items of the rule for detailed setting. For details, please refer to "3.4.5 Customizing Checking Specification".	Yes
(9)	Rule content display column	Display "Rule content" of the selected rule.	No
(10)	Check content display column	Display "Check content" of the selected rule.	No

3.4.2 Apply/Cancel Rules

Set rule application or cancellation for coding rules.

3.4.2.1 Apply Rules

Apply the selected rules as the coding rules.

<Operations>

1. Select the checkbox of rule application column from the rule list.

3.4.2.2 Cancel Rule Application

Exclude the selected rule from the coding rules.

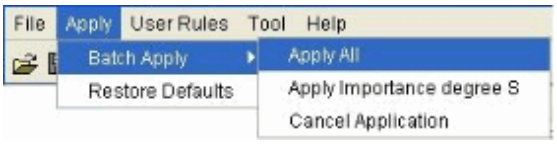
<Operations>

1. The selection on the checkbox of rule application column is removed from the rule list.

3.4.2.3 Batch Application of Rules

Perform batch application for the rule of category selected from the rule category tree.

The rule that exists in an exclusive relation is excluded.



<Operations>

1. Select the category that is applied in batch from the rule category tree.
2. Select [Apply] >[Batch Apply] - [Apply All] or select [Apply All] from the pop-up menu displayed by right-click.
3. Once the confirmation dialog is displayed, select [Yes].

- The operation when the [Yes] button is clicked

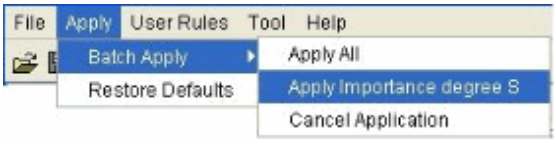
  Change the [Apply Rule] and [Check allowed] checkbox of all rules under the selected category into selected state.
  *In case of rules for which the check is disallowed, change only the [Apply Rule] checkbox into selected state.

- The operation when the [No] button is clicked

  All applications are cancelled.

3.4.2.4 Apply the Rules of Importance degree S

Apply the rules of importance degree S from the category selected in the rule category tree.



<Operations>

1. Select the category that is applied in batch from the rule category tree.
2. Select [Apply] > [Batch Apply] - [Apply Importance degree S] or select [Apply Importance degree S] from the pop-up menu displayed by right-click.
3. Once the confirmation dialog is displayed, select [Yes].

- The operation when the [Yes] button is clicked

  Change the [Apply Rule] and [Check allowed] checkbox of all rules of importance degree S in the rule list into selected state.
  *In case of rules for which the check is disallowed, change only the [Apply Rule] checkbox into selected state.

- The operation when the [No] button is clicked

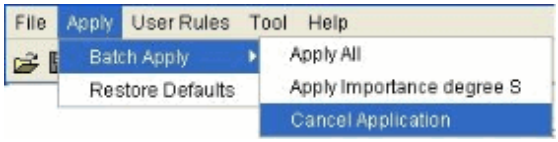
  All operations are cancelled.

3.4.2.5 Batch Cancellation of Rule Application

Exclude all rules from the coding rules.

<Operations>

1. Select the category to be cancelled in batches from the rule category tree.
2. Select [Apply] > [Batch Apply] - [Cancel Application] or select [Batch Apply] - [Cancel Application] from the pop-up menu displayed by right-click.



3. Once the cancellation of application is confirmed, select [Yes].

- The operation when the [Yes] button is clicked

  Change the [Apply Rule] checkbox and [Check allowed] checkbox of all rules into unselected state.

- The operation when the [No] button is clicked

  The application cancellation is cancelled.

3.4.3 Modify Importance degree

Modify the importance degree of rules.

The following two types of importance degree can be selected for a rule.

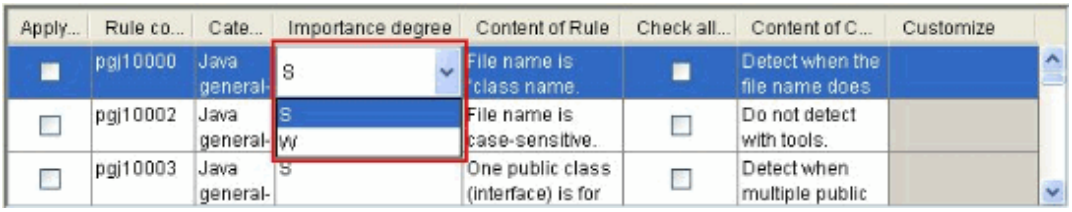
Importance Degree

S: Rules required for correcting programs.

W: Rules recommended for correcting programs.

<Operations>

1. Click the cell of the importance degree of the rule to be modified.
2. Select the importance degree from the selection list.



3.4.4 Apply/Cancel Check

The application of code check according to Agile* Relief J or application cancellation can be set in check rules.

After the check is applied, it becomes the target for the code check according to Agile* Relief J.

3.4.4.1 Apply Check

Apply the selected rules as the target for code check.

<Operations>

1. Select the checkbox of check application column from the rule list.

3.4.4.2 Cancel Check Application

Exclude the selected rules from the target for code check.

<Operations>

1. The check on the checkbox of check application column is removed from the rule list.

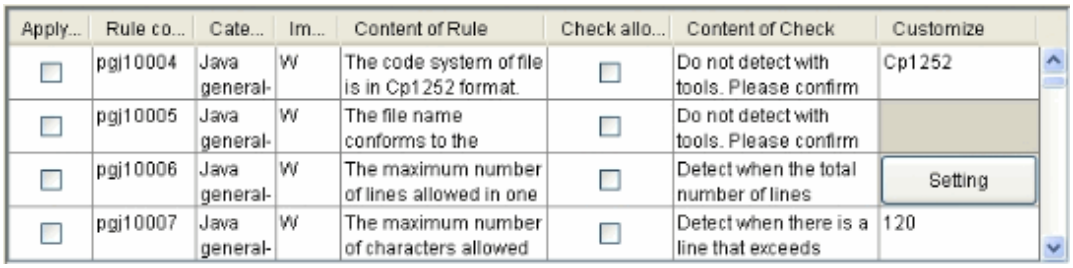
3.4.5 Customizing Checking Specification

The content of check can be customized for all check rules.

The customization methods are as follows.

- Directly double-click the customized column to input contents.
- Select the content from project list displayed in the customized column.

- Click the [Setting] button displayed in the customized column, and the detailed setting dialog box is displayed to enable setting.



The following section describes the details about the customization methods.

3.4.5.1 Input Numeric Values

Input the numeric values that will be used for check.

Input is enabled by double-clicking the customized column of the applicable rules.

<Applicable Rules>

pgj10007, pgj10013, pgj10008, pgj10031, pgj10052, pgj10056, pgj10010,
pgj10011, pgj10057, pgj20218*

* pgj20218 can input an '*'(asterisk) other than numeric value.

3.4.5.2 Input the Range of Numeric Values

Input the range of numeric value used for check and separate the minimum and maximum numeric values by comma.

Input is enabled by double-clicking the customized column of the applicable rules.

<Applicable Rules>

pgj10096, pgj10114, pgj10120, pgj10144

3.4.5.3 Input Strings

Input the character strings that will be used for check. Multiple character strings are separated by colon (,).

Input is enabled by double-clicking the customized column of the applicable rules.

<Applicable Rules>

pgj10076

3.4.5.4 Select Item

Select an item from the selection list.

Click the customized column of the applicable rule and select an item from the combo box.

<Applicable Rules>

pgj10004, pgj10019, pgj10061, pgj10068, pgj10078, pgj10214, pgj10231

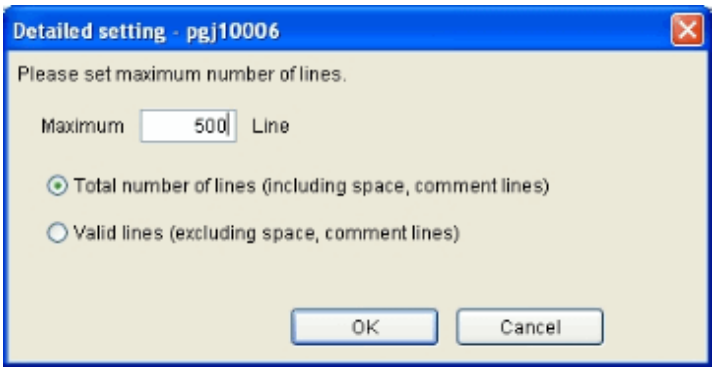
3.4.5.5 Specify Maximum Number of Lines

Set the maximum number of lines and set whether the maximum number of lines is the total number of lines or valid lines.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10006, pgj10009



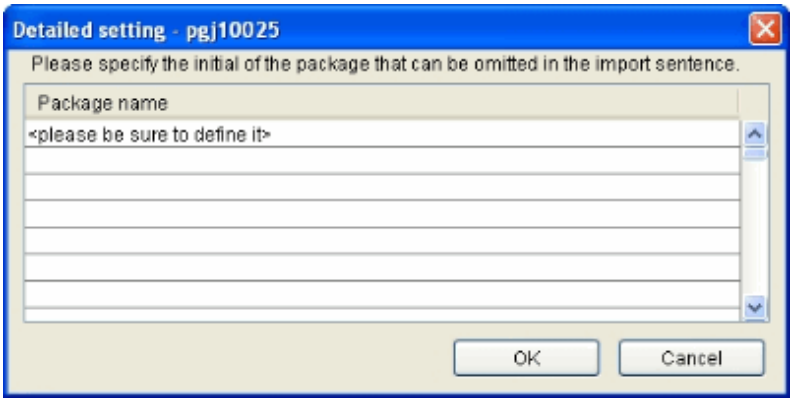
3.4.5.6 Specify the Omissible Package

Set the list of initial characters of package becomes omissible in the import statement.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10025



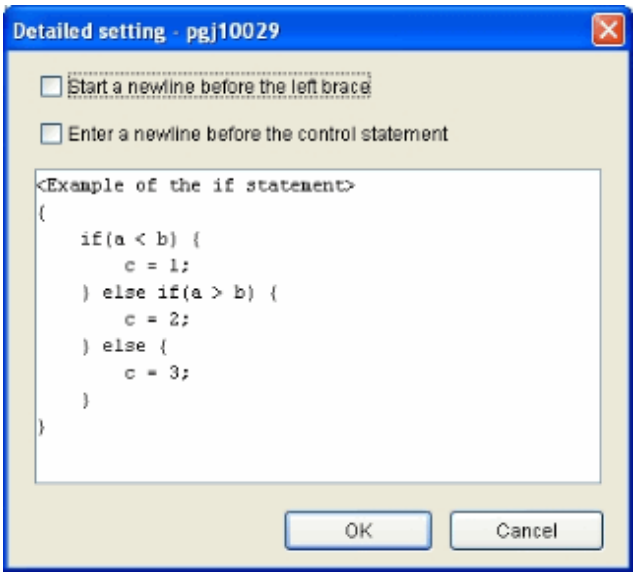
3.4.5.7 Specify Coding Style

Set the format for the control syntax {}.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10029, pgj10030, pgj10032, pgj10033, pgj10036, pgj10049, pgj10066



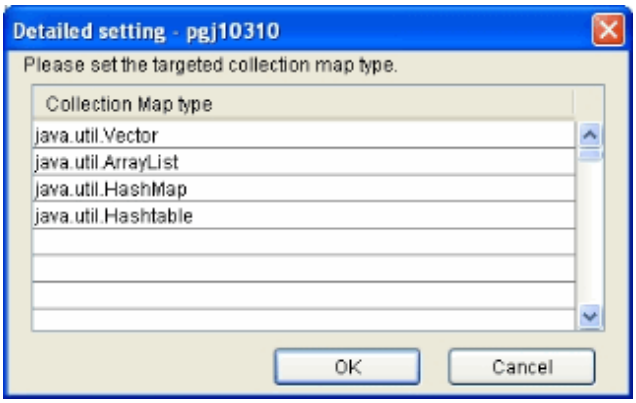
3.4.5.8 Specify Collection/Map Type

Set the list of collection types or map types that become the target.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10310



3.4.5.9 Specify Allowed Characters

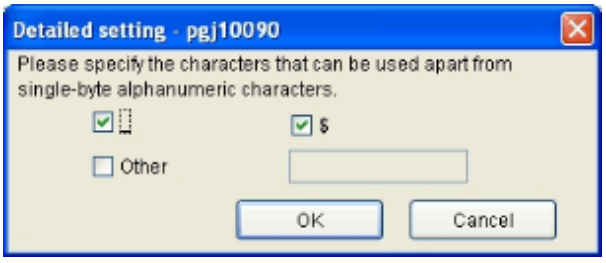
Set the characters except single-byte alpha-numeric characters that can be used as the identifier of each element.

When specifying multiple characters in "Others", please separate these characters by comma.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10090, pgj10109, pgj10116, pgj10121, pgj10125, pgj10134, pgj10150



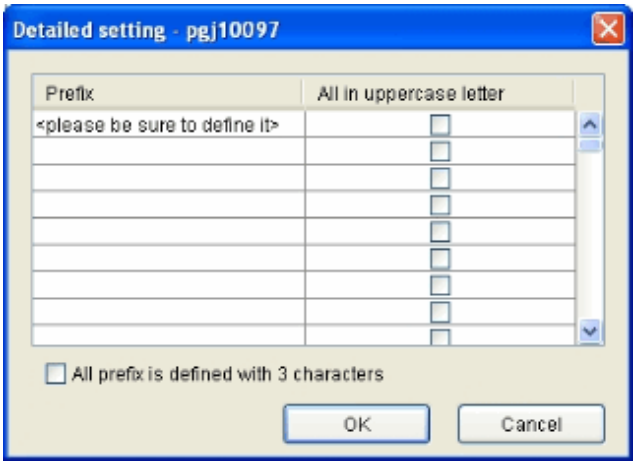
3.4.5.10 Specify Prefix

Set the list of valid prefixes that are set as the prefixes of the class/interface.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10097



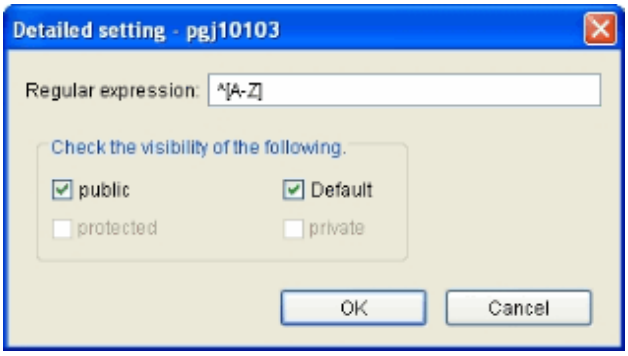
3.4.5.11 Specify Naming Rule of Java Elements (a)

Specify the naming rules of Java elements of class etc. through regular expression. In addition, the visibility of the element that becomes the target for code check is also specified.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10103, pgj10104, pgj10105, pgj10106, pgj10128, pgj10129, pgj10130,
pgj10131, pgj10147, pgj10148



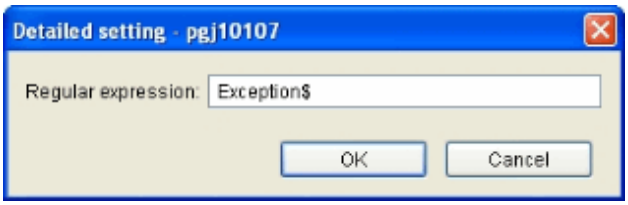
3.4.5.12 Specify Naming Rule for Java Elements (b)

Specify the naming rules of Java elements of class etc. through regular expression.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10107, pgj10108, pgj10132, pgj10133, pgj10149, pgj10151



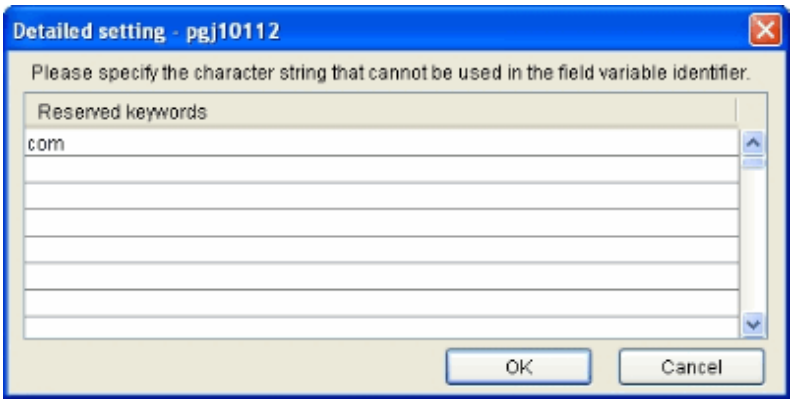
3.4.5.13 Specify reserved keywords

Set the Java reserved keywords as the identifier.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10112



3.4.5.14 Specify Local Variable Name (Single-Character Name)

Set the local variable name and type of one character, as well as the use of inheritance destination class.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

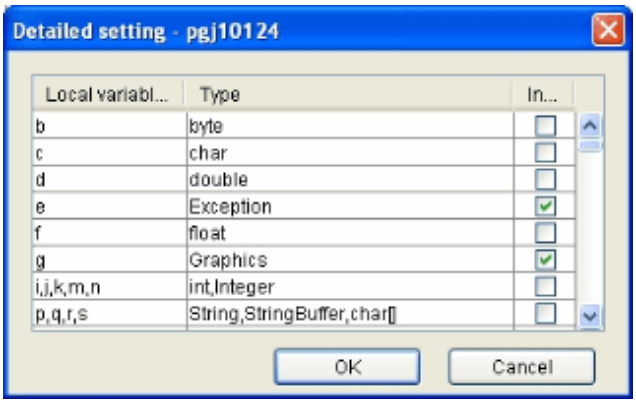
When specifying multiple local variable names and types, please input and separate them by comma.

When the use of inheritance destination class is allowed, please select the checkbox of inheritance column.

The number of inherited hierarchies that becomes the check target is three.

<Applicable Rules>

pgj10124



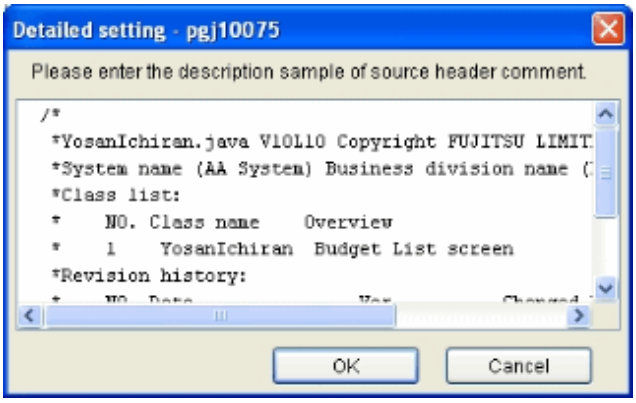
3.4.5.15 Specify Source Header Comment

Please set the example of source header comment.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10075



3.4.5.16 Specify Javadoc Comment Tag

Set the Javadoc comment tag that becomes the target of javadoc comment check.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10084, pgj10085, pgj10086, pgj10087

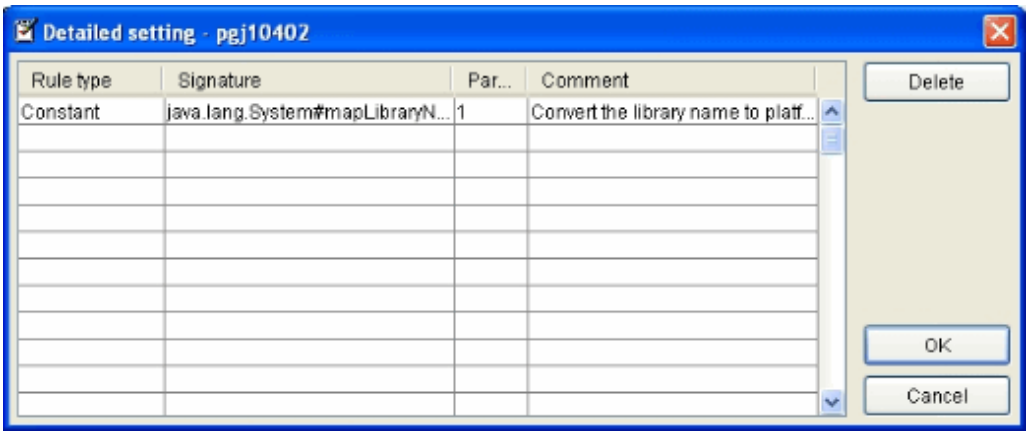
3.4.5.17 Specify Security Rules

Set security rules.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10402, pgj10403, pgj10404, pgj10405, pgj10406, pgj10407, pgj10408, pgj10409, pgj10410, pgj10413, pgj10414, pgj10433



List of Configuration Items

Item	Description
Rule type	Specify the constant or constant parameter. In case of constant, the specified signature is considered as the constant. In case of constant parameter, the temporary variable of the specified signature is considered as the constant.
Signature	Specify the field or procedure (method or constructor) of Java. - Specify '#' for the separation with class name. - The field writes only the names. - The procedure writes with brackets. - The parameter of procedure only enumerates type by separating with ','(comma). In addition, please use "[]" for the variable parameter instead of "…". - Specify the parameter of the procedure excluding receiver(this) of the instance method. - The constructor name is specified with <init>. Example 1) Specification of field java.lang.System#out Example 2) Specification of method java.util.Map#put(java.lang.Object, java.lang.Object) Example 3) Specification of method (Specification of variable parameter) java.io.PrintStream#println(java.lang.String, java.lang.Object[]) Example 4) Specification of constructor java.lang.Integer#<init>(int)
Parameter specification	When specifying the procedure in the signature, specify the order of the applicable parameter from the parameter of procedure. In case of multiple specifications, enumerate by ',' (comma) separation.
Comment	Specify comments for the created rules. The code check will not be affected.

- When the rule category is constant

  The values specified in signature are regarded as constant.

When the procedure is specified in signature, the return value of this procedure is considered as constant.

In addition, the constant judgment of the parameter for that condition that considers the procedure as a constant can be specified. When the orders of parameter specified through parameter specification are all constants, the return value of procedure is considered as constant.

When field is specified in signature, the field will be considered as a constant.

Example

The specification examples are as follows.

Example 1) When the field is considered as a constant

Rule Type	Signature	Parameter Specification	Comment
Constant	p.C#f		

Result

```
Object o = p.C.f; // constant
```

Example 2) When the return value of method is considered as a constant

Rule Type	Signature	Parameter Specification	Comment
Constant	p.C#f(java.lang.String,java.lang.String)		

Results

```
Object o;
o = p.C.f("literal", "literal"); // constant
o = p.C.f(input(), input()); // constant
o = p.C.f("literal", input()); // constant
```

Example 3) When the first parameter is a constant, and the return value of method is considered as a constant

Rule Type	Signature	Parameter Specification	Comment
Constant	p.C#f(java.lang.String,java.lang.String)	1	

Results

```
Object o;
o = p.C.f("literal", "literal"); // constant
o = p.C.f(input(), input());
o = p.C.f("literal", input()); // constant
```

Example 4) When the first and the second parameters are constants, and the return value of the method is considered as a constant

Rule Category	Signature	Parameter Specification	Comment
Constant	p.C#f(java.lang.String,java.lang.String)	1,2	

Results

```
Object o;
o = p.C.f("literal", "literal"); // constant
o = p.C.f(input(), input());
o = p.C.f("literal", input());
```

- When the rule category is constant parameter

  The parameter of the specified procedure is considered as a constant.

  All parameters that match to the order specified by parameter specification are considered as constants.

Example

The specification examples are as follows.

Example 1) Consider all the parameters as constants.

Rule Type	Signature	Parameter Specification	Comment
Constant parameter	p.C#f(java.lang.String,java.lang.String)		

Results

```
package p;
class C {
  void f(String arg1, String arg2) {
    output(arg1); // arg1 is processed as a constant
    output(arg2); // arg2 is processed as a constant
  }
...
```

Example 2) Consider the first parameter as a constant.

Rule Category	Signature	Parameter Specification	Comment
Constant parameter	p.C#f(java.lang.String,java.lang.String)	1	

Results

```
package p;
class C {
  void f(String arg1, String arg2) {
    output(arg1); // arg1 is processed as a constant
    output(arg2);
  }
...
```

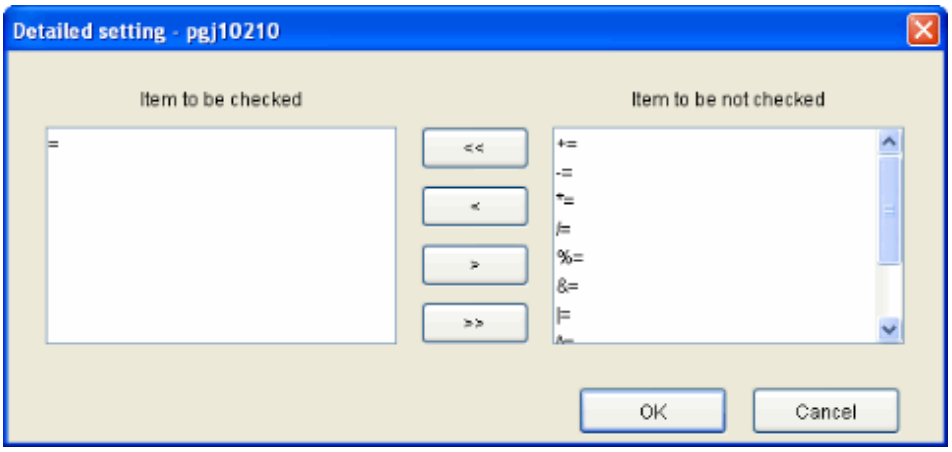
3.4.5.18 Specify assignment operator

The assignment operator to be checked is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10210



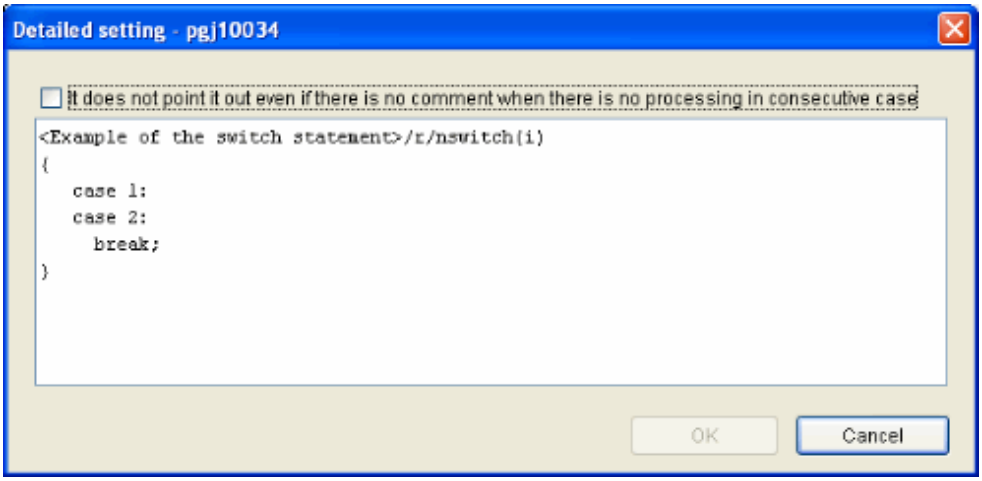
3.4.5.19 case without processing is excluded from the check

Whether case without processing is checked is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10034



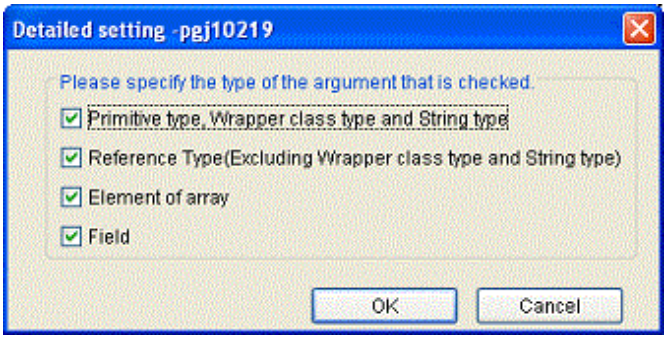
3.4.5.20 Specify the type of the argument

The type of the argument that is checked is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

＜Applicable Rules＞

pgj10219



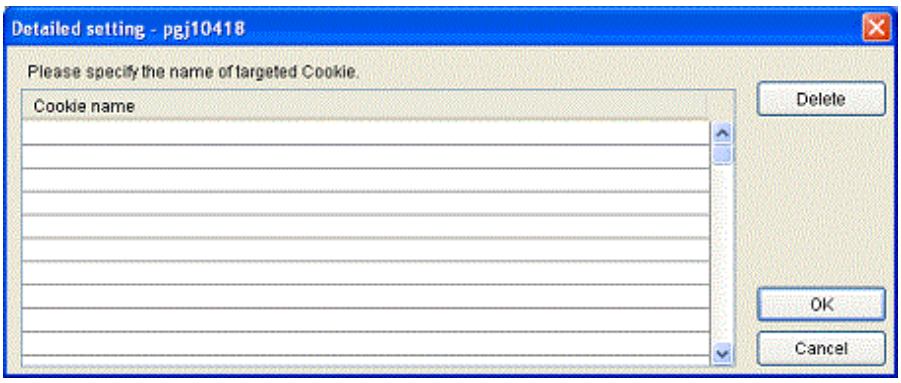
3.4.5.21 Specify cookie name

The cookie name to be checked is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10418, pgj10419, pgj10420, pgj10421, pgj10422



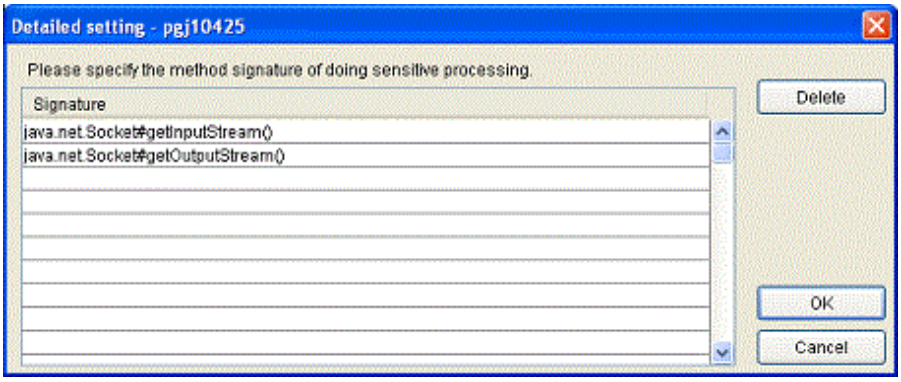
3.4.5.22 Specify sensitive processing

The signature of sensitive processing to be checked is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10425



List of Configuration Items

Item	Description
Signature	Specify the procedure (method or constructor) of Java. - Specify '#' for the separation with class name. - The procedure writes with brackets. - The parameter only enumerates type by separating with ','(comma). - Use "[]" for the variable parameter instead of "…" when you specify the variable argument for the parameter. - Use the real type for the variable parameter instead of the type variable name, when you specify the type variable for the parameter. For example, specify Object type when you specify type variable T of the type parameter <T> for the parameter. - Specify the parameter of the procedure excluding receiver(this) of the instance method.

Item	Description
	- The constructor name is specified with <init>. Example) java.net.Socket#getInputStream()

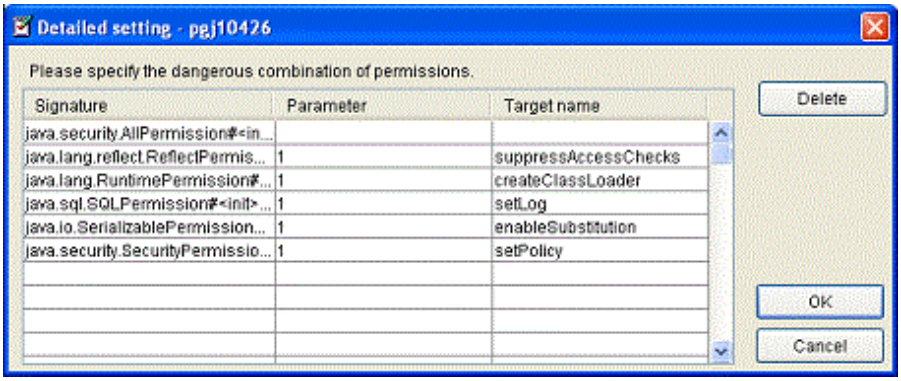
3.4.5.23 Specify dangerous combination of permissions.

The dangerous combination of permissions to be checked is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10426



List of Configuration Items

Item	Description
Signature	Specify the procedure (method or constructor) of Java that generates the instance of the permission type. - Specify '#' for the separation with class name. - The procedure writes with brackets. - The parameter only enumerates type by separating with ','(comma). When the parameter is not limited, '*'(asterisk) is specified. - Use "[]" for the variable parameter instead of "…" when you specify the variable argument for the parameter. - Use the real type for the variable parameter instead of the type variable name, when you specify the type variable for the parameter. For example, specify Object type when you specify type variable T of the type parameter <T> for the parameter. - Specify the parameter of the procedure excluding receiver(this) of the instance method. - The constructor name is specified with <init>. Example) java.lang.reflect.ReflectPermission#<init>(java.lang.String)
Parameter	Specify the index of the argument that specifies the target name. All invocations are detected when not specifying it.
Target name	Specify the target name that becomes a dangerous permission. When plurals are specified, it enumerates it by the ','(comma) delimitation.

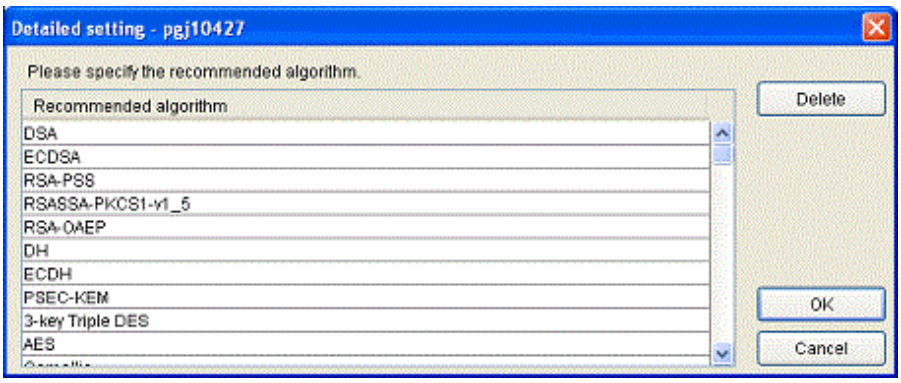
3.4.5.24 Specify recommended algorithm

The algorithm name to not be detected is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10427, pgj10428



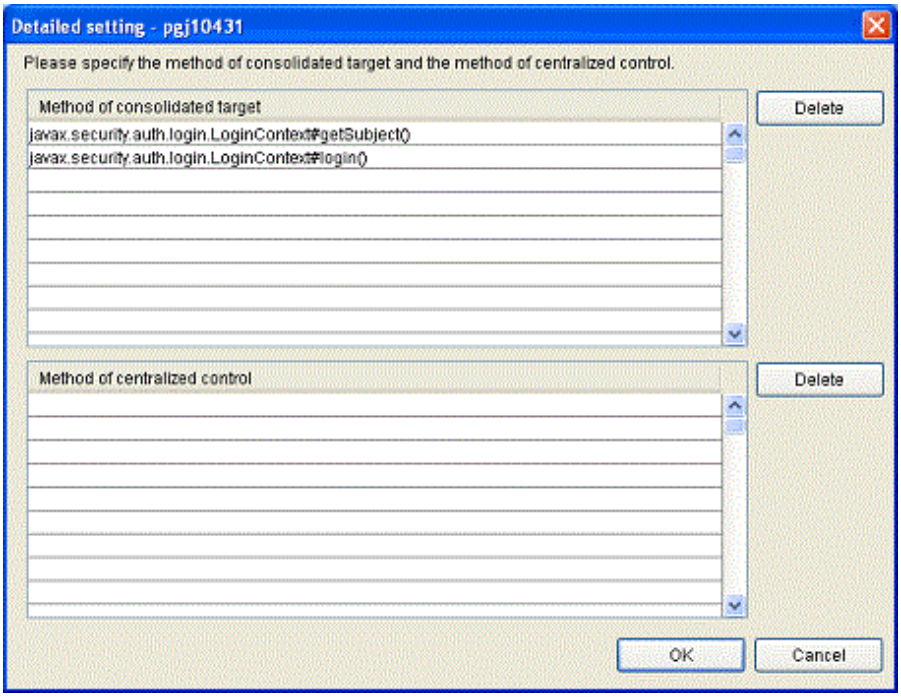
3.4.5.25 Specify the method of consolidated target and the method of centralized control

The method of consolidated target and the method of centralized control is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj10431, pgj10432



List of Configuration Items

Item	Description
Method of consolidated target	Specify the procedure (method or constructor) of Java that is unified. - Specify '#' for the separation with class name. - The procedure writes with brackets.

Item	Description
	- The parameter only enumerates type by separating with ','(comma). - Use "[]" for the variable parameter instead of "…" when you specify the variable argument for the parameter. - Use the real type for the variable parameter instead of the type variable name, when you specify the type variable for the parameter. For example, specify Object type when you specify type variable T of the type parameter <T> for the parameter. - Specify the parameter of the procedure excluding receiver(this) of the instance method. - The constructor name is specified with <init>. Example) javax.security.auth.login.LoginContext#getSubject()
Method of centralized control	Specify the procedure of centralized control(method or constructor) of Java .The specification method is the same as "Method of consolidated target".

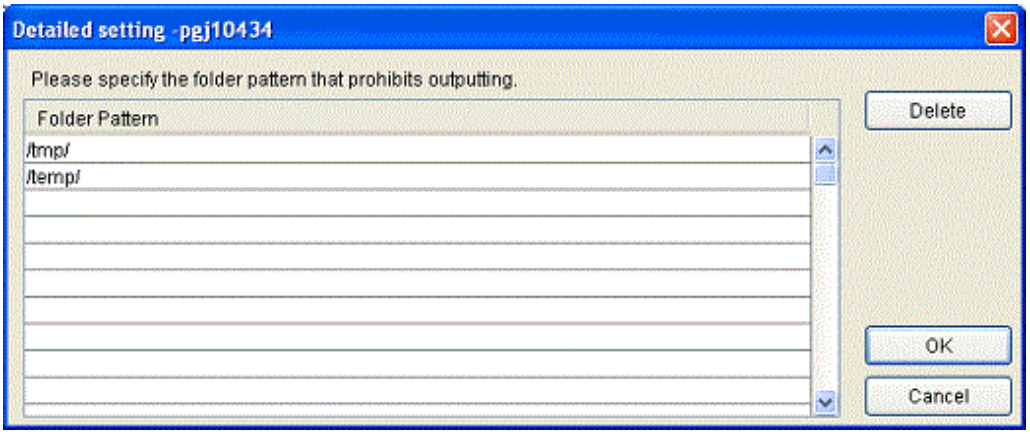
3.4.5.26 Specify the folder pattern that prohibits outputting

The folder pattern that is detected is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

＜Applicable Rules＞

pgj10434



List of Configuration Items

Item	Description
Folder Pattern	Specify the folder pattern that is detected. The folder pattern can be specified with the following wildcards. * : It means the character string of the arbitrary length. The separator of the path is not included. ** : It means the 0 or more folders. It is the same meaning, when the character of the end is \ or /. Specify "/temp/" when all folders under the temp folder are detected. All folders become detected when not specifying it.

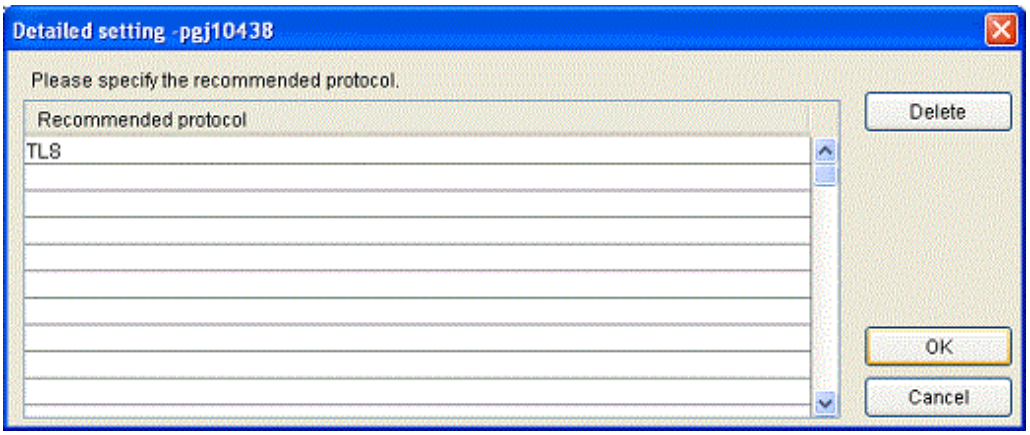
3.4.5.27 Specify recommended protocol

The protocol name to not be detected is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification.

＜Applicable Rules＞

pgj10438



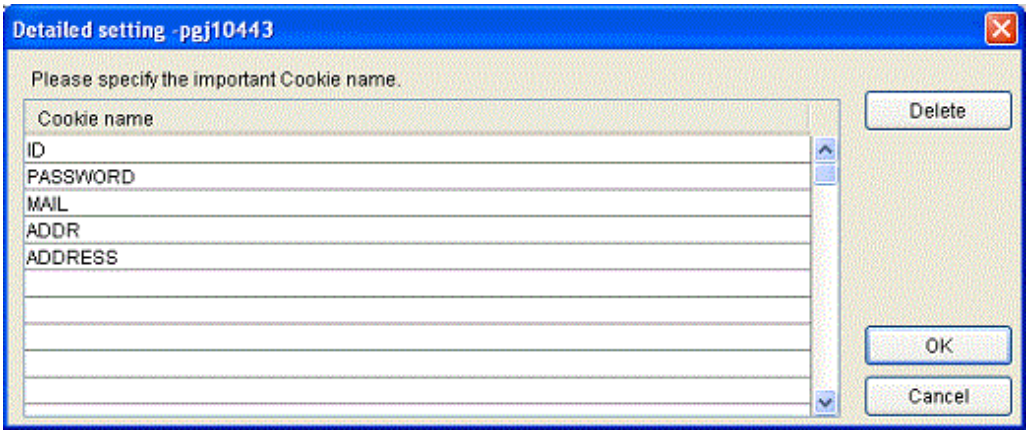
3.4.5.28 Specify the important Cookie name

The Cookie name to be detected is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification. All Cookies become detected when not specifying it.

＜Applicable Rules＞

pgj10443



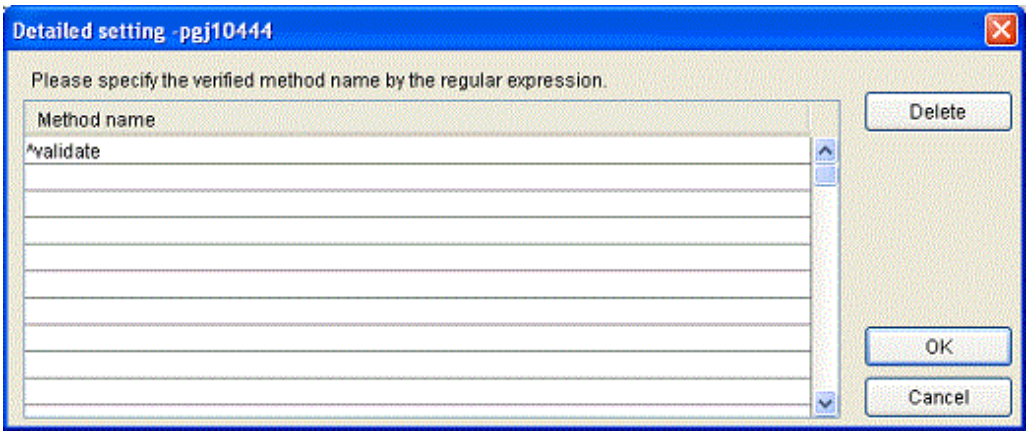
3.4.5.29 Specify the verified method

The method to verify is set by the regular expression.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification. When the method that starts from validate is defined, it is set as ^validate.

＜Applicable Rules＞

pgj10444



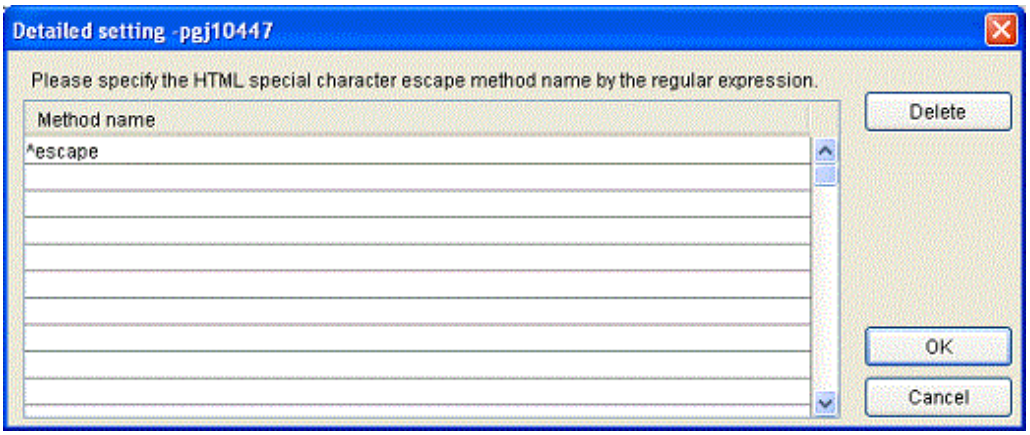
3.4.5.30 Specify the method to escape the HTML special character

The method to escape the HTML special character is set by the regular expression.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification. When the method that starts from escape is defined, it is set as ^escape.

＜Applicable Rules＞

pgj10447



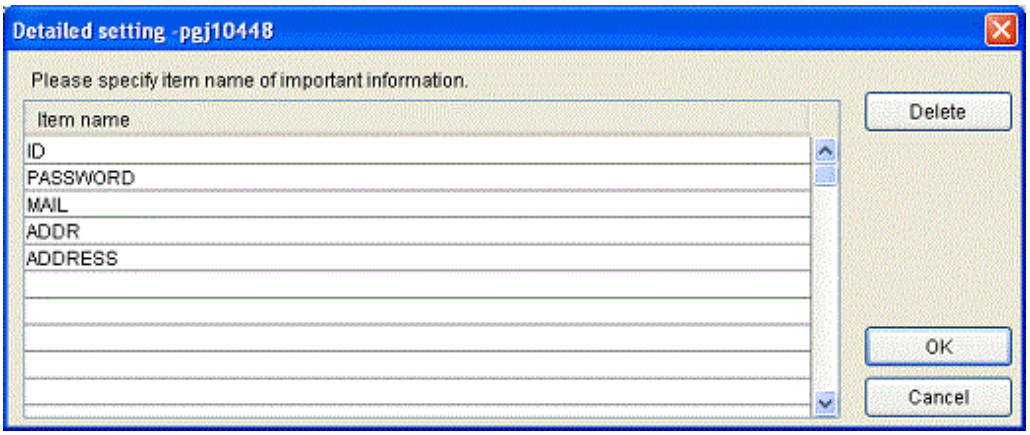
3.4.5.31 Specify item name of important information

The item name to be detected is set.

Click the [Setting] button displayed in the customized column, and the dialog box will be displayed to enable specification. All items become detected when not specifying it.

＜Applicable Rules＞

pgj10448

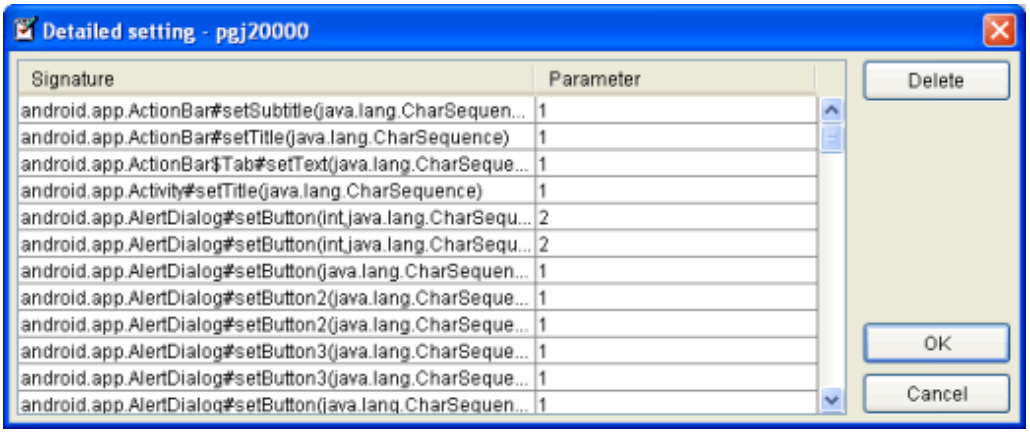


3.4.5.32 [Android] Specify API for Displaying Onscreen Character Strings

Set API that enables character string display on the screen, which is a check target. Click the [Setting] button displayed in the customize column of an applicable rule, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj20000



List of Configuration Items

Item	Description
Signature	Specify the procedure (method or constructor) of Java. - Specify '#' for the separation with class name. - The procedure writes with brackets. - The parameter only enumerates type by separating with ','(comma). - Use "[]" for the variable parameter instead of "…" when you specify the variable argument for the parameter. - Use the real type for the variable parameter instead of the type variable name, when you specify the type variable for the parameter. For example, specify Object type when you specify type variable T of the type parameter <T> for the parameter. - Specify the parameter of the procedure excluding receiver(this) of the instance method. - The constructor name is specified with <init>.

Item	Description
	Example) android.app.ProgressDialog#show(android.content.Context, java.lang.CharSequence, java.lang.CharSequence)
Parameter	When specifying string literal, specify the parameter index as a targeted for check. In case of multiple specifications, enumerate by ',' (comma) separation. If nothing is specified, then all parameters are targeted for check. Example) Detect when either the second or the third parameter is specified as a string literal. [Signature] android.app.ProgressDialog#show(android.content.Context, java.lang.CharSequence, java.lang.CharSequence) [Parameter] 2, 3

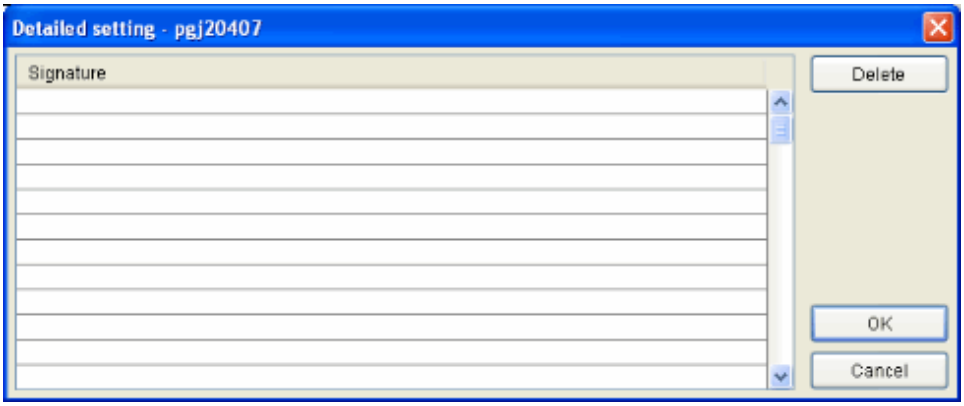
3.4.5.33 [Android] Specify the method that controls the log output call

Specify the method including the call of the Android log output such as android.util.Log#d(String, String) to be excluded from the check target.

[Setting] button displayed in the customize column of an applicable rule, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj20407



List of Configuration Items

Item	Description
Signature	Specify the procedure (method or constructor) of Java. - Specify '#' for the separation with class name. - The procedure writes with brackets. - The parameter only enumerates type by separating with ','(comma). - Use "[]" for the variable parameter instead of "…" when you specify the variable argument for the parameter. - Use the real type for the variable parameter instead of the type variable name, when you specify the type variable for the parameter. For example, specify Object type when you specify type variable T of the type parameter <T> for the parameter. - Specify the parameter of the procedure excluding receiver(this) of the instance method. - The constructor name is specified with <init>.

Item	Description
	Example) com.foo.sample.LogUtil#d(java.lang.String, java.lang.String)

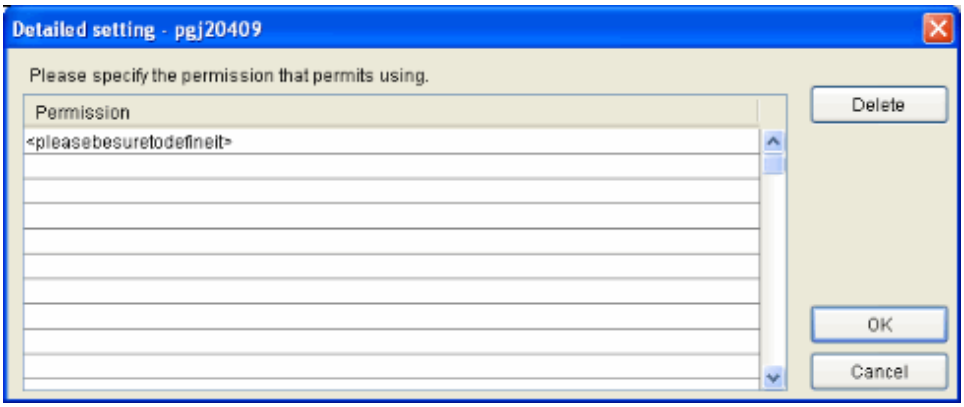
3.4.5.34 [Android] Specify the permission of the application

Set the permission of the Android application that is the check target.

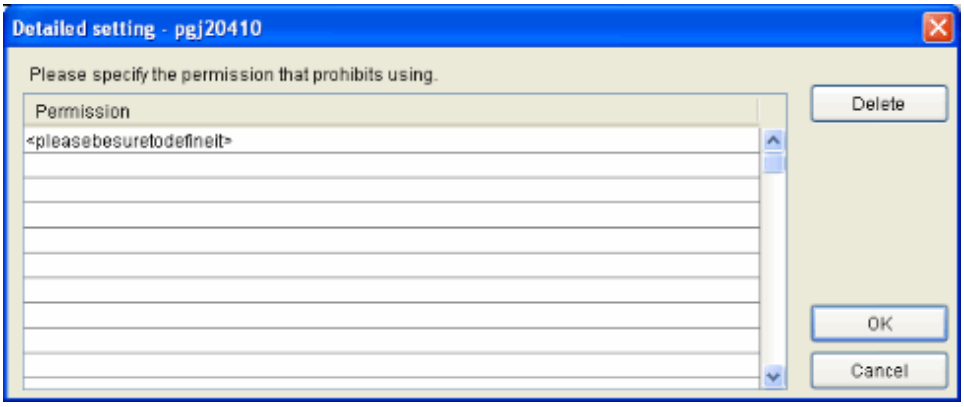
[Setting] button displayed in the customize column of an applicable rule, and the dialog box will be displayed to enable specification.

<Applicable Rules>

pgj20409



pgj20410



List of Configuration Items

Item	Description
Permission	Specify the permission that can be set to < uses-permission > tag of AndroidManifest.xml. Even if the permission not defined by Android is specified, it does not become an error. Please note it. Example) android.permission.INTERNET

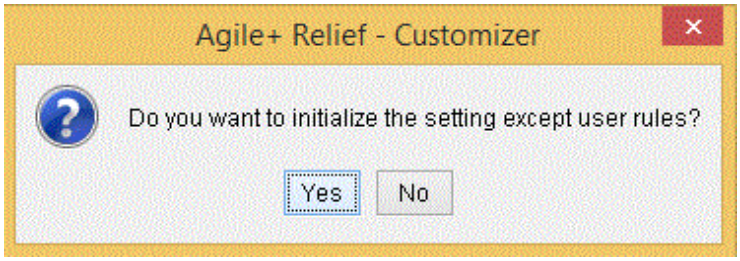
3.4.6 Restore to Initial Setting

Make all the modified content invalid and restore to contents at installation.

<Operations>

1. Select [Apply] > [Restore Defaults].

2. Once the confirmation dialog box is displayed, select the processing.



- [Yes]: Restore to the state of initial setting.
- [No]: Cancel the processing.

3.4.7 Lock / Unlock Applied Rules

An applied rule can be locked in order to prevent any alteration of rule selection or customize information that are checked objects, so that developers cannot freely change check rules that are specific for a particular project.

Select [Apply] > [Rule Lock] to lock all rules with checked [Apply Rule]. Or if rules are already locked, then they can be unlocked. If there is no rule with checked [Apply Rule], then nothing can be locked. An unlocked rule can be edited freely.

If there is a locked rule, the following function restrictions will apply.

- Rules in exclusive relationship

  If either pgj10013 or pgj10014 rule is locked, then the other rule cannot be applied.

- [Apply]>[Batch Apply]>[Apply Importance degree S]

  A locked rule that has an importance degree of "W" will be applicable.

- [Apply]>[Batch Apply]>[Cancel Application]

  If a rule is locked, then its application will not be canceled.

- [Apply]>[Restore Defaults]

  If a rule is locked, this function cannot be used.

- [User Rules]>[Change]

  If the selected rule is locked, this function cannot be used.

- [User Rules]>[Delete]

  If the selected rule is locked, this function cannot be used.

- [User Rules]>[Delete All]

  If a rule is locked, it will not be deleted.

- [User Rules]>[Duplicate]

  A copied rule will not be locked.

- [User Rules]>[Output User Rule Template]

  The state of lock will not be transferred automatically. When reading an output user rule template, the rule will not be locked.

The current state of lock can be confirmed by the following steps.

- [Apply]>[Rule Lock]

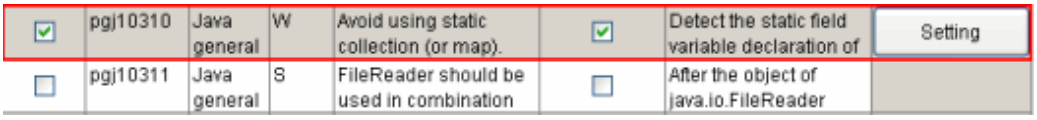
  By the checkmark of [Apply]>[Rule Lock], judge the state of the lock.

  <Unlocked>

  <Locked> A check mark is placed in the menu.

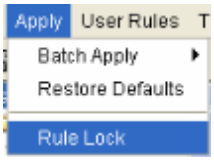
- Rules List

  A locked rule has a gray background display, and all of the controls including checkbox are invalid. However, you can hold down the [Setting] button to confirm the content of setting.



<Operations> Lock a rule

1. Select [Apply] > [Rule Lock].



2. The password input dialog box will be displayed, and enter a password.

   A password consisting of more than 5 characters, using alphanumerical characters and symbols, is entered.

   - [OK]: Lock all of the rules with [Apply Rule] checked.

- [Cancel]: Cancel the processing.

<Operations> Unlock a rule

1. Select [Apply] > [Rule Lock].

2. The password input dialog box will be displayed, and enter a password.

- [OK]: Unlock all of the locked rules.
- [Cancel]: Cancel the processing.

3.5 Create Project-Specific Rules

This section describes how to create project-specific rules.

3.5.1 Overview

The rules that are not offered by standard in Agile* Relief J can be created as the project-specific rules. Such rules are called user rules.

The rules of naming rule check, existence check, description order check and number check etc. for Java elements of any class or method etc. can be created by user rules.

This section describes how to achieve the check of user rules.

3.5.2 Create New User Rule

The user rule that contains the elements of "Basic information", "Rule definition", and "rule violation judgment" is used for setting or modifying each item.

3.5.2.1 Create User Rule

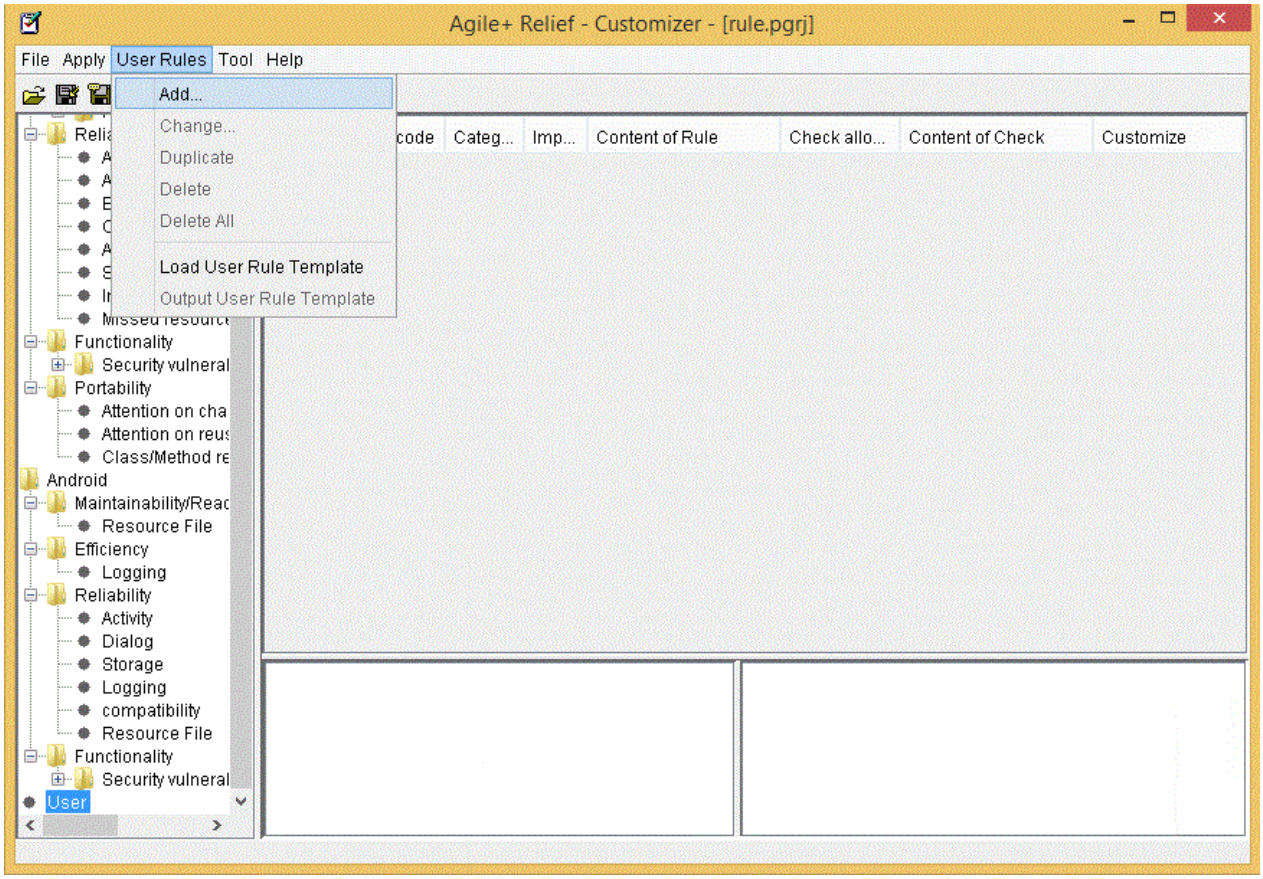
Add a new user rule.

Once the [User Rule Configuration] dialog is displayed, set the required items and create a user rule.

<Operations>

1. Select a User from the rule category tree.

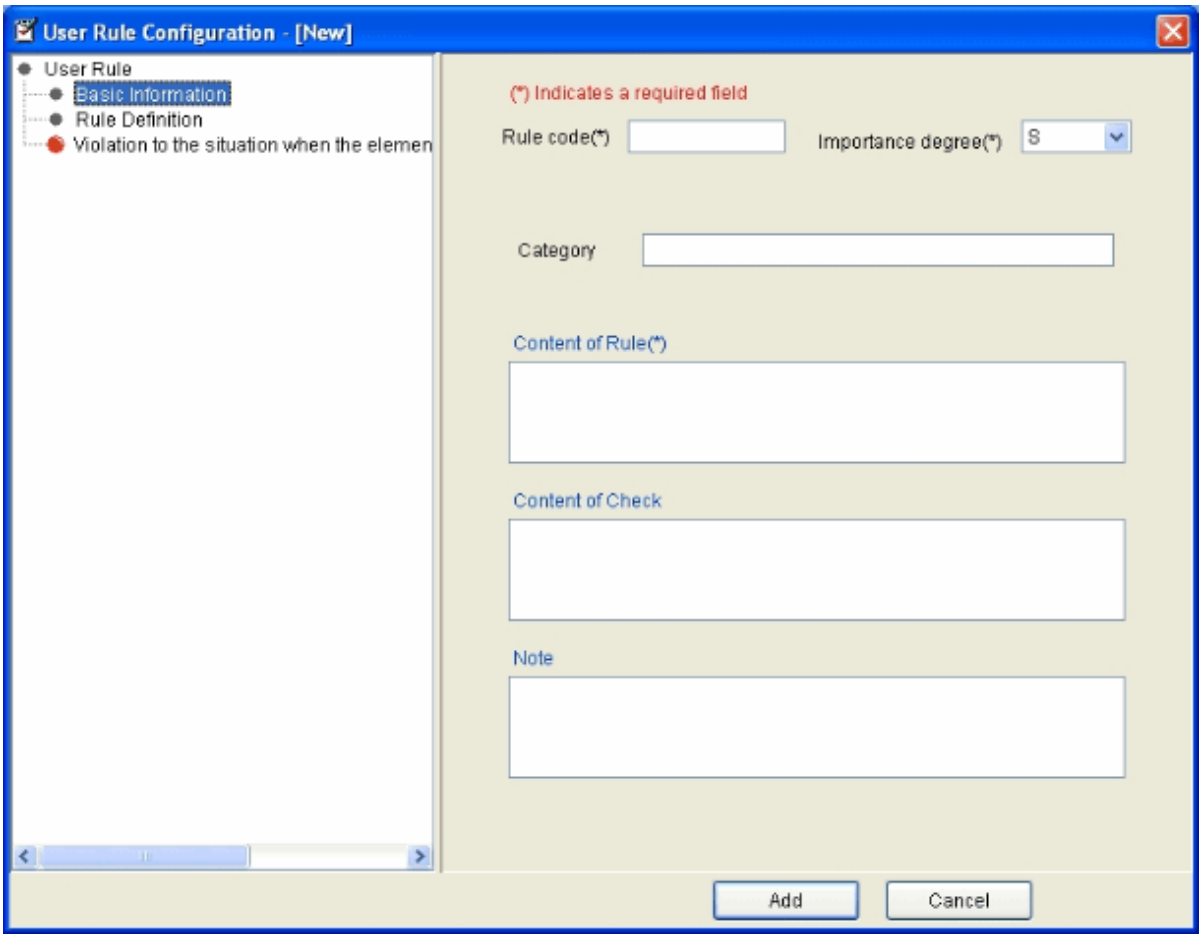
2. Select [User Rules] > [Add], or select [Add] from the pop-up menu displayed by right-click.



3. Set the required items in the [User Rule Configuration] dialog and create a user rule.

3.5.2.2 Set Basic Information

The view of basic information setting is displayed.



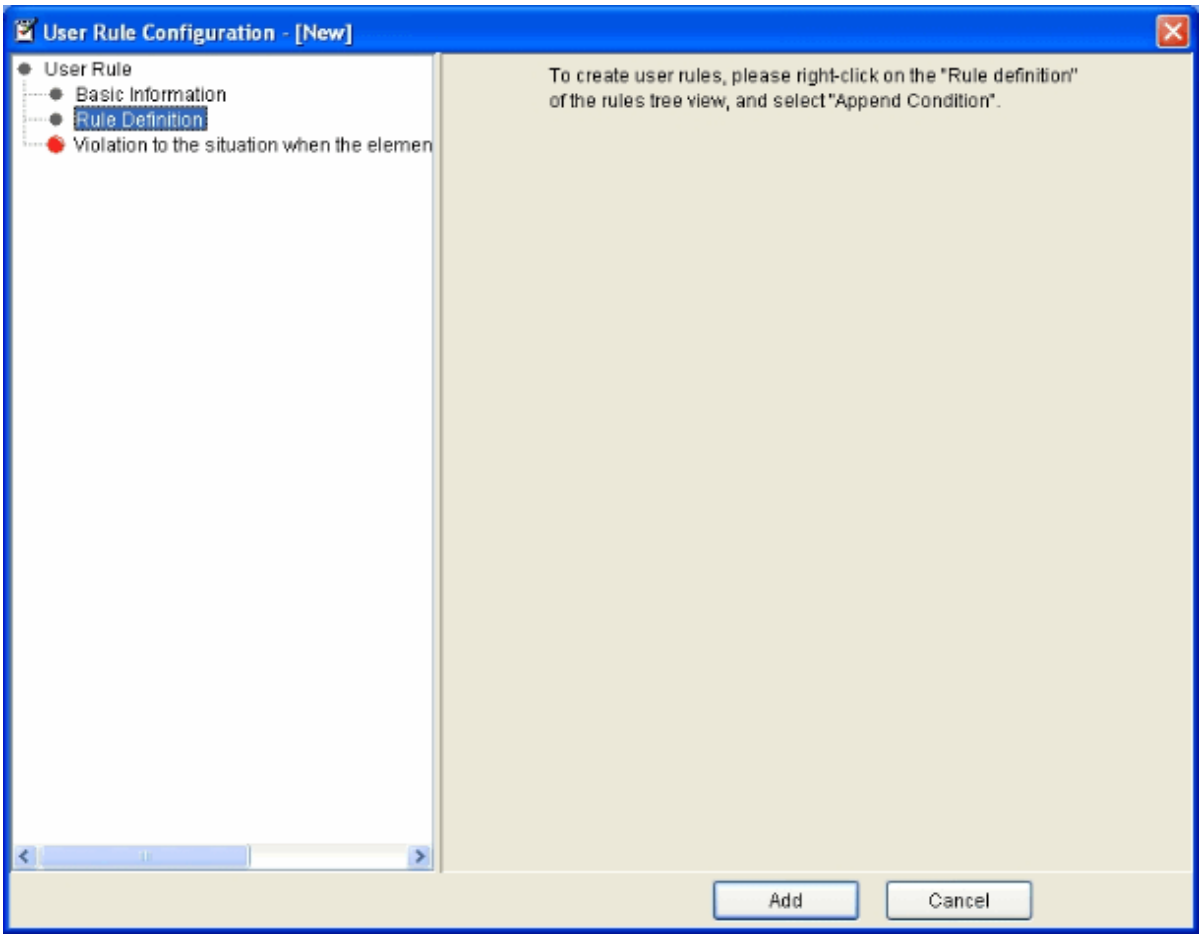
List of Configuration Items

Configuration Items	Item Description
Rule code (*)	Input the rule code. The single-byte alphanumeric characters, '_', and '-' are allowed.
Importance degree (*)	Select the importance degree. The types of importance degree are S and W.
Category	Input the category of rule.
Content of Rule(*)	Input the definition content of rule.
Content of Check	Input the processing contents of the checked rule.
Note	Input the notes.

The items marked with (*) are mandatory.

3.5.2.3 Set Rule Definition

In the state when rule definition is selected, the guidance view on the right of [User Rule Configuration] dialog is displayed.

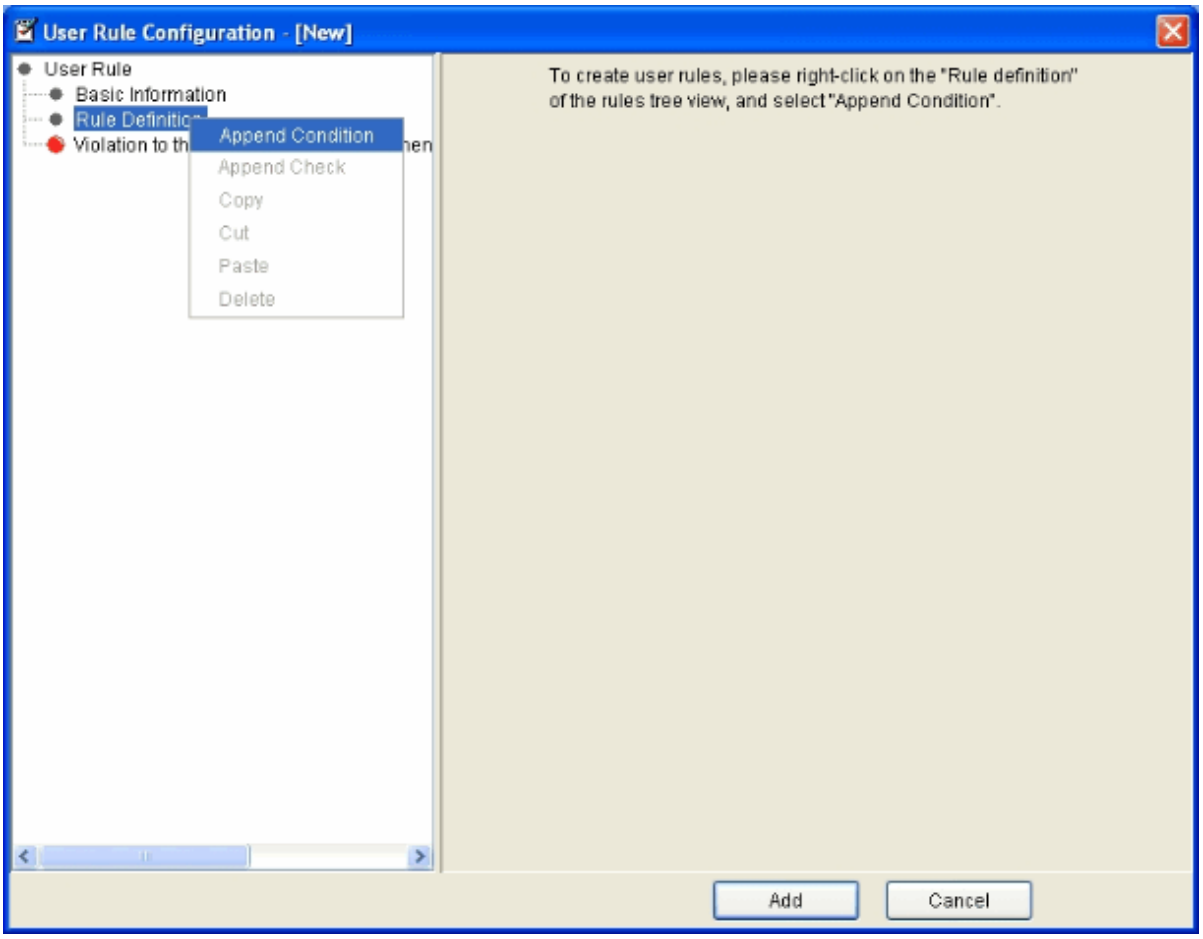


3.5.2.3.1 Append Condition

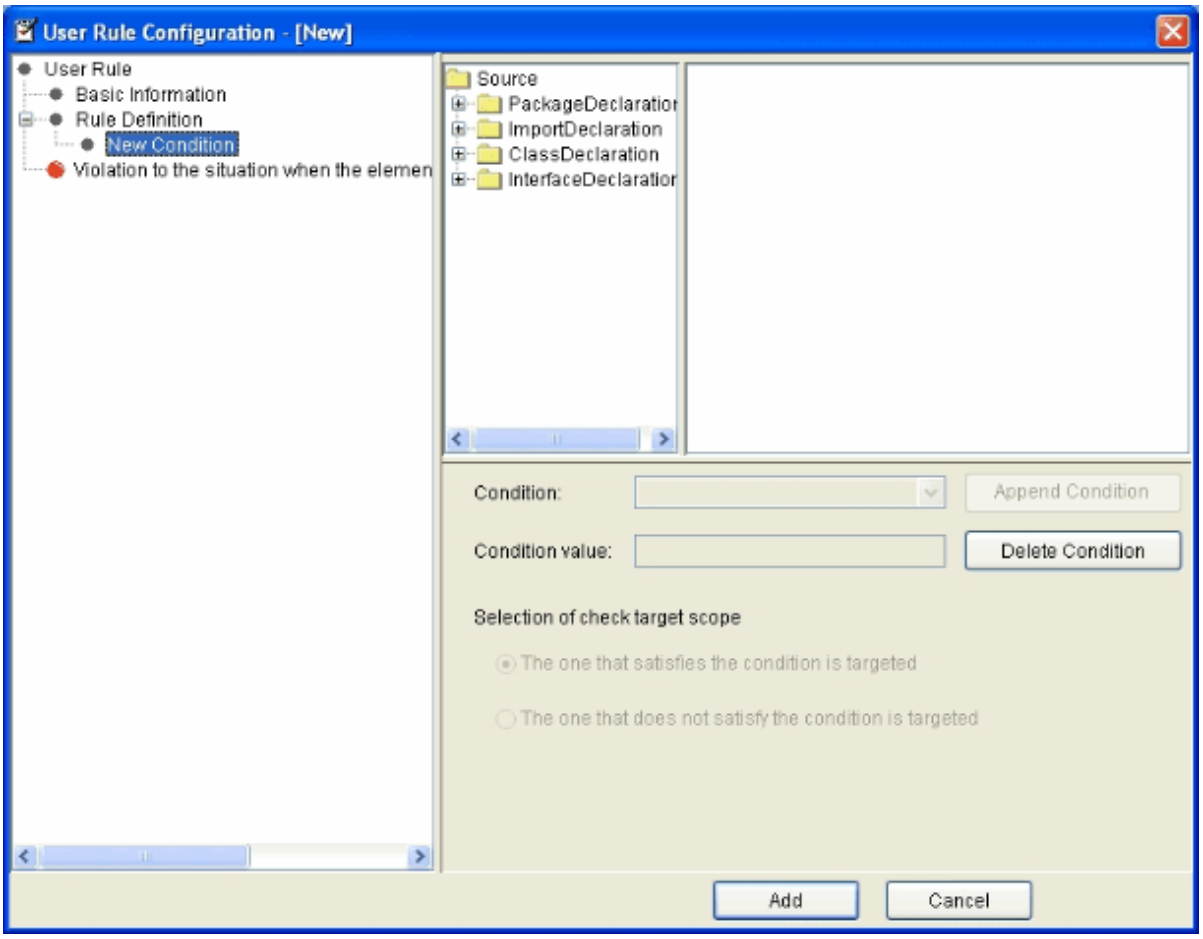
Append the conditions of rule definition.

For details about the setting of condition part, please refer to "3.5.2.4 Set Condition Part of Rule Definition".

In the state when the rule definition or condition elements are selected, right-click to display a pop-up menu.



Select [Append condition] from the pop-up menu displayed by right-click. to add "New condition" under the selected element of rule tree view, and condition element setting view is displayed on the right of [User Rule Configuration] dialog.



Set the condition element in the condition element setting view. (Condition element setting)

Append check elements after appending condition elements. (Append check)

3.5.2.3.2 Append Check

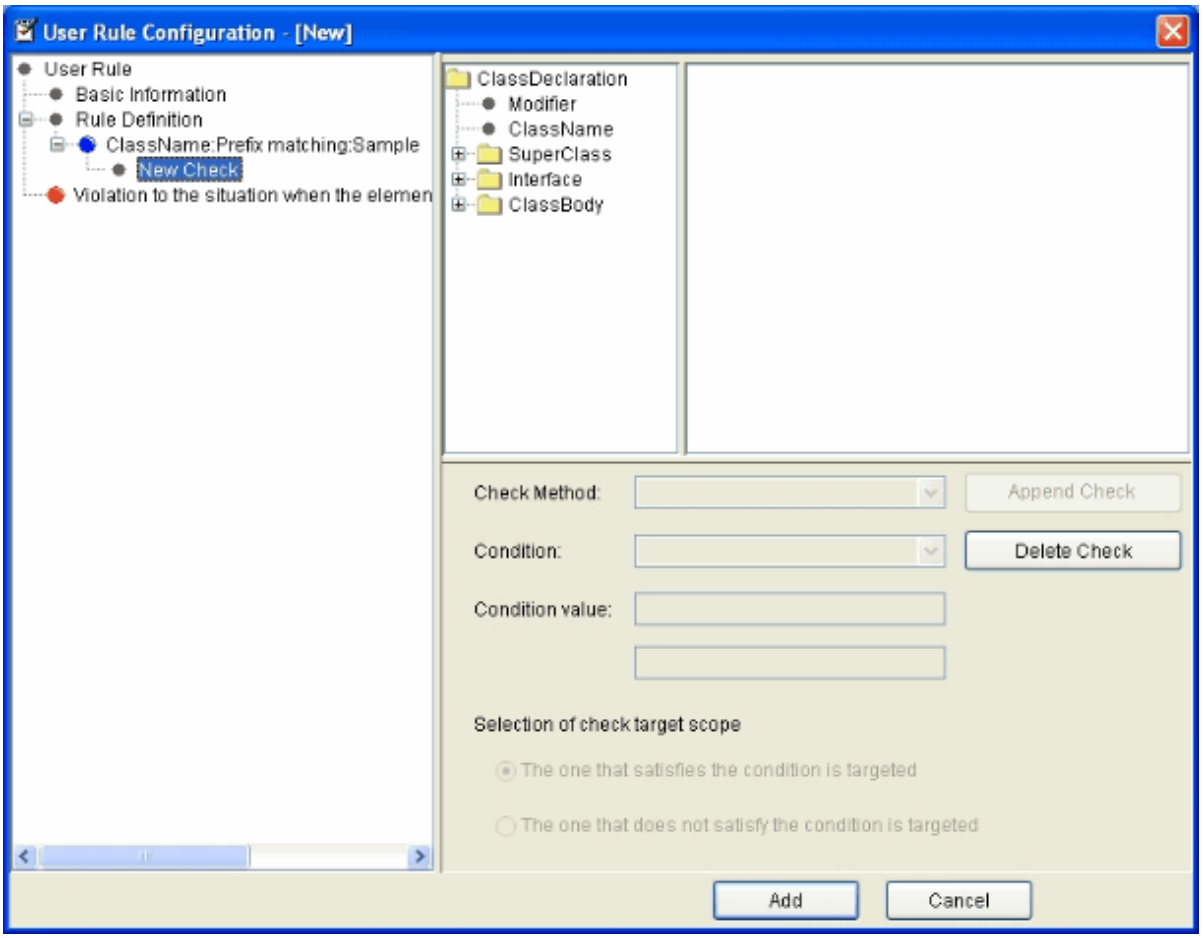
Append check elements in condition.

For the setting of check, please refer to "3.5.2.5 Set the Check Sequence of Rule Definition".

Select a condition and right-click, once the pop-up menu is displayed, and select append check.

Add [New Check] is added under the rule definition of rule tree view.

In addition, the check element configuration view is displayed on the right of [User Rule Configuration] dialog.



(After "Append check")

The [Delete Check Result] button is enabled in the check element configuration view. When [Delete Check Result] is selected, the added "New check" will be deleted.

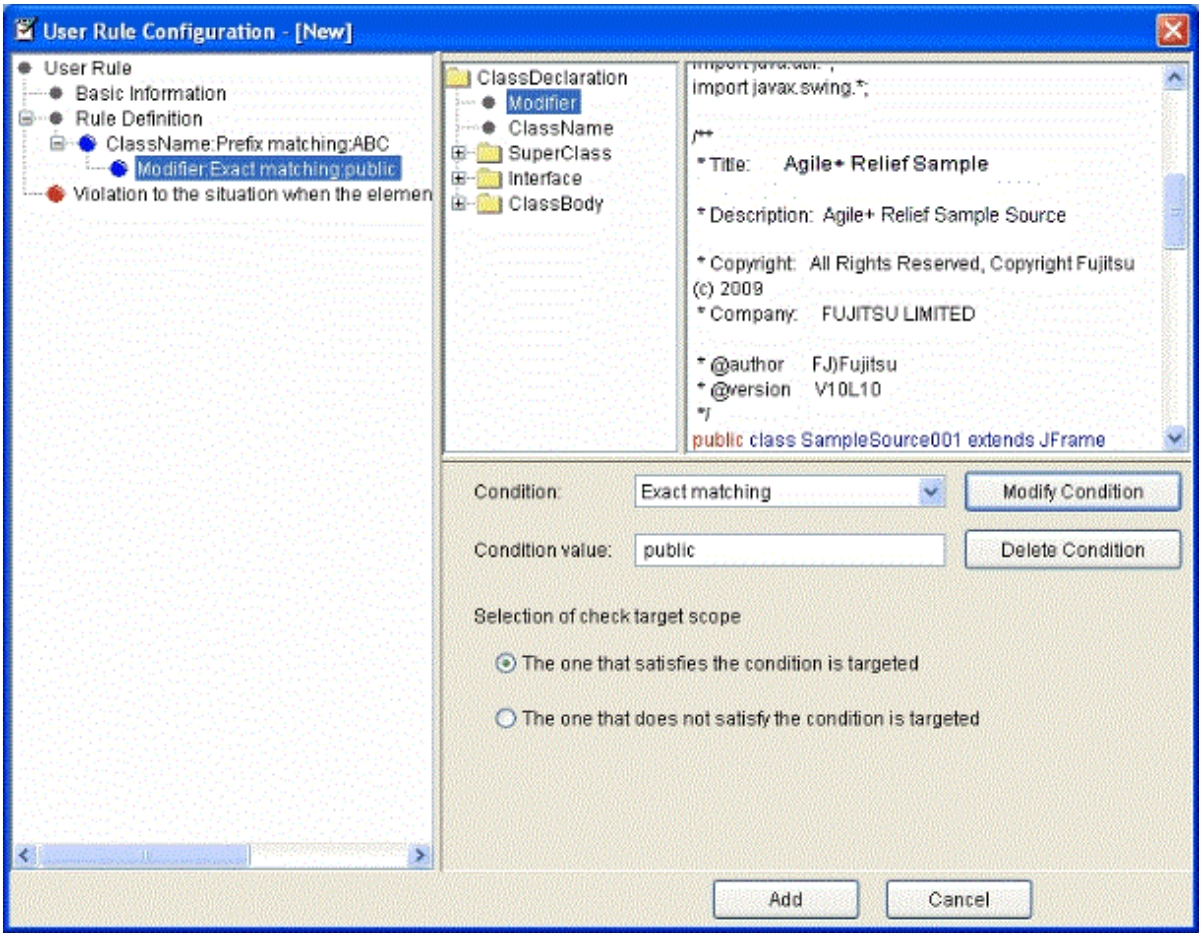
Set the check elements in the check element configuration view. (Check Element Configuration)

3.5.2.3.3 Combine Conditions with Check

The conditions in the rule definition can be combined with the check through the AND condition or OR condition.

- AND condition

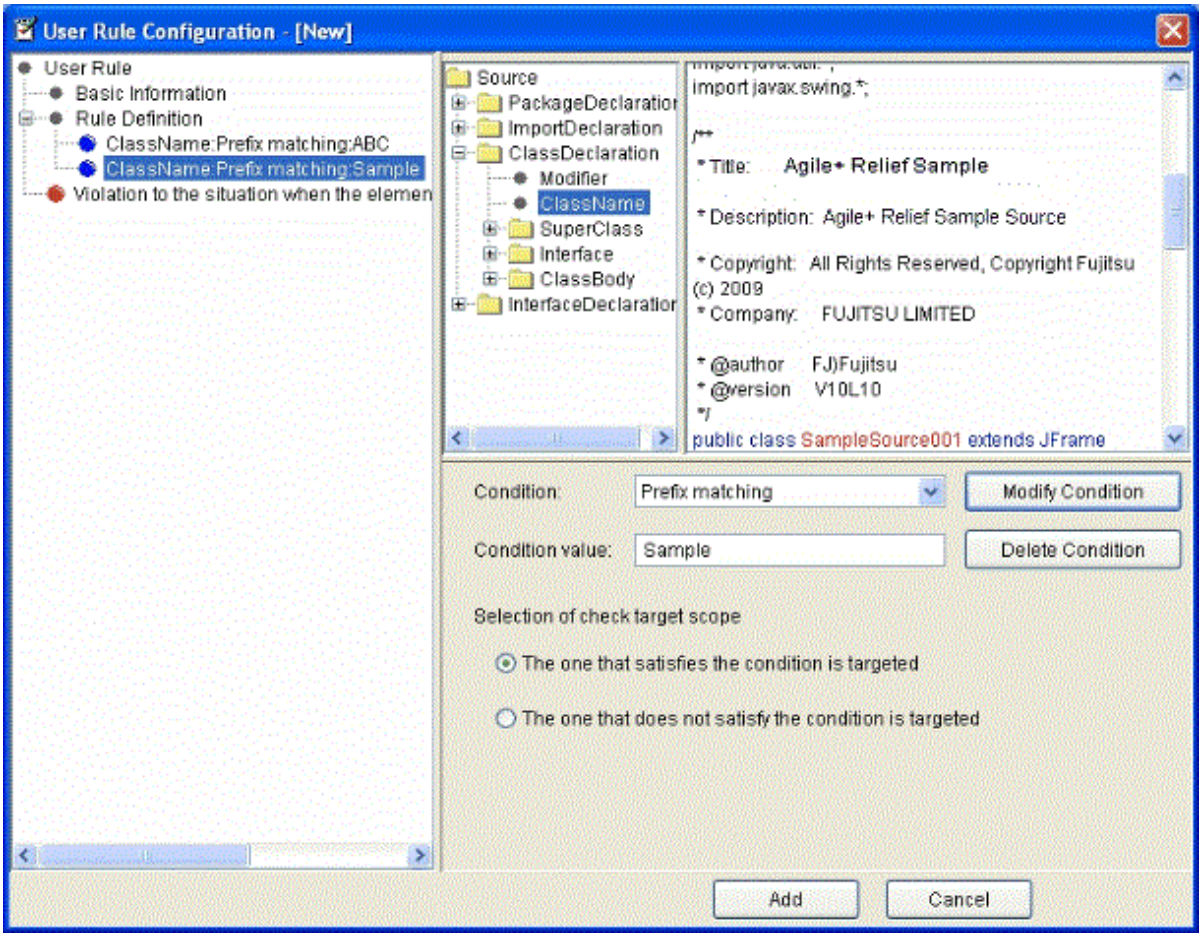
When the condition part has been set under the condition part, the AND condition will be used.



In the above settings, the initial of class name begins with ABC, and the qualifier specifies the public class as the target range for check.

- OR condition

  When multiple conditions are set at the same level, the OR condition will be used.



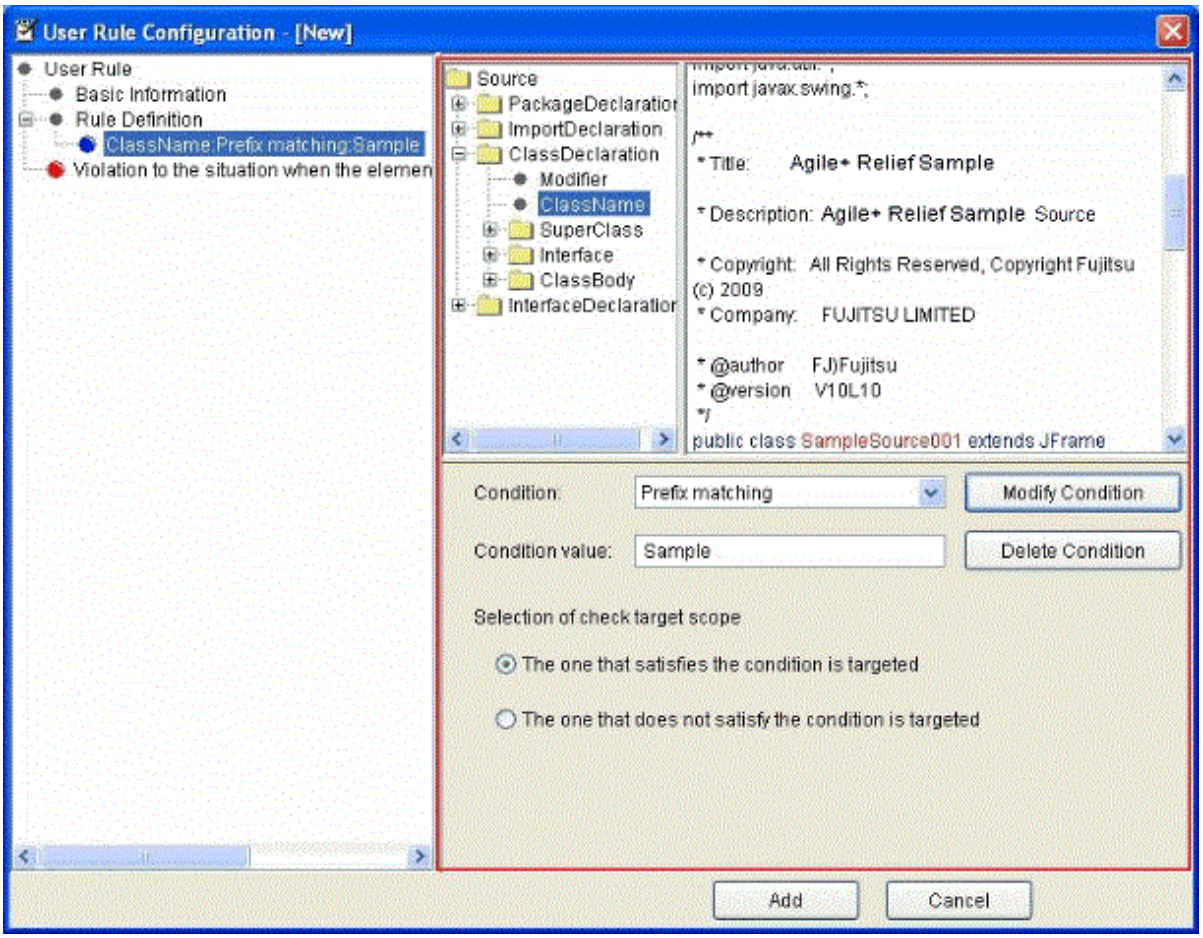
  In the above settings, the initial of class name begins with ABC, or the class whose name starts with Sample is specified as target range for check.

3.5.2.4 Set Condition Part of Rule Definition

Specify the check target scope in the Java elements for the Java file of check target.

In addition, limit the check target scope through condition and condition value.

Set the condition elements in the condition element setting view.



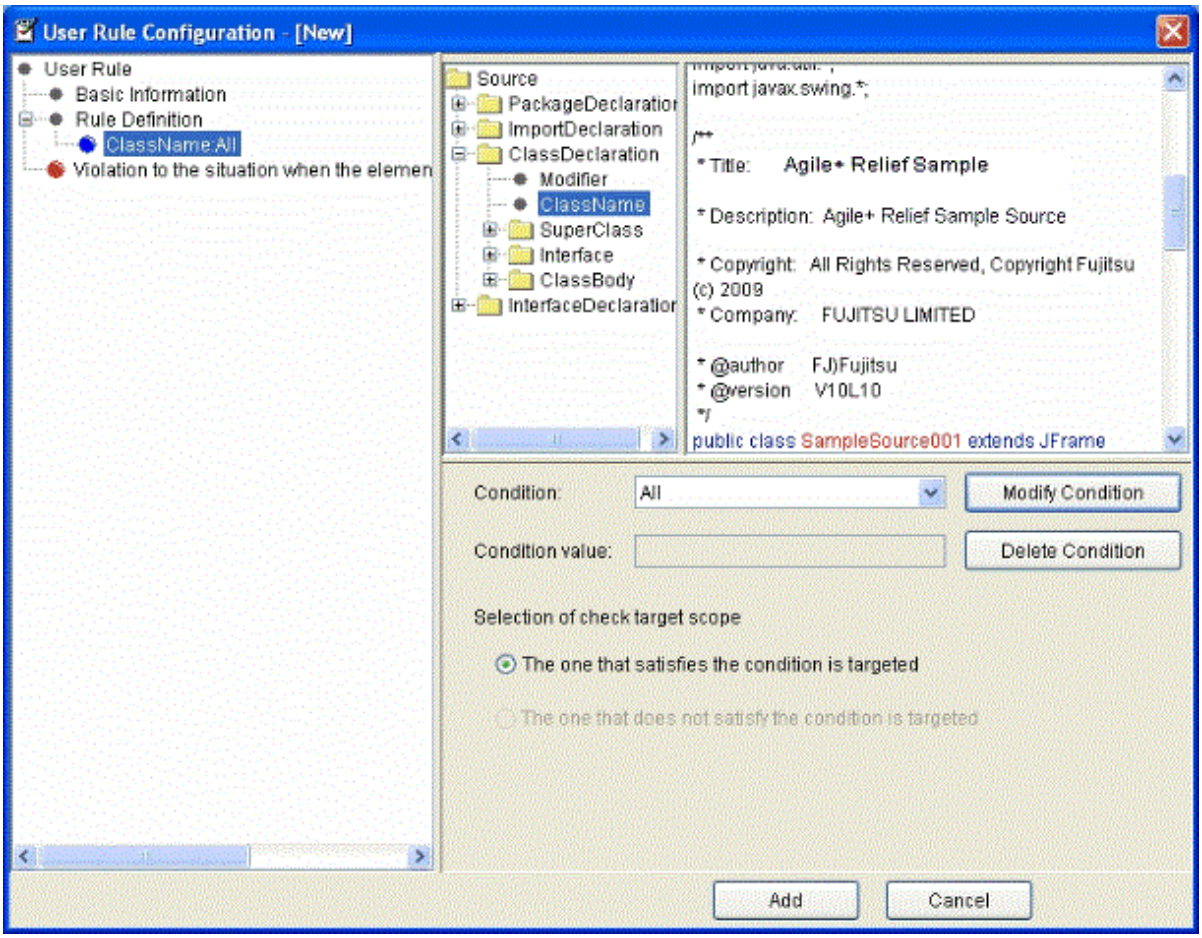
<Operations>

1. Select the element from the Java element tree that shows the syntax element of Java.
2. Specify the condition for the element selected in Java element tree ("All", or conformance condition of name) in "Condition".

   In case of the rule that checks all Java files, select "Source" from the Java element tree and specify [All] in "Condition".
3. Specify [Check target scope].
   - When Java element tree and the scope specified in condition are set as the check target, select [Targeted at the one that satisfies the condition].
   - When Java element tree and the elements outside the scope specified in the condition are set as the checked target, select [Targeted at the one that does not satisfy the condition].
   - When "All" is selected in "Condition", [Targeted at the one that satisfies the condition] cannot be selected.

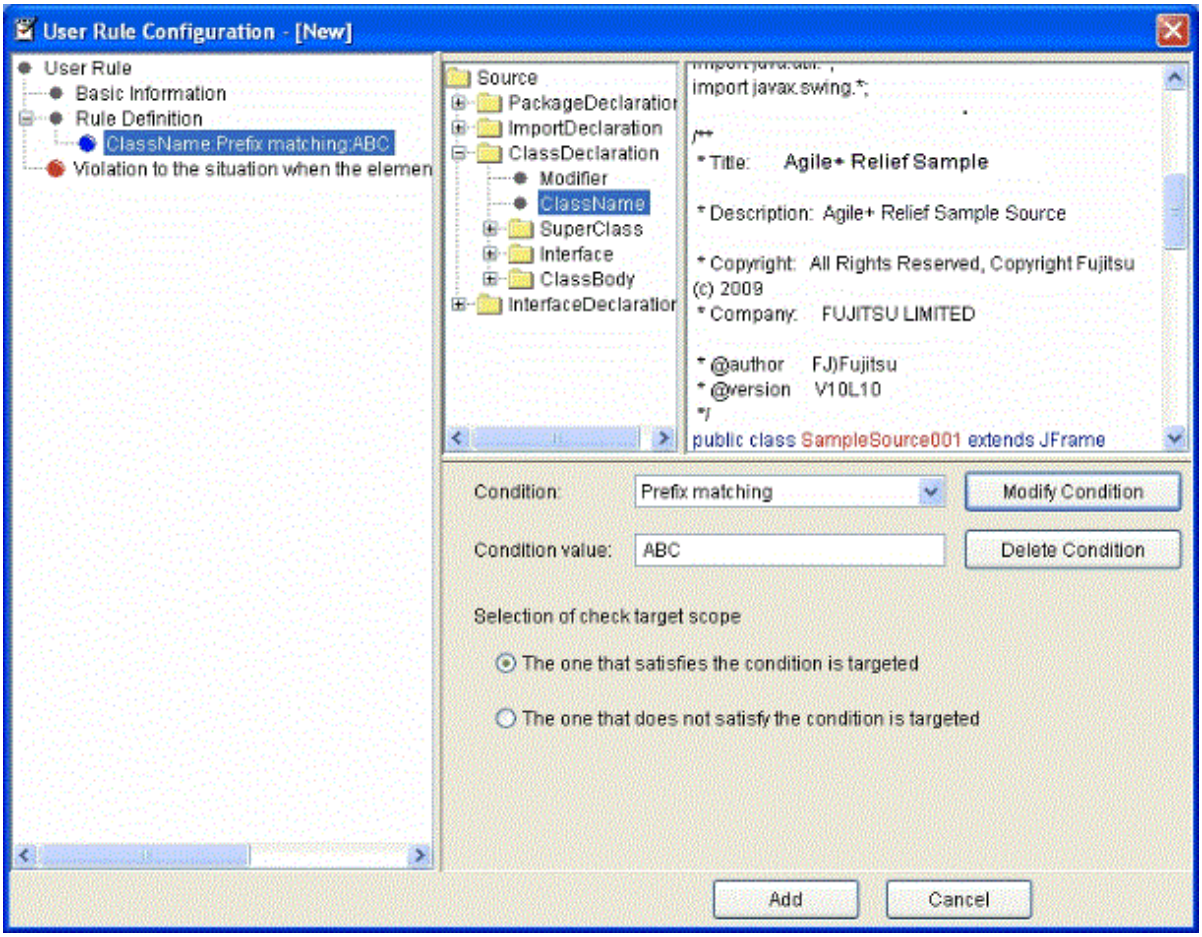
Example

- Select the class name of Java element and set Condition to "All"



All the classes in the Java file to be checked are set as the checked target scope.

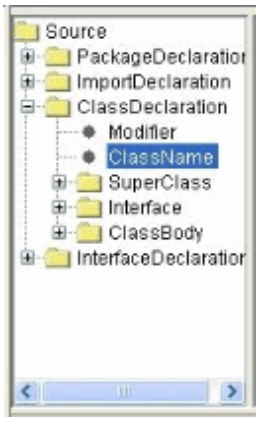
- Select the class name of Java element, and set Condition: "Prefix matching" and Condition value: "ABC".



For the class existing in the Java file to be checked, the classes with "ABC" being attached to its front will become the check target scope.

3.5.2.4.1 Java Element Tree

Select the limited Java elements (scope of check target) from the Java element tree.



is the branch element. Java elements exist under it.

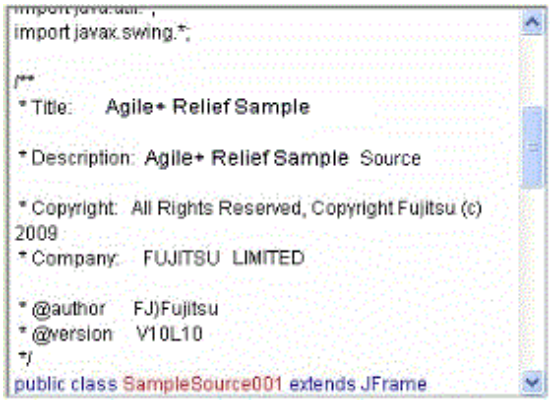
is a terminal element.

*After a Java element is selected, the Java element corresponding to Sample Source will be displayed in colors.

*After a Java element is selected, the Condition that allows specification can be selected.

3.5.2.4.2 Sample source

The Java elements selected from the Java element tree will be displayed as the sample source for confirming the location of Java source.



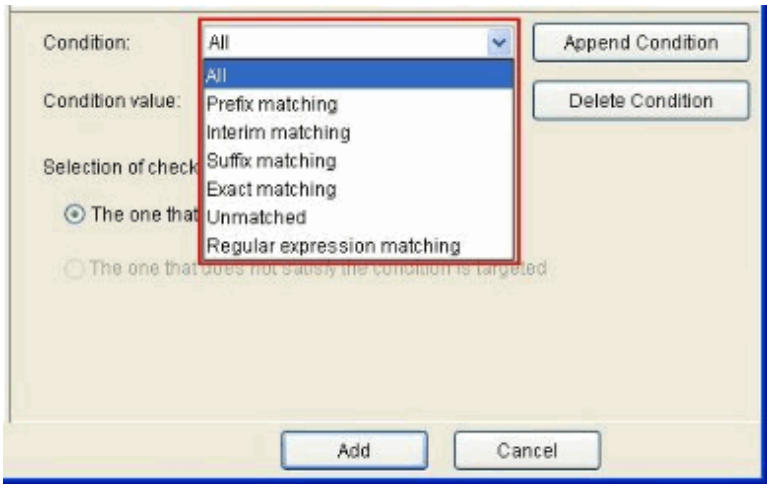
When the Java elements selected from the Java element tree are the branch elements, these Java elements are displayed in blue.

When the Java elements selected from the Java element tree are the terminal elements, the upper-level Java elements are displayed in blue, and the selected Java elements are displayed in red.

3.5.2.4.3 Condition

Set the restriction conditions for the Java elements (check target scope) selected from the Java element tree.

Select a condition from the condition selection combo box.



The following table describes the list of conditions.

Condition	Description
All	All elements are set as the check target scope.
Prefix matching	Limit the check target scope by prefix.
Interim matching	Limit the check target scope by interim.
Suffix matching	Limit the check target scope by suffix.

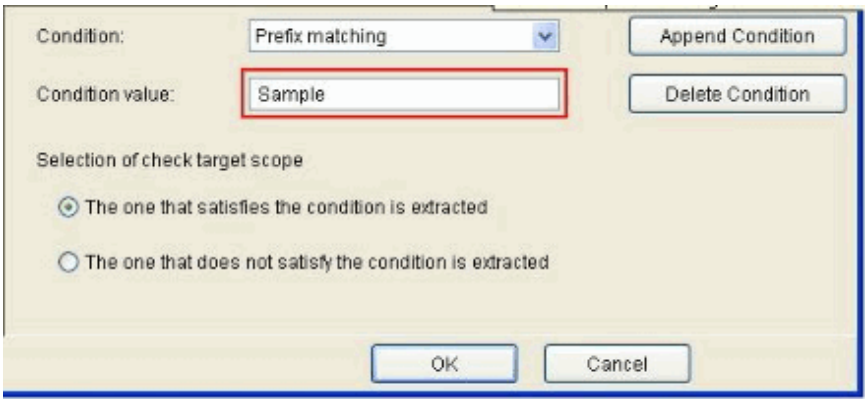
Condition	Description
Exact matching	Limit the check target scope by exact matching.
Unmatched	Limit the check target scope by unmatched.
Regular expression matching	Limit the check target scope by regular expression.

*When the Java elements selected from the Java element tree are the branch elements, only "All" can be selected.

*In case of inner class declaration and inner interface declaration of Java elements, only "All" can be selected.

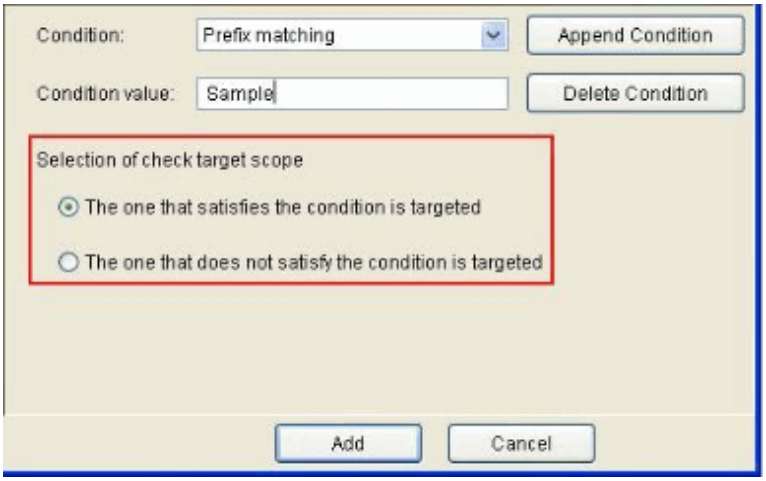
3.5.2.4.4 Condition Value

Input the condition value corresponding to the condition.



*When the Java elements selected from the Java element tree are the branch elements, it is unable to input.

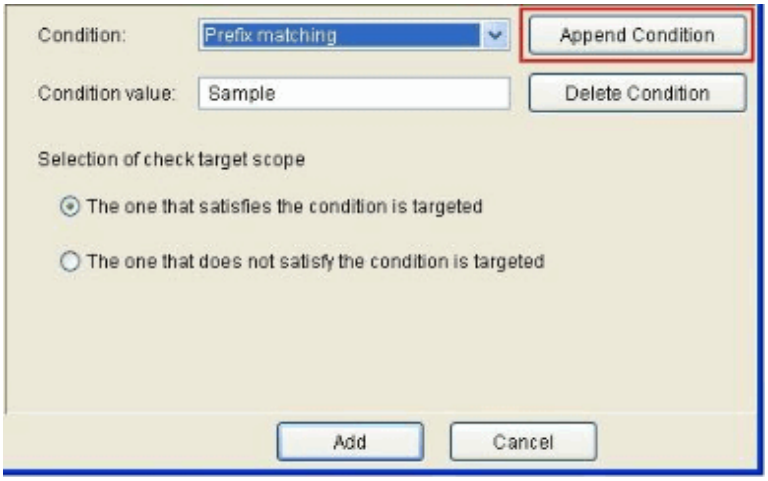
3.5.2.4.5 Selection of check target scope



The following table describes the list of options.

Item	Description
Targeted at the one that satisfies the condition	Set the one that satisfies the condition set in the condition configuration view as the comparison object.
Targeted at Targeted at the one that does not satisfy the condition	Set the one that does not satisfy the condition set in the condition setting view as the comparison object.

3.5.2.4.6 Append Condition

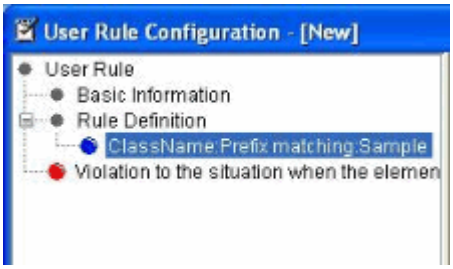


Click the [Append Condition] button in the condition configuration view to append a condition.

* After a "New condition" is appended, when the condition has been set in condition configuration view, it is necessary to click the [Append Condition] button to add a condition.

After the [Append Condition] button is selected, append a check.

After the condition is appended, "New condition" in the rule tree view will be changed to set condition.



(Display of condition in the rule tree view)

ClassName:Prefix matching:Sample

From the left hand side, it turns into "Java element selected in the selection Java element tree of check target scope: Condition: Condition value".

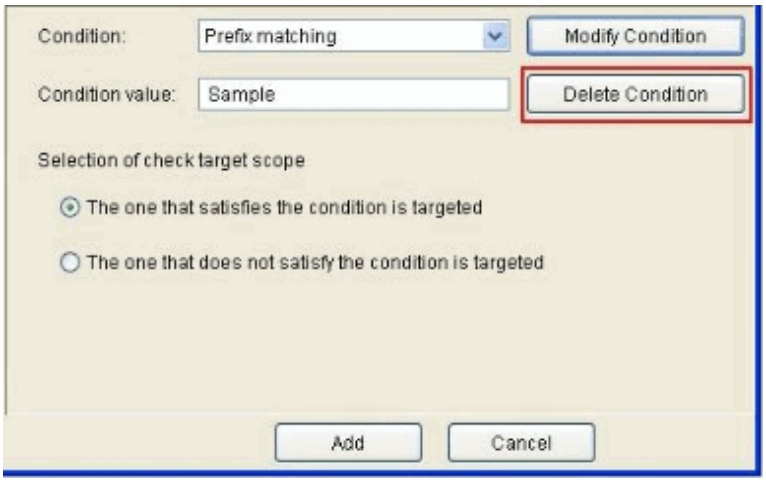
* is displayed when the selection of check target scope is set to "Targeted at the one that satisfies the condition"; is displayed when it is set to "Targeted at the one that does not satisfy the condition".

*When condition is set to "All", ": Condition value" does not exist.

In addition, the [Append Condition] button is changed to [Modify Condition] button.

*To modify the setting of condition, please make sure to select the [Modify Condition] button.

3.5.2.4.7 Delete Condition



To delete a condition, select the [Delete Condition] button.

*Other conditions or checked elements existing under the selected condition will also be deleted.

3.5.2.5 Set Check Part of Rule Definition

In the check part, define the check method for the Java elements that are limited by the condition part.

The check method is defined using the check content and condition, as well as condition value.

<Operations>

1. Select elements from the Java element tree that shows Java syntax element.
2. In "Content of check", the patterns that can be checked are different from the specified check target for the elements selected from the Java element tree.

List of check contents

Check Content	Description
Existence check	Check whether the Java elements selected from the Java element tree exist. Example of rule that can be created) - The constructor declaration must exist in the specified class - For the initial of class name that begins with "ABC", the initial of method name should defined to begin with "Init" - The local variable cannot be defined in the class whose initial of name begins with "ABC"
Uniqueness check	Check whether the Java element selected from the Java element tree exists. Example of rule that can be created) - One constructor defined in the class exists - There is at most one class and interface defined in the source
Order check (beginning) Order check (later) Order check (last)	The order check checks whether the Java elements selected from the Java element tree are described in specified order within the check scope. * Check whether it conforms to the check content specified in "Order check (beginning)" check element and determine whether to judge according to the specified order of description when the contents specified in the check element "Order check (later)" or "Order check (last)" exist.

Check Content	Description
	Example of rule that can be created) - When the invocation of method with prefix "get" exists, invoke the method with prefix "set" => It is achieved through the combination of Order check (beginning) and Order check (later) - After the method with prefixed "init" has been invoked, invoke the method with prefix "open" immediately => It is achieved through the combination of Order check (beginning) and Order check (last)
Naming check	Check whether the Java elements selected from the Java element tree conform to the specified strings. Example of rule that can be created) - The prefix "m_" should be attached to the name of field variable - When the field variable type is String, the prefix "str" must be attached - The suffix "Class" must be attached to the end of class name
Number check	Check whether the number of Java elements selected from the Java element tree is within the specified scope. Example of rule that can be created) - At most 10 field variables can be defined. - When the prefix of method name is "Init", more than one parameters exist - When the prefix of method name is "Init", no parameter exists
Pre-comment existence check	Check whether the comments are described before the Java elements selected from the Java element tree. Example of rule that can be created) - Describe comments before the try statement - Describe comments before the switch statement
Post-comment existence check	Check whether the comments are described after the Java elements selected from the Java element tree. Example of rule that can be created) - Describe comments after the field variable - Describe comments after the local variable
Pre-comment content check	Check whether the comments are described before the Java elements selected from the Java element tree and whether the specified strings exist. Example of rule that can be created) - Describe the class comment and describe "FUJITSU" in the comment - Describe comments before method declaration in Javadoc format (/**)
Post-comment content check	Check whether the comments are described after the Java elements selected from the Java element tree and whether the specified strings exist. Example of rule that can be created) - Describe comments after the return statement and describe "return" in the comment

3.5.2.5.1 "Extraction condition specification" option

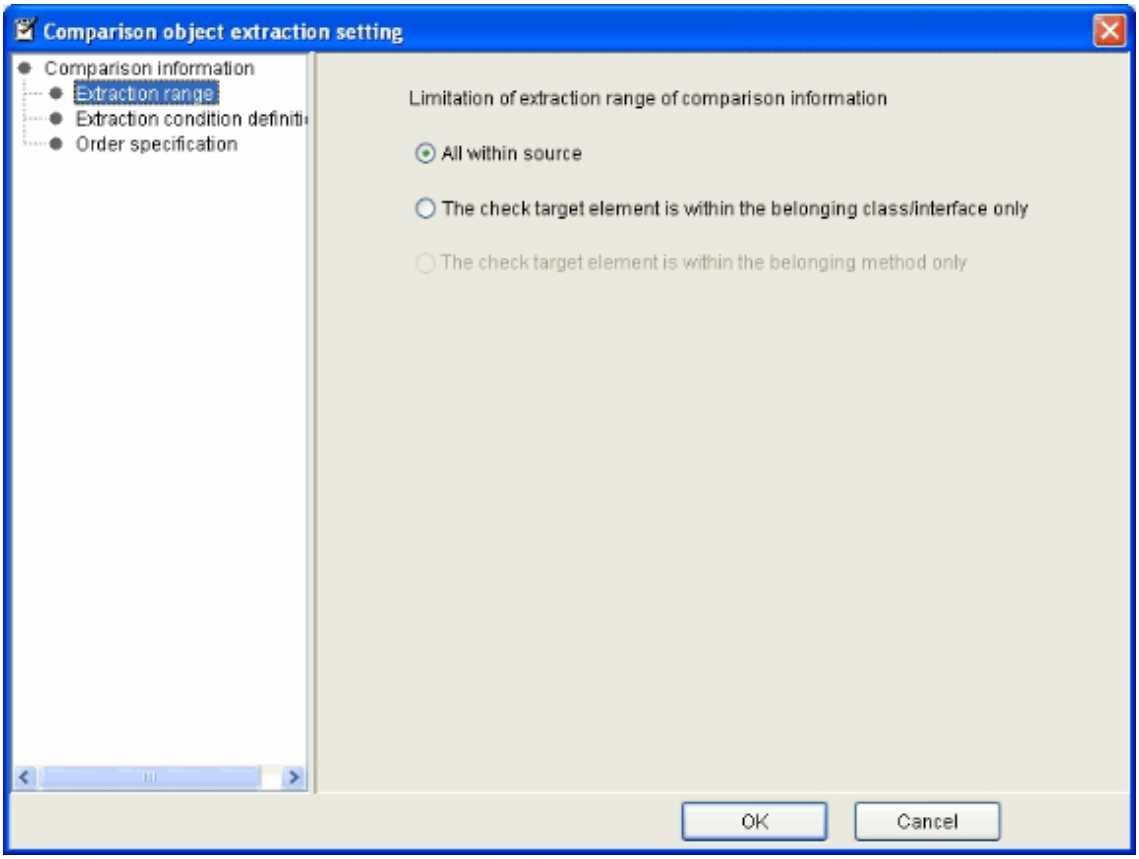
This option is used to compare the Java elements specified in the check part with other information (information extracted according to the extraction conditions) in the source.

When "Existence check", "Uniqueness check" and "Naming check" has been selected from check content, "Extraction condition conformance" can be selection in condition.

When "Extraction condition conformance" has been selected, "Extraction condition specification" option can be selected.

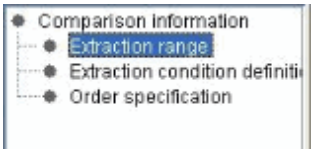
(Rule example) Do not invoke the public method defined in its own class (the class to which method belongs) in the method.

Similar to the rule example, "Extraction condition specification option" may have multiple comparison candidates (public method of rule example) of Java element and becomes effective when it is difficult to set the check in fixed string.



The tree view of extraction information and the setting views are displayed in the comparison object extraction setting dialog.

- **Tree view of extraction information**



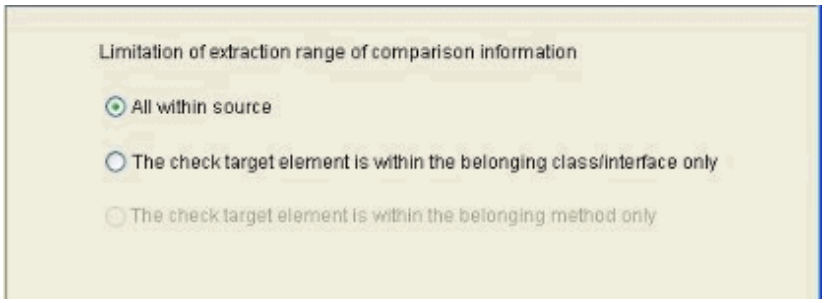
  - Extraction range

- Extraction condition definition
  - Append condition
  - Set extraction elements
- Order specification

Extraction range

Extraction range specification view

Specify the extraction range for the extraction information in Extraction range specification view.

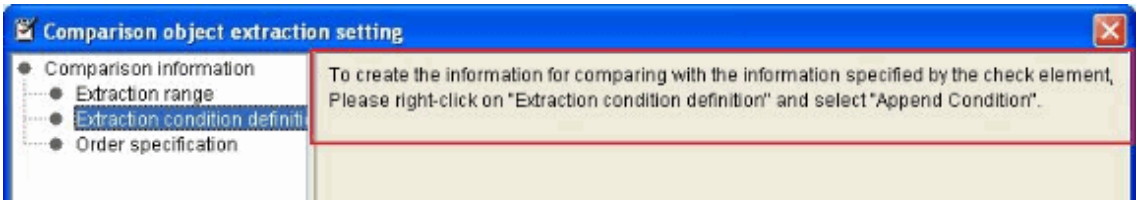


The following table describes the list of options and meanings.

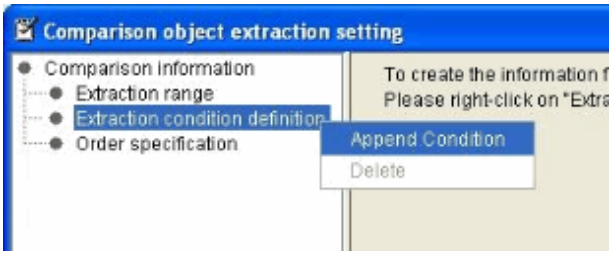
Option Name	Meaning
All within source	Select when the check elements are compared with all the extraction information defined in the extraction condition definition.
The check target element is only within the belonging class/interface	Select when the check elements are compared with the extraction information defined in the extraction condition definition of the belonging class/interface.
The check target element is only within the belonging method	Select when the check elements are compared with the extraction information defined in the extraction condition definition of the belonging method.

Extraction condition definition - Append condition

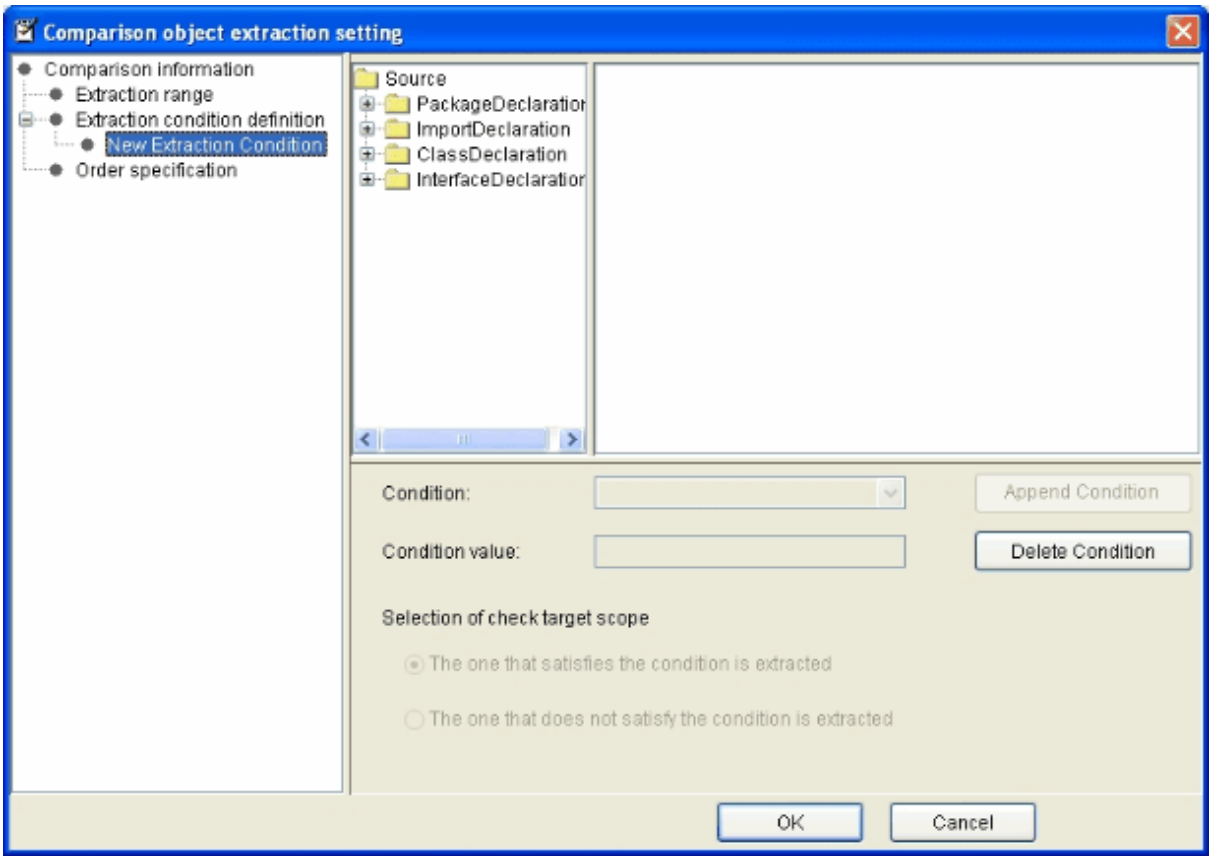
When "Extraction condition definition" of the extraction information tree view has been selected, the guidance view of creating extraction information is displayed.



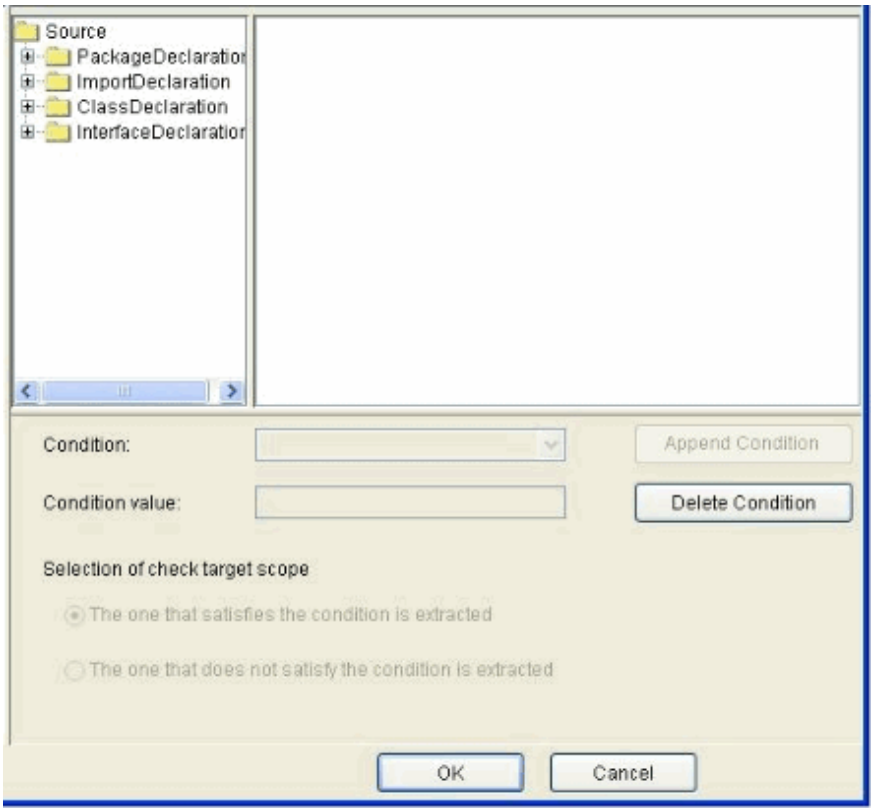
Right-click "Extraction condition definition" of the extraction information tree view, and select to append a condition.



New extraction condition is added in the extraction information tree view, the "Extraction element setting view" is displayed on the right of dialog.



Extraction condition definition - Extraction element setting



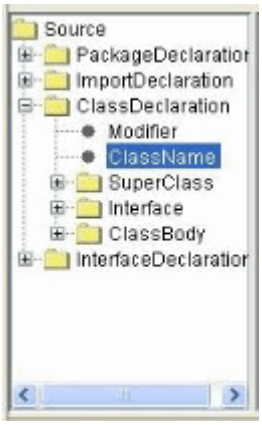
Set the extraction element in the extraction element setting view.

(Operations)

- Java element tree
- Sample source
- Condition
- Condition value
- Select check target scope
- Append condition
- Delete condition

Java element Tree

Select the limited elements (scope of check target) from the Java element tree.



is branch element, under which the Java elements exist.

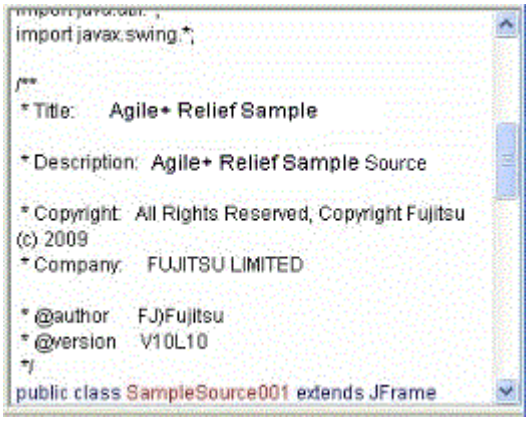
● is terminal element.

* After the Java element is selected, the Java element corresponding to Sample sourcee is displayed in colors.

*After the Java element is selected, the Condition that can be specified can be selected.

Sample source

The Java element selected from the Java element tree will be displayed as the sample source that confirms the location of Java source.



If the Java element selected from the Java element tree is branch element, the selected Java elements will be displayed in blue.

If the Java element selected from the Java element tree is terminal element, the upper-level Java elements will be displayed in blue, and the selected Java elements will be displayed in red.

Condition

Set the extraction conditions for the Java elements (scope of check target) selected from the Java element tree.

Select a condition from the condition selection combo box.

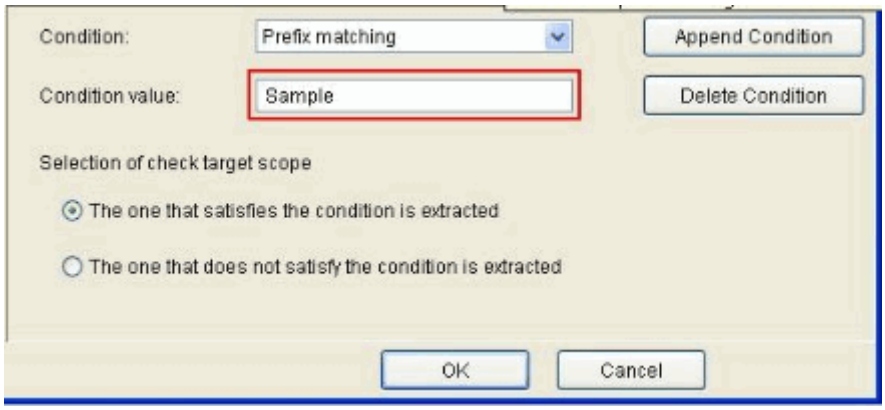
The following table shows the list of conditions and the meanings of each condition.

Condition	Description
All	All the Java elements are set as the comparison target.
Prefix matching	Limit the Java element with prefix matching as the comparison target.
Interim matching	Limit the Java element with interim matching as the comparison target.
Suffix matching	Limit the Java element with suffix matching as the comparison target through the condition value.
Exact matching	Limit the unmatched Java element as the comparison target.
Unmatched	The different Java elements are set as the comparison object.
Regular expression matching	Targeted at Java elements selected from the Java element tree, search using the condition value according to the regular expression and limit it as the comparison target.

*If the Java elements selected from the Java element tree are branch elements, only "All" can be selected.

Condition Value

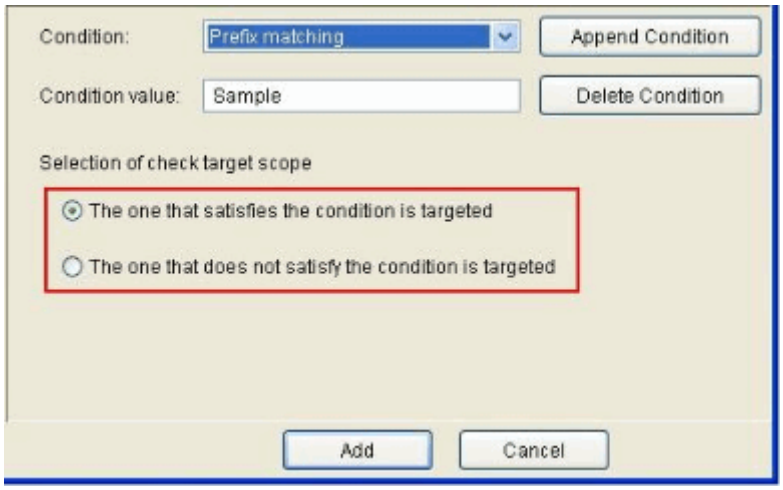
Set the condition value corresponding to a condition.



*If the Java element selected from the Java element tree is branch element, input is disabled.

Selection of Check Target Scope

For the selection of the check target scope, set whether the elements will becomes the extraction target in which cases for the conditions set in the condition configuration view.



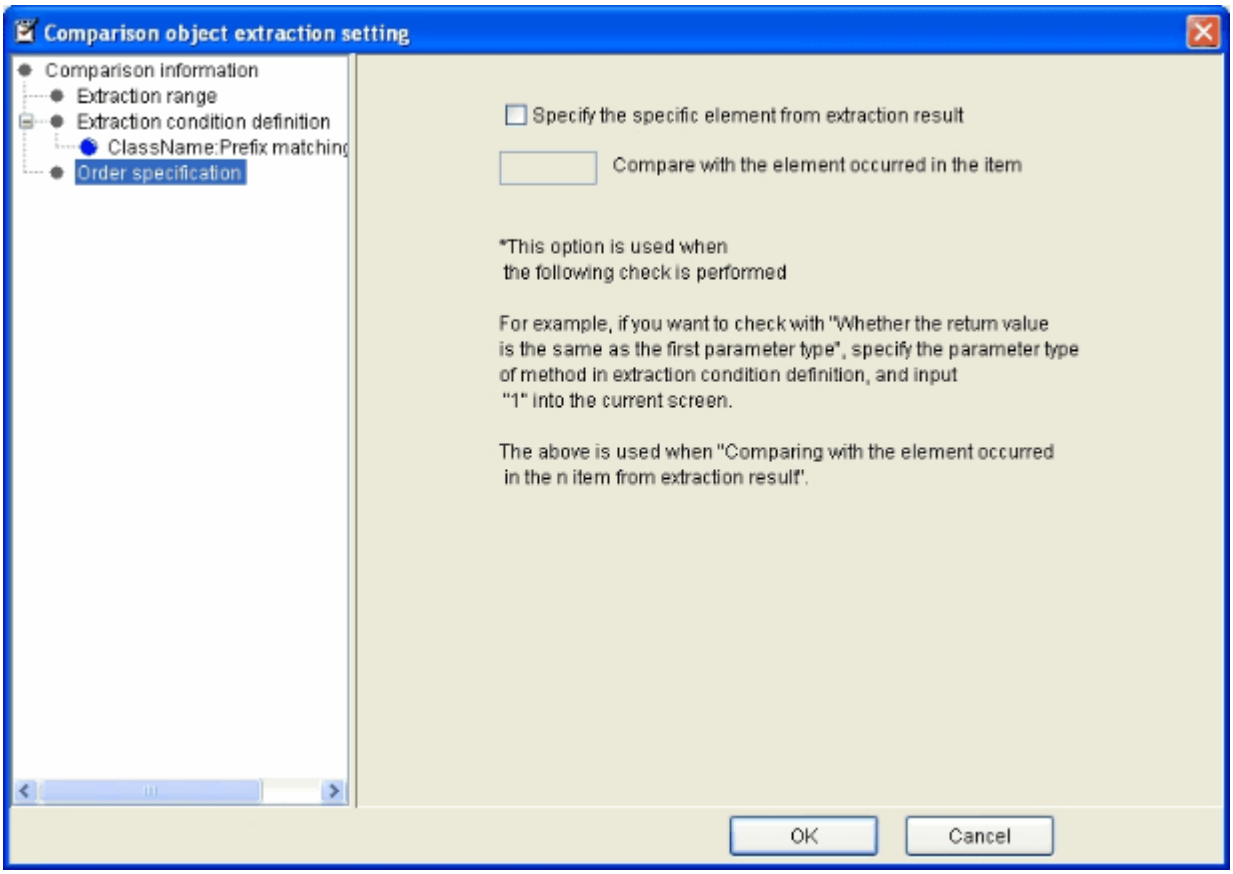
The following table describes the options and their meanings.

Item	Description
Targeted at the one that satisfies the condition	Set the one that satisfies the condition set in the condition configuration view as the comparison target
Targeted at the one that does not satisfy the condition	Set the one that does not satisfy the condition set in the condition configuration view as the comparison target

*If the Java element selected from the Java element tree is branch element, "Targeted at the one that does not satisfy the condition" cannot be selected.

Order Specification

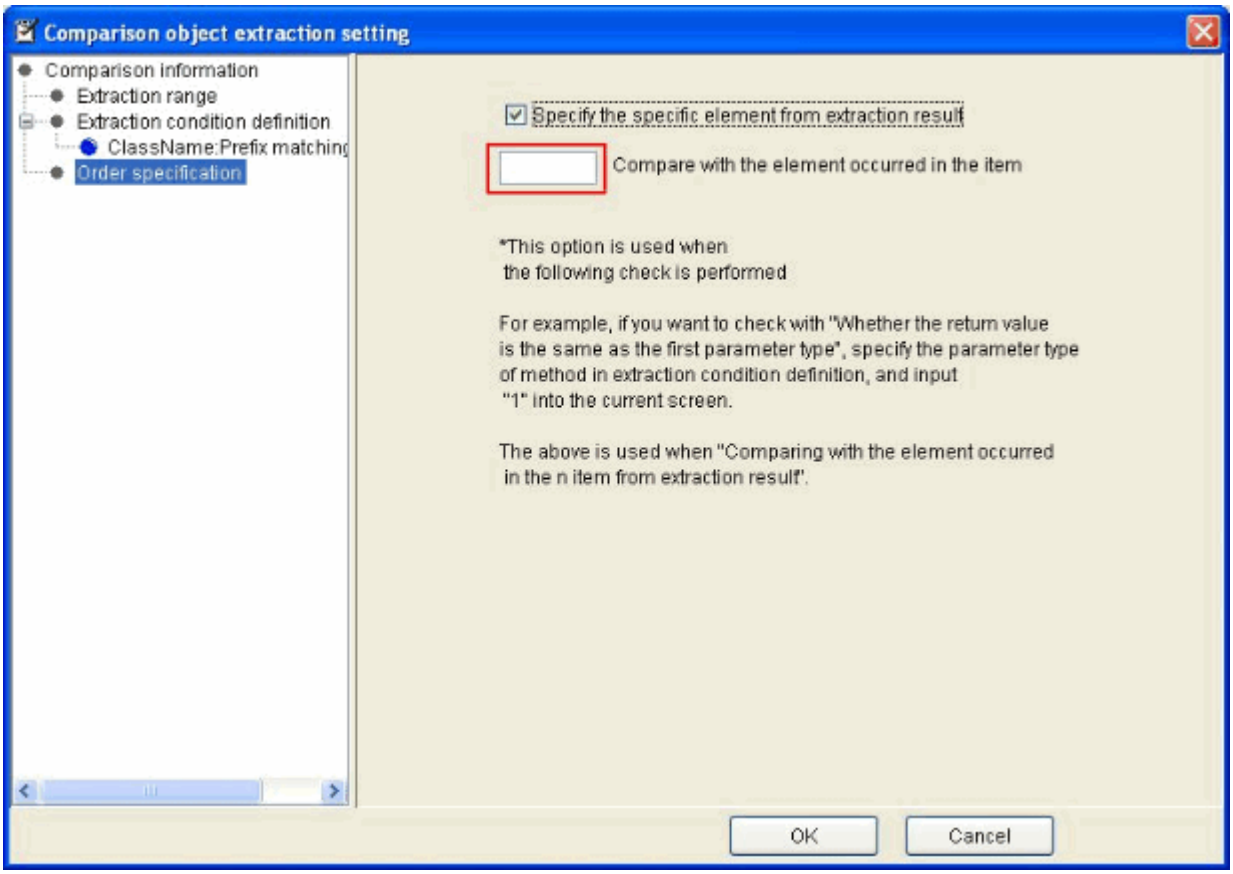
After "Order specification" is selected in the extraction information tree view, the order specification view is displayed.



In the order specification view, whether to compare with the information that appears simultaneously with the extraction information can be specified.

When this function is enabled, please select the [Specify the specific element from extraction result] checkbox.

Set the occurrence count.

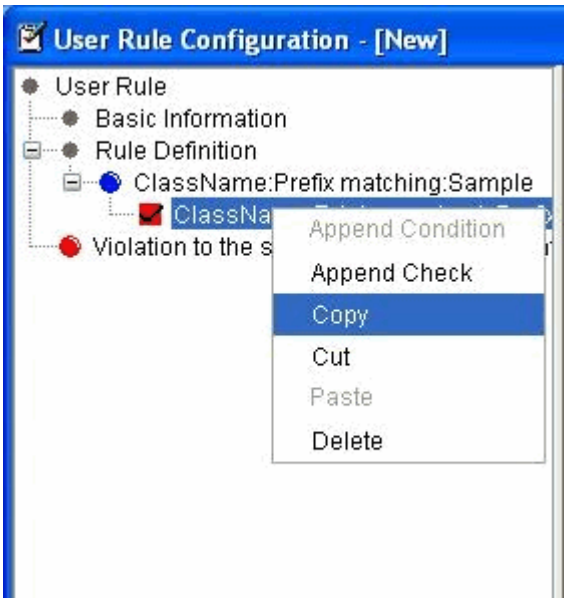


Copy

Copy the content under "Rule Definition".

When performing right-click on the condition or check under of the rule tree view in the status of being selected, the pop-up menu is displayed.

Select [Copy] from the pop-up menu displayed by right-click to copy the selected condition or check.



* When a condition is copied, it can be pasted to "Rule Definition" or to the selected condition.

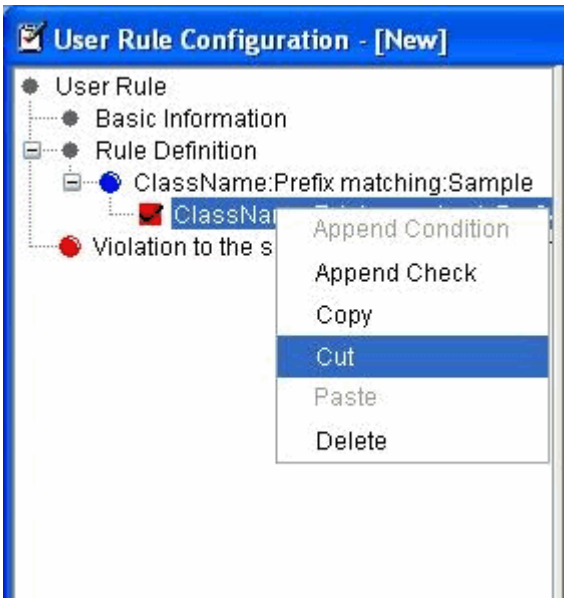
When a check is copied, it can be pasted to the selected condition or to the check.

Cut

Cut the content under "Rule Definition".

When performing right-click on the condition or check under "Rule Definition" of the rule tree view in the status of being selected, the pop-up menu is displayed.

Select [Cut] from the pop-up menu displayed by right-click to cut the selected condition or check.



* When a condition is cut, it can be pasted to "Rule Definition" or to the selected condition.

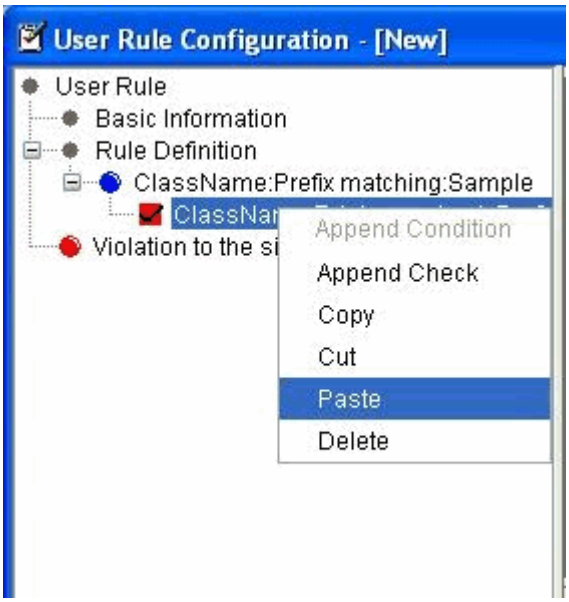
When a check is cut, it can be pasted to the selected condition or to the check.

Paste

Paste a condition or check.

When performing right-click on the condition or check under "Rule Definition" of the rule tree view in the status of being selected, the pop-up menu is displayed.

Select [Paste] from the pop-up menu displayed by right-click to paste the copied or cut condition or check to the selected condition or check.



* A condition can be pasted to "Rule Definition" or to the selected condition.

A check can be pasted to the selected condition or to the check.

Delete

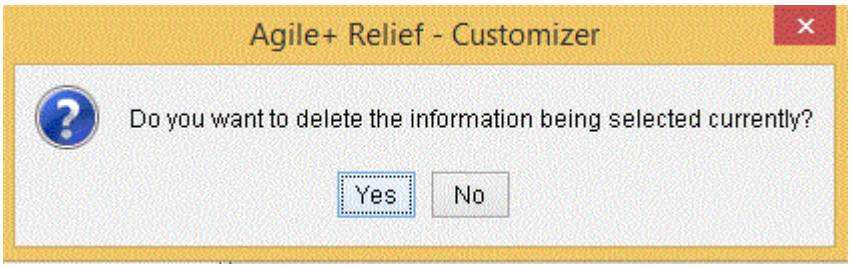
Delete the selected information (condition or check).

When performing right-click on the condition or check under "Rule Definition" of the rule tree view in the status of being selected, the pop-up menu is displayed.

Select [Delete] from the pop-up menu displayed by right-click to delete the selected condition or check.

When "Yes" is selected in the message of deletion confirmation, the selected information is deleted.

When "No" is selected, the deletion is cancelled.

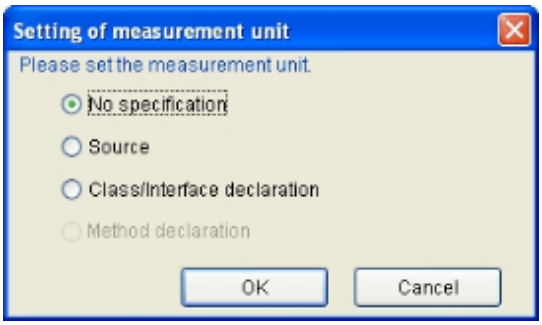


3.5.2.5.2 "Set Measurement Unit" option

Define the measurement unit of the Java element when performing "Number check".

When "Number check" is selected in "Check content", "Measurement unit setting option" can be selected.

Set the measurement unit for number check in [Setting of measurement unit] dialog.



The following table shows the list of options and their meanings.

Measurement Unit	Meaning
No specification	Measurement of the range specified in the condition element
Source	Measurement of the entire source
Class/interface declaration	Measurement in class/interface declaration
Method declaration	Measurement in each method declaration

(Rule example) At most three local variables are defined.

(Details of measurement unit)

The difference in the number of measurement when the measurement unit is changed in the (Rule example) is described as follows.

For the condition element, set the following information.

- Java element: source
- Condition: All
- Condition value: -

For the check element, set the following information.

- Java element: Variable name (of local variable)
- Check method: Number check
- Condition: Below
- Condition value: 3

Violation judgment method: violation to the situation when the element of check sequence is not satisfied.

After the following Sample.java has been checked according to the above rule, the result is as follows.

<No specification of measurement unit (default)>

The number of local variables in the entire source will be measured in case of (Rule example) because of the dependence on the range specified in condition element.

The check result will contain four local variables in the source.

Rule violation will occur in the (Rule example).

<Source>

The number of local variables in the entire source will be measured.

The check result will contain four local variables exist in the source.

Rule violation will occur in the (Rule example).

<Class/interface declaration>

The number of local variables in each class/interface will be measured.

The check result will contain four local variables exist in the Sample class.

Rule violation will occur in the (Rule example).

<Method declaration>

The number of local variables in each method will be measured.

The check result will contain two local variables exist in the paint method.

Two local variables will exist in the init method.

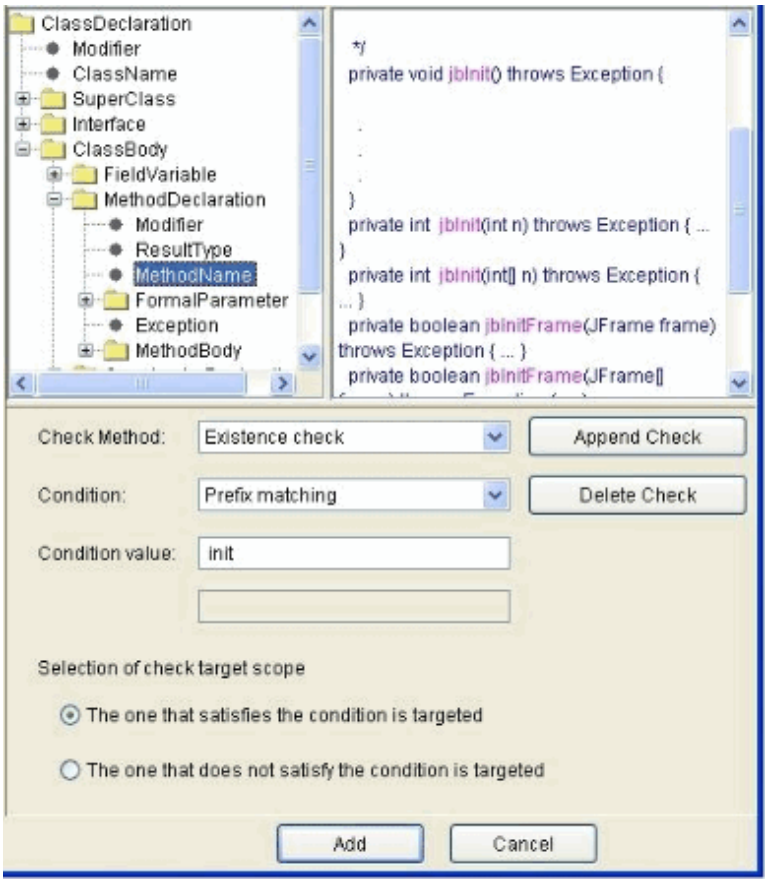
No variable will exist in the run method.

Rule violation will not occur in the (Rule example).

```
// Sample.java
import java.awt.Graphics;
import java.applet.*;
import java.util.*;
import co.jp.*;

// Sample classpublic class Sample extends Applet {

Vector v = new Vector();

// Drawing
public void paint( Graphics g ) {
int xPos = 25;
int yPos = 25;

g.drawString( "Hallo,Java!!", xPos, yPos );
}

public void init(){
  int xPos = 25;
  int yPos = 25;
}

public void run(){
}

}
```

3.5.2.5.3 Set Check Elements

Set the check elements in the check element configuration view.



Item description

Item	Description
Java element tree	Java elements are displayed in the tree. Select the Java element that becomes the check target.
Sample source	The sample source for confirming the selected Java elements is displayed.
Check content	Set the check content for the selected Java element.
Condition	Set the condition for the check content.
Condition value	Set the condition value limited in the condition.
Selection of check target scope	Select the check target scope.
[Append check] button	Append a check element.
[Delete check] button	Delete a check element.
[Measurement unit] button	The "Setting of measurement unit" dialog is displayed. When "Number check" is selected in check content, the "Measurement unit" button will be displayed.
[Extraction condition] button	The "Comparison object extraction setting" dialog is displayed. When "Extraction condition matching" is selected in condition, the "Extraction condition" button will be displayed.

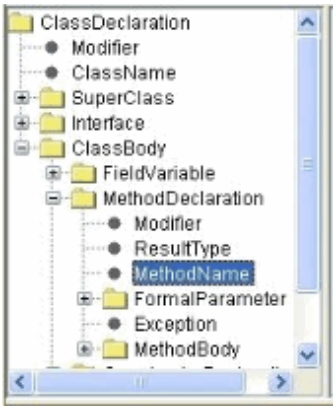
The following section describes the detail of each item.

Java Element Tree

Select the Java element that becomes the check target from the Java element tree.

(Available Range)

- In condition setting, if the selected Java element is branch elements, the Java element itself or its subordinate Java elements will be selected.
- In the condition setting, if the selected Java element is terminal elements, the branch element at least one level over the selected terminal element or the subordinate Java elements will be selected.



is branch element, under which Java elements exist.

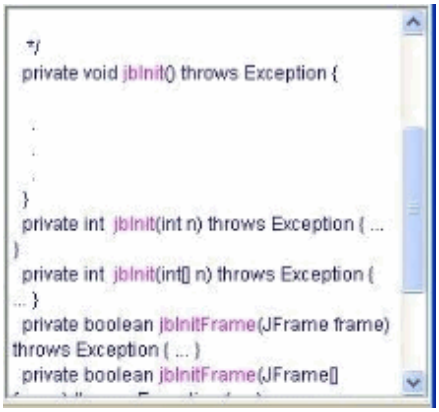
is terminal element.

* After the Java element is selected, the Java element corresponding to Sample source will be displayed in colors.

* After the Java element is selected, Check content, Condition, and Condition value that can be specified can be selected.

Sample Source

The Java elements selected from the Java element tree are displayed as the sample source that confirms the location of Java source.



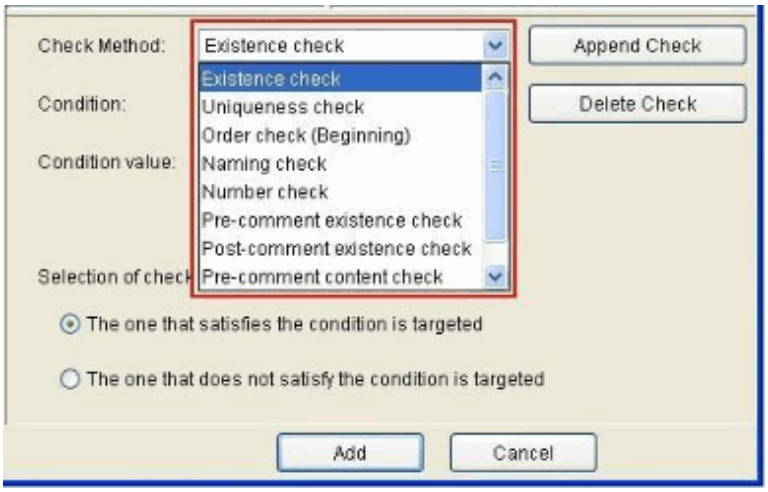
If the Java element selected from the Java element tree is branch element, the Java element will be displayed in green.

If the Java element selected from the Java element tree is terminal element, the upper-level Java elements will be displayed in green, and the selected Java elements will be displayed in amaranth.

Check Content

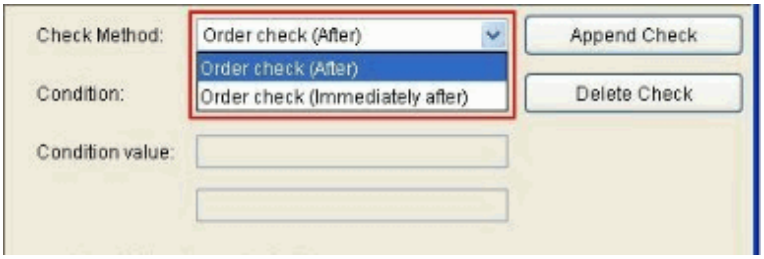
Set the check content for the Java element (check target) selected from the Java element tree.

Select the check content from the combo box.



In addition, after the check is created (click the "Append check" button) through the check content of "Order check (Beginning)", when "Order check (Beginning)" is selected for appending a check,

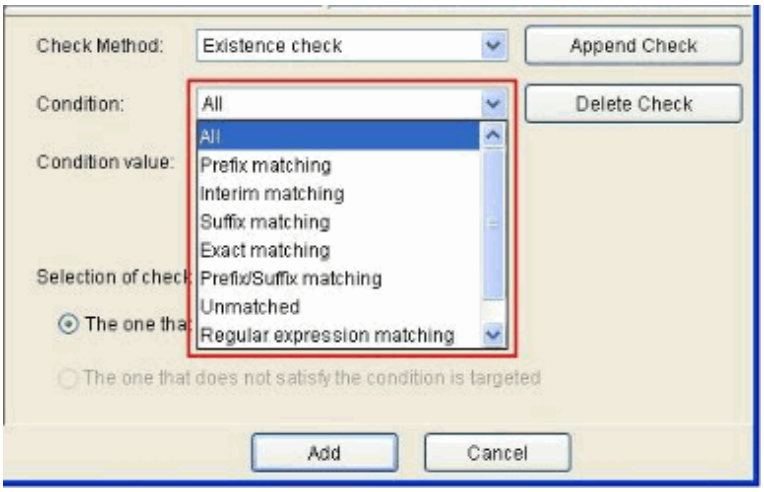
the content that can be selected in check content will become "Order check (After)" and "Order check (Immediately after)".



The following check operations can be performed for the Java elements selected from the Java element tree within the check scope limited in the condition part.

Check content	Description
Existence check	Check whether the Java element that satisfies the condition exists.
Uniqueness check	Check whether only one Java element satisfies the condition.
Order check (Beginning) Order check (After) Order check(Immediately after)	Check whether the Java element that satisfies the condition is described in specified order. For this check, "Order check (Beginning)" must be specified at initial stage. Then, specify to "Order check (After)" or "Order check (Immediately after)" arbitrarily.
Naming check	Check whether the Java element matches to the condition.
Number check	Check whether the number of Java elements that satisfies the condition matches to the condition.
Pre-comment existence check	Check whether the comment is described before the Java element.
Post-comment existence check	Check whether the comment is described after the Java element.
Pre-comment content check	Check whether the specified string exists in the comment before Java element.
Post-comment content check	Check whether the specified string exists in the comment after the Java element.

Condition



Set the conditions of check content.

The conditions are different depending on check content and Java elements selected from the Java element tree.

Check content: Existence check, Uniqueness check, Order check(Beginning), Order check(After), Order check(Immediately after), Naming check, Pre-comment content check, and Post-comment content check.

The Java elements selected from the Java element tree are checked using condition value according to the following conditions.

Condition	Description
All	Do not check (All elements satisfy the condition)
Prefix matching	Check the elements according to the prefix matching.
Interim matching	Check the elements according to the interim matching.
Suffix matching	Check whether it is matched according to the suffix matching.
Exact matching	Check whether it is matched according to exact matching.
Prefix/suffix matching	Check whether it is matched according to the prefix/suffix matching.
Unmatched	Check whether the elements are not matched.
Regular expression matching	Check whether it is matched according to the regular expression matching.
Extraction condition matching	Check whether it is matched with the information extracted through "Specification of extraction condition".

* Extraction condition matching can be selected when Check content is Existence check, Uniqueness check, or Naming check.

When the Java element selected from the Java element tree is branch element, only "All" can be selected.

Check content: Number check

The Java elements selected from the Java element tree are checked according to the following conditions.

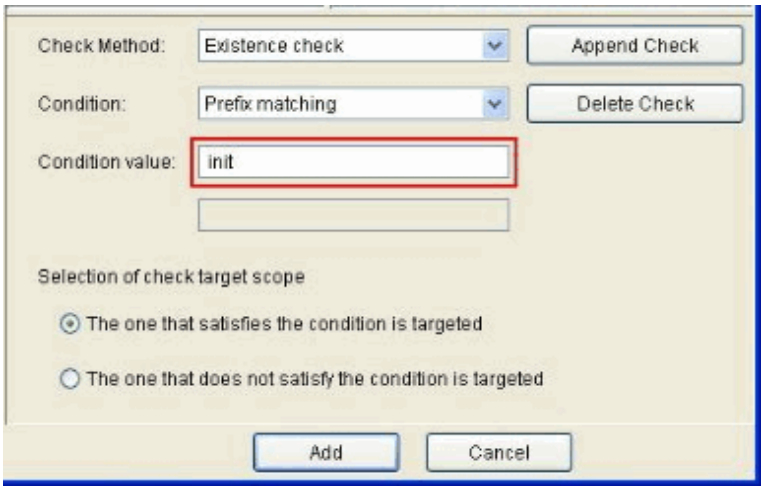
Condition	Description
Same	Check whether the number of Java elements is the same as the specified number.
Above	Check whether the number of Java elements is greater than the specified number.
Below	Check whether the number of Java elements is smaller than the specified number.

Check content: Pre-comment existence check and post-comment existence check

Cannot be selected.

Condition Value

Input the condition value limited by the condition.



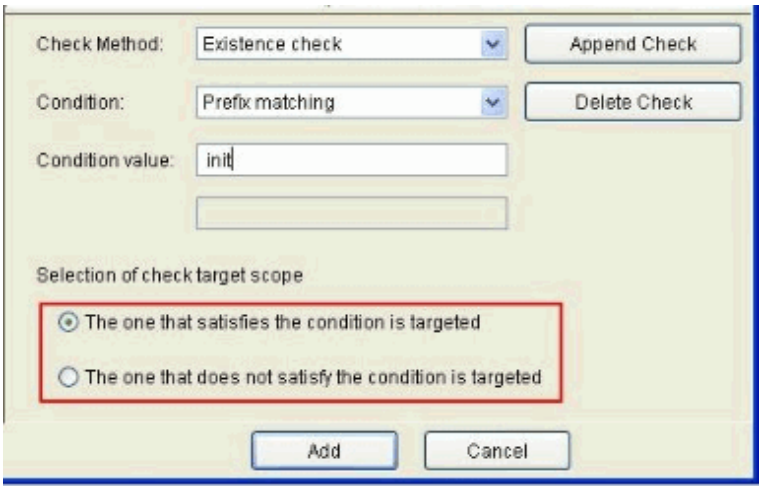
* When "Prefix/Suffix matching" has been specified in the condition, the second line of the input column of condition value is enabled.

In case of "Prefix/Suffix matching", the upper side is for prefix matching, and the lower side is for suffix matching.

* When "Number check" has been specified in condition content, only numeric values (0~999) can be input.

* If the Java element selected from the Java element tree is branch element, and "All" and "Extraction condition matching" is selected for condition, input will be disabled.

Selection of check target scope



The following table describes the options and their meanings.

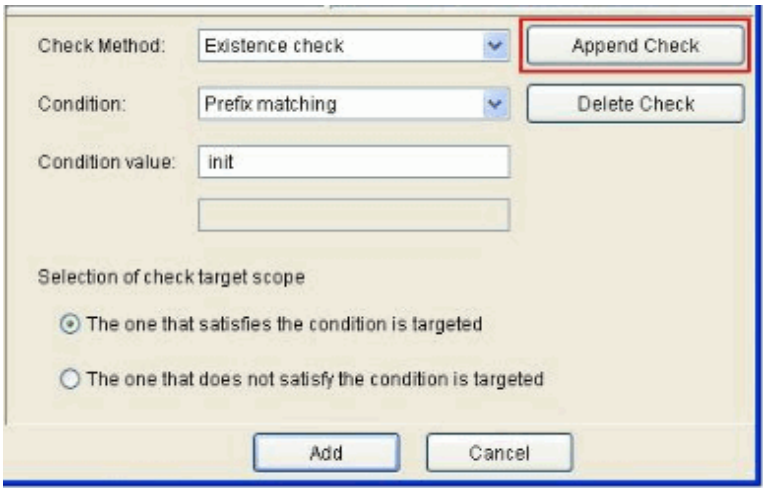
Option Name	Meaning
The one that satisfies the condition is targeted	Selected when the one that satisfies the conditions set in the condition configuration view becomes the check target.
The one that does not satisfy the condition is targeted	Selected when the one that does not satisfy the conditions set in the condition setting view becomes the check target.

* "Tthe one that does not satisfy the condition is targeted " cannot be selected in the following cases.

- The Java element selected from the Java element tree is branch elements.

- "All" is selected for condition.

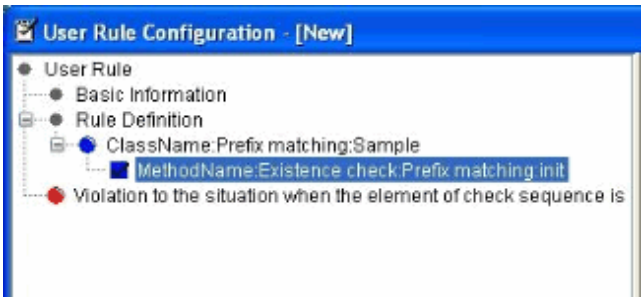
Append Check



Select "Append check " to add check elements set in the check element configuration view.

* After "New condition" has been added, it is necessary to add by the "Append check" button in the check element setting view.

After the check has been added, "New check" of the rule tree view is replaced with the set check contents.



(Display check in rule tree view)

MethodName:Existence check:Prefix matching:init

From the left hand side, it becomes "Java element selected in Java element tree of check target scope selection: Check content: Condition: Condition Value: upper side: Condition Value: lower side"

* The following icons indicate the selection of the check target scope according to the selection status.

: Select "The one that satisfies the condition is targeted ".

: Select "The one that does not satisfy the condition is targeted ".

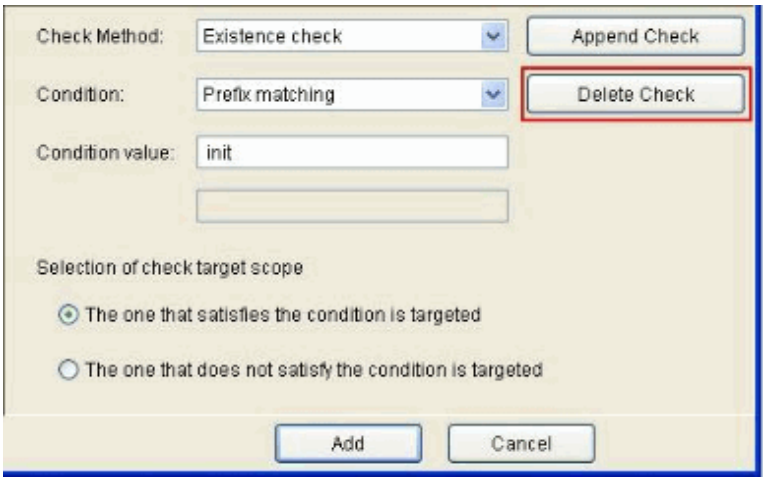
* When condition is set to "All" or "Extraction condition matching", ": upper side of condition value" does not exist.

* When condition is set to "Prefix/Suffix matching", ": bottom side of condition value" is displayed.

In addition, the [Append check] button is changed to [Modify check].

* To modify the settings of check, please make sure to click the [Modify Check] button.

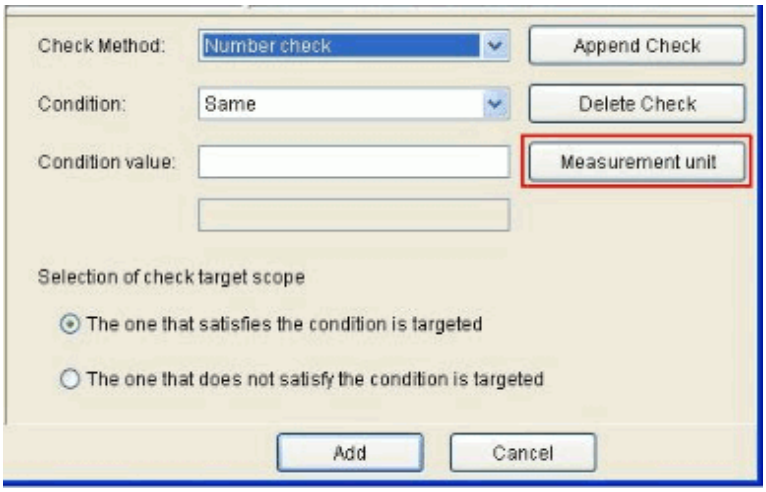
Delete Check



Click the [Delete Check] button to delete "Check".

* Other check elements existing under the selected condition will also be deleted.

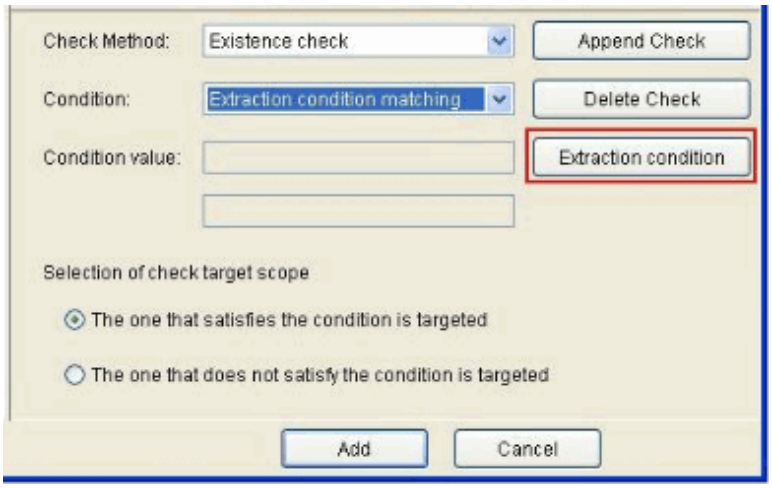
Set Measurement Unit



When "Number check" is selected in "Check content", the [Measurement unit] button will be displayed.

Click the [Measurement unit] button, the measurement unit setting dialog will be displayed.

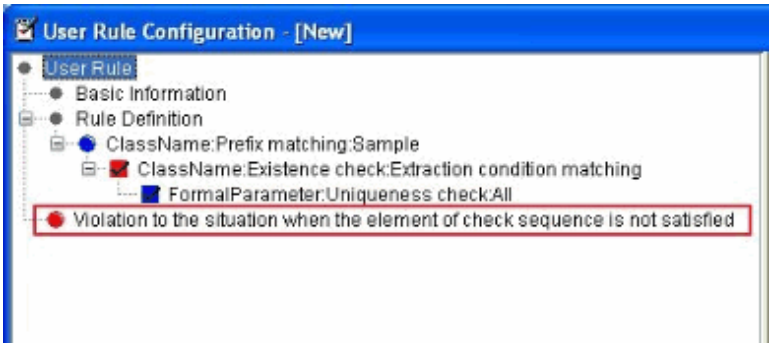
Specify Extraction Condition



When "Extraction condition matching" is selected in condition, the [Extraction condition] button will be displayed.

Click the [Extraction condition] button, and extraction condition specification dialog of comparison target will be displayed.

3.5.2.6 Rule Violation Judgment



Determine whether the elements in the check part have rule violations in any case.

Left-click the rule violation judgment part, and it will switch to display the rule violation judgment.

List of displayed content

Displayed content	Action
Violation to the situation when the element of check sequence is not satisfied	Rule violation occurs when the element of check sequence is not satisfied.
Violation to the situation when the element of check sequence is satisfied	Rule violation occurs when the element of check sequence is satisfied.

3.5.3 Add User Rule by Copying

Add a new user rule by copying the existing user rule definition.

<Operations>

1. Select the user rule to be copied from the rule list.

2. Select [User Rules] menu > [Duplicate]
3. The dialog of adding the copy of user rule is displayed. Input the new rule code and click the [OK] button.

When the [OK] button is clicked, the user rules with same definition content as the copy source will be added as the new rule code.
When the [Cancel] button is clicked, add of copy will be cancelled.

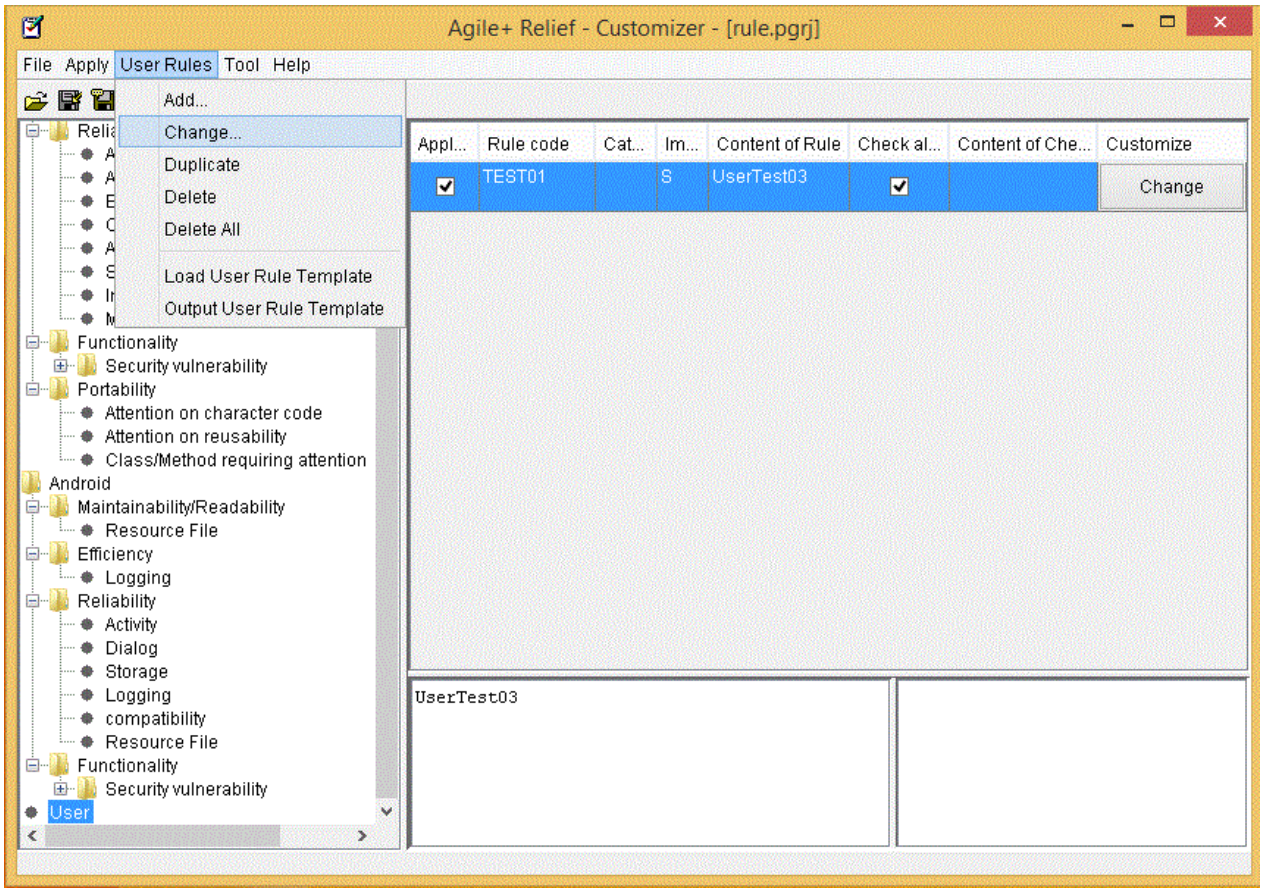
3.5.4 Modify User Rule

Modify the user rule definition.

Click the [Change] button. [User Rule Configuration] dialog will be displayed.

<Operations>

1. Select the rule to be modified from the rule list.
2. Select [User Rules] menu > [Change] or click the [Change] button of the customized column.



3. Modify the rule definition in the "User Rule Configuration".

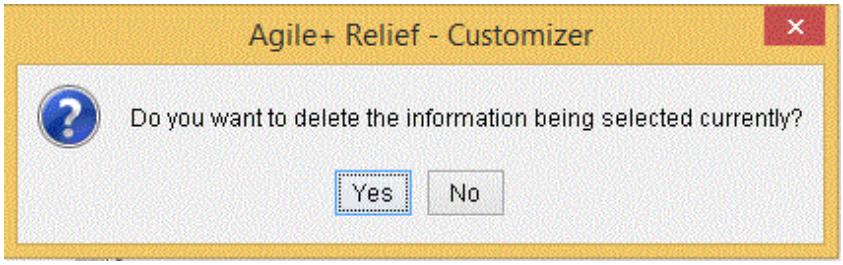
For details about the [User Rule Configuration] dialog, please refer to "Create a new user rule".

3.5.5 Delete User Rule

Delete any user rule from the rule list.

<Operations>

1. Select the user rule to be deleted from the rule list, and select [User Rules] > [Delete].
2. The deletion confirmation message is displayed, select [Yes].



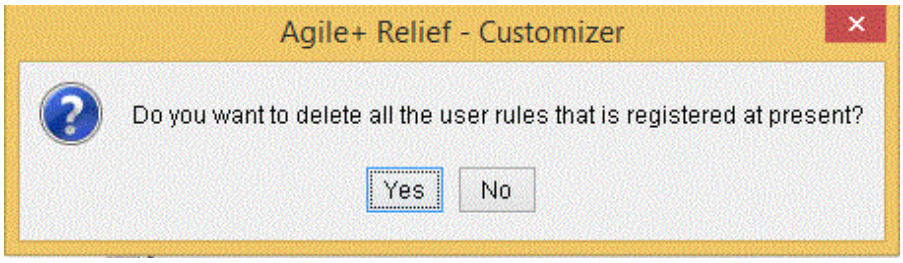
- When the [Yes] button is clicked, the user rules will be deleted.
- When the [No] button is clicked, the deletion will be cancelled.

3.5.6 Delete All User Rules

Delete all user rules from the rule list.

<Operations>

1. Select a user from the rule category tree.
2. Select [User Rules] menu > [Delete All], or select [Delete All] from the pop-up menu displayed by right-click.
3. Once the deletion confirmation message is displayed, and select [Yes].



- When the [Yes] button is clicked, all the user rules will be deleted.
- When the [No] button is clicked, the deletion will be cancelled.

3.5.7 Apply User Rule

The method for applying rules is the same as the one for applying coding rules.

For details, please refer to [Apply Rule].

3.5.8 Load/Output User Rule Template

The user check rule can be read or output as a template.

Through preparation as a template, the defined user rule can be used in other projects.

The sample of the user rule template is prepared under the folder of "Agile+ Relief installation folder\PGReliefJava\sample\UserTemplate".

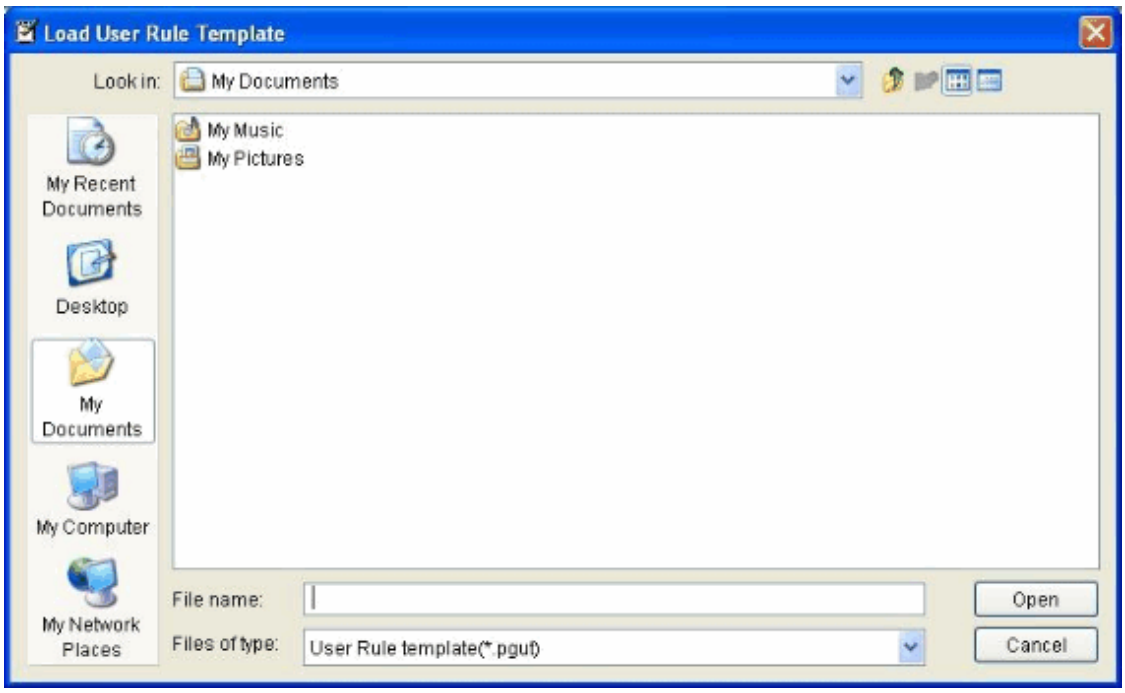
File Name	Description content
sample.pgut	It is the rule example for naming rule check of variable and prohibited command etc.

3.5.8.1 Load User Rule Template

Read the existing template in the [Load User Rule Template] dialog, and add it to the list of user rules.

<Operations>

1. Select [User Rules] menu > [Load User Rule Template].
2. Click the [Open] button next to the selected user rule template (sample.pgut), the user rules defined in the template are read and displayed in the rule list.



When the duplicate user rules exist, whether the duplicate user rule should be removed or not will be confirmed.

3.5.8.2 Output the User Rule Template

Output user rules of the rule list as a template.

After [User Rules] menu > [Output User Rule Template] is selected, the [User Rule Template Output] dialog will be displayed.

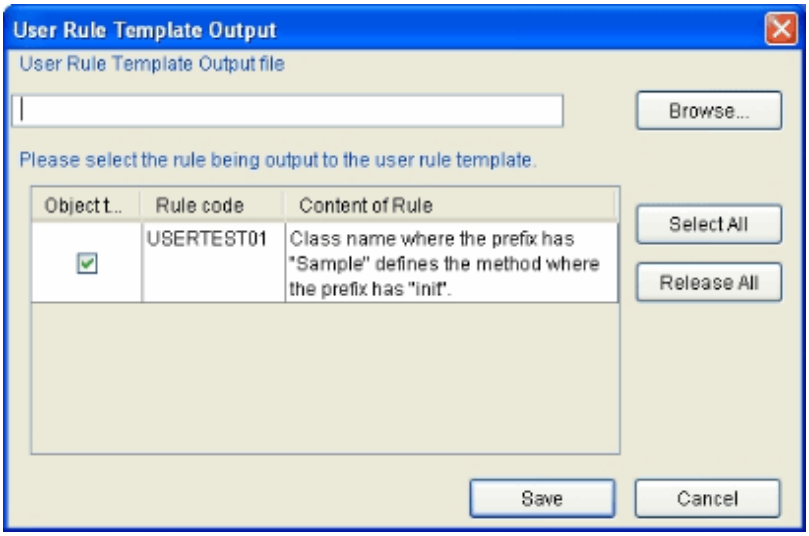
Perform the following setting in the user rule template output dialog.

- Setting of the file path of user rule template output file.

  The user rule template file to be output can be set by clicking the [Browse] button.

- Selection of the user rule for the output object

  The user rule to be output can be set by clicking the [Select All] button and [Release All] button.



  - Click the [Save] button to output the user rule template.
  - Click the [Cancel] button to cancel the output of user rule template.

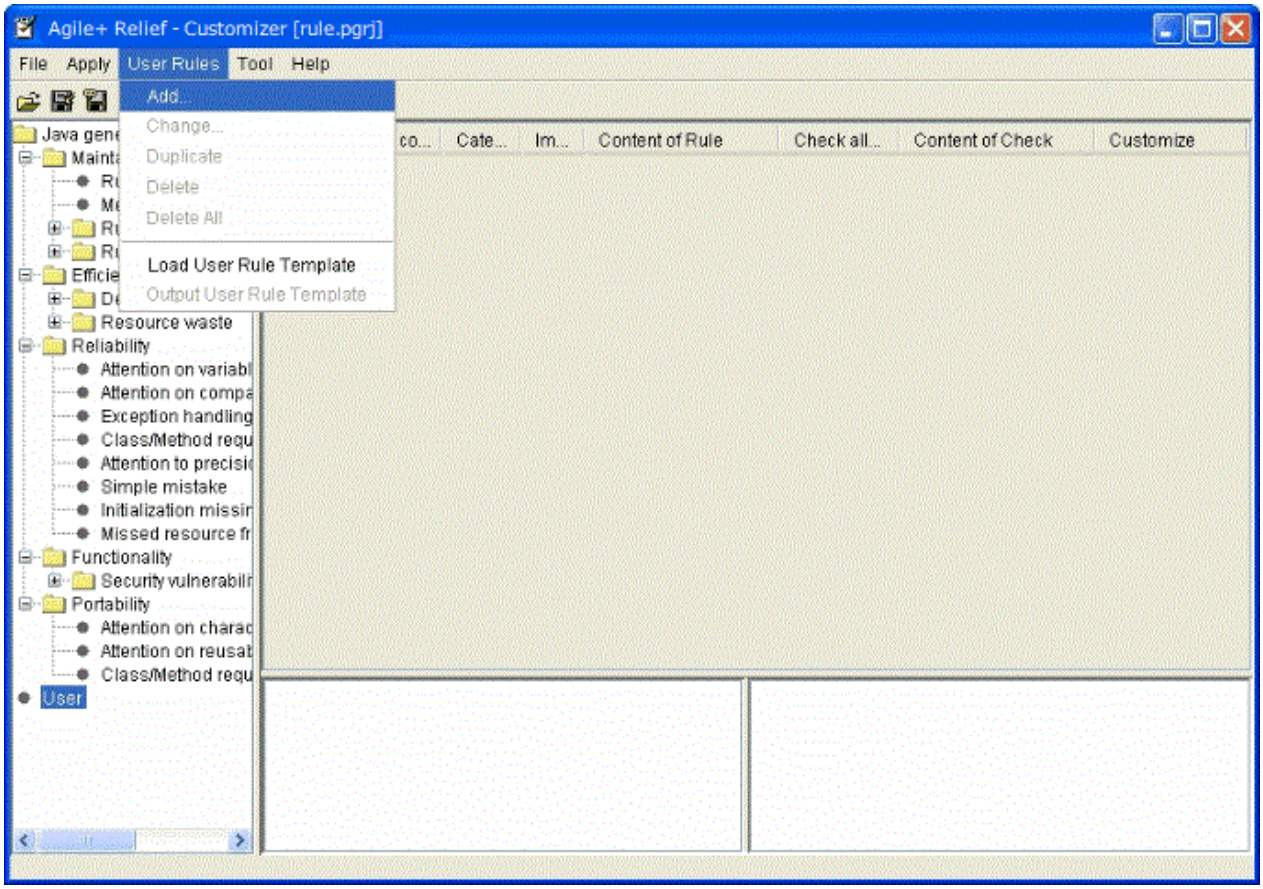
3.5.9 Examples of User Rule Creation

3.5.9.1 Creation Example 1

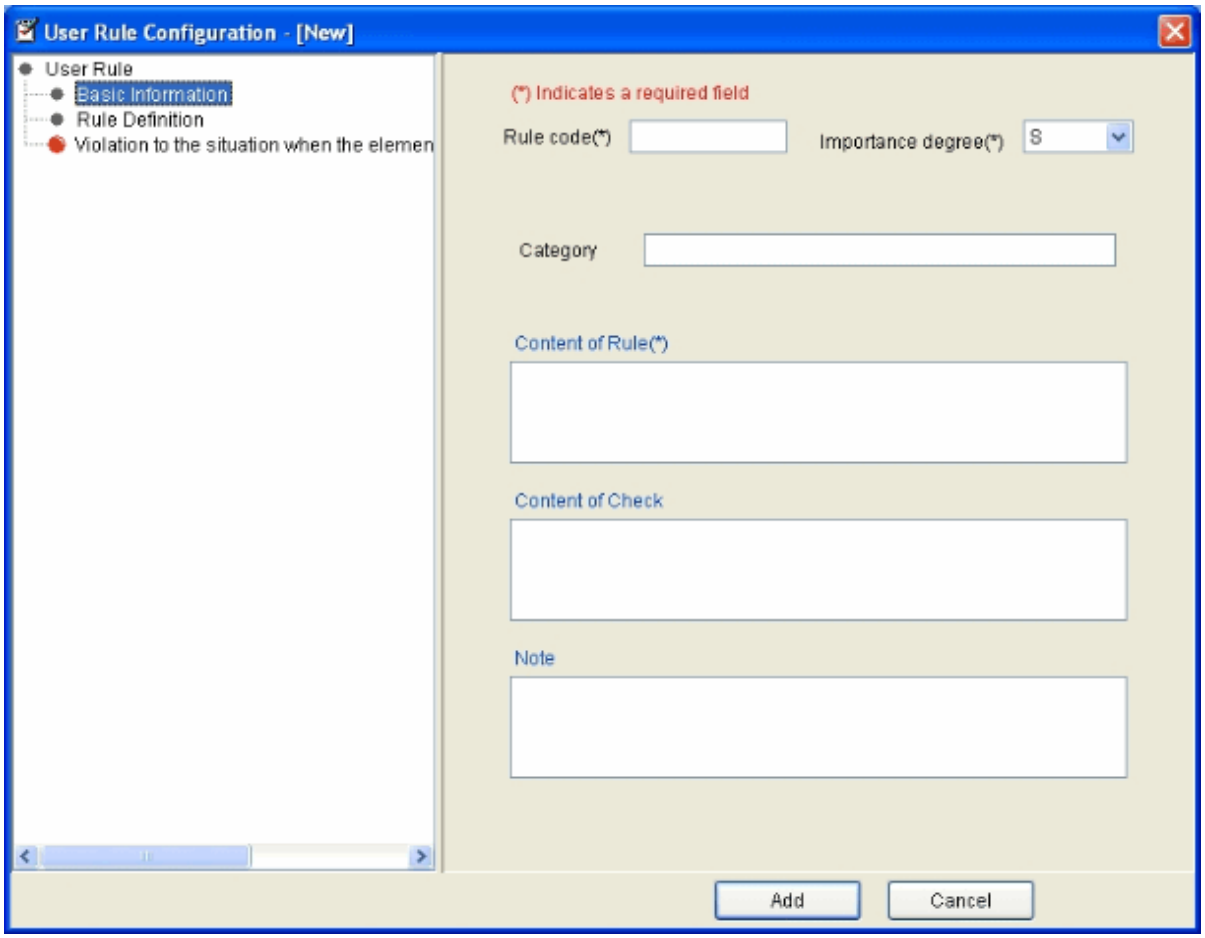
For the class with the initial of class beginning with Sample, the method with the initial of method name is beginning with Init is defined

1. Start the user rule configuration dialog

   Select [User Rules] menu > [Add] of the Customizer.

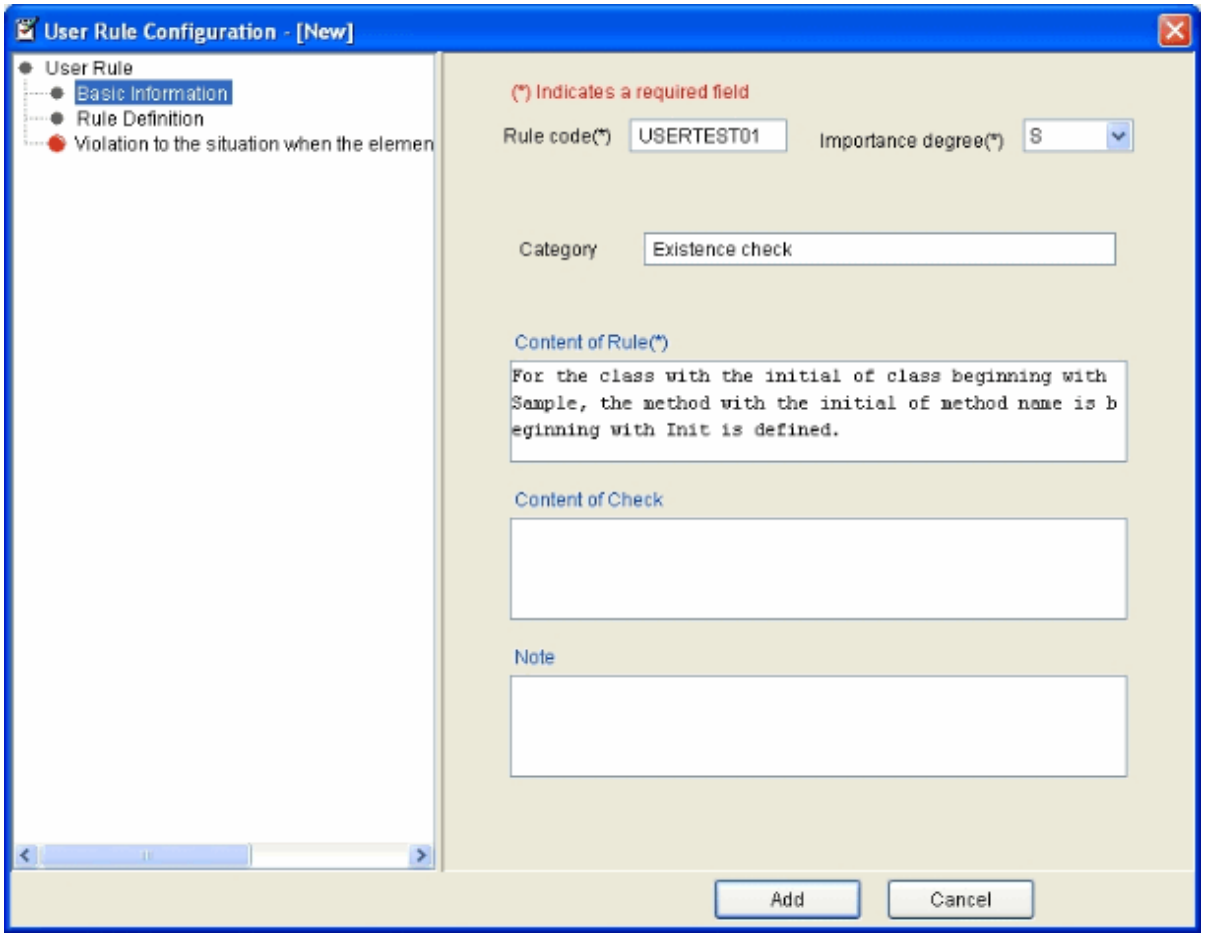


The [User Rule Configuration] dialog is started.



2. Set basic information

Set the basic information of user rule in the [User Rule Configuration] dialog.

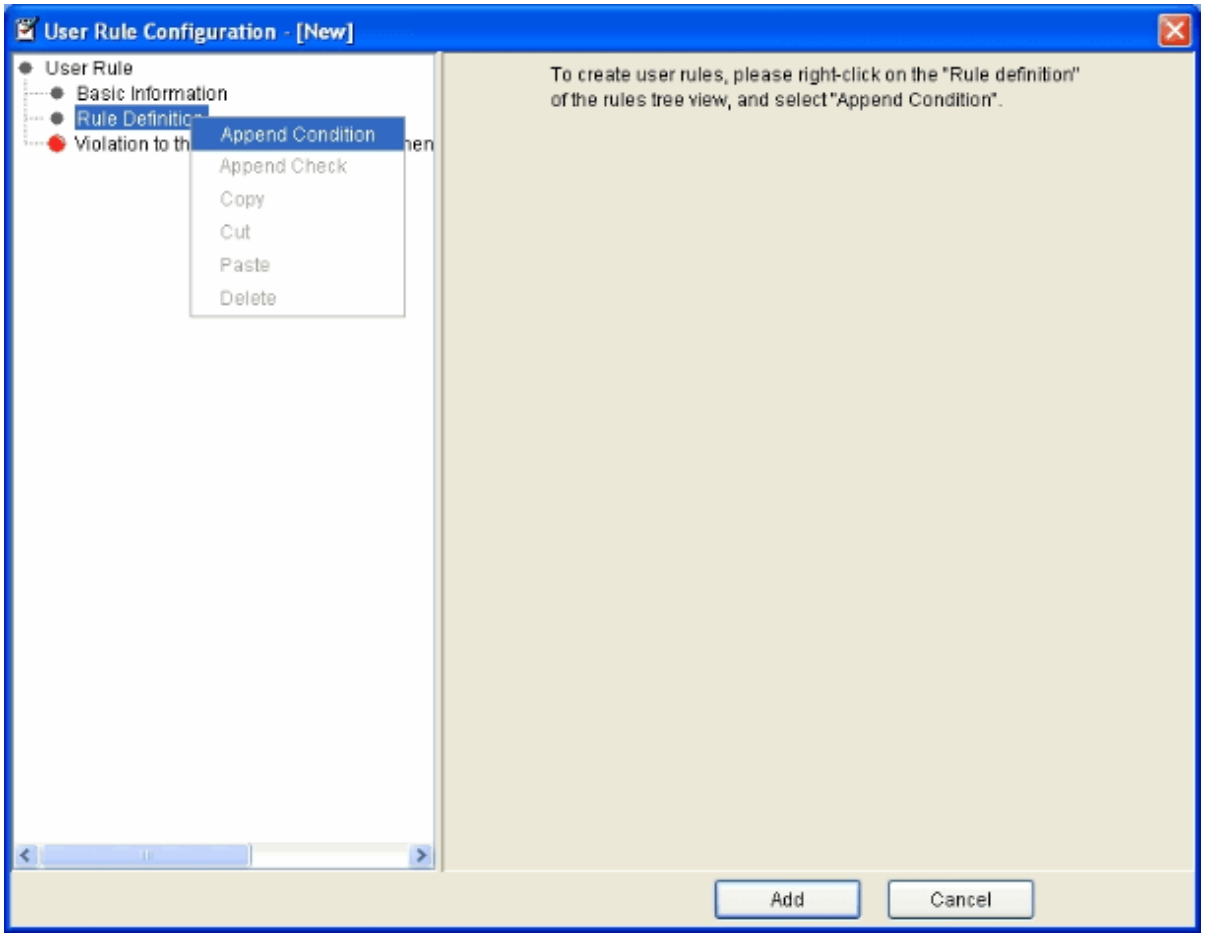


The following items are described as basic information. (In this example, only the mandatory items marked with * are set)

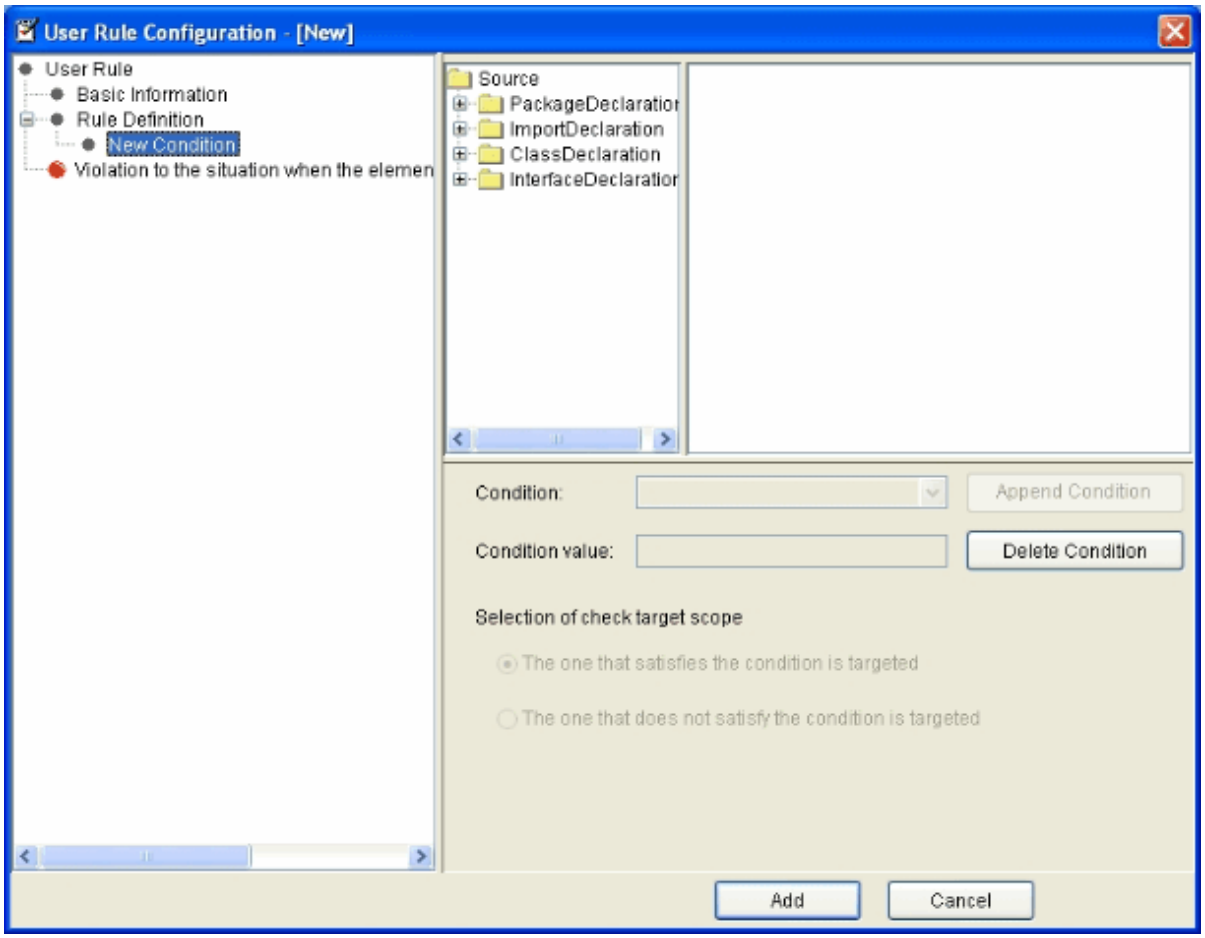
Configuration Items	Value
Rule code	USERTEST01
Importance degree	S
Category	Existence check
Content of Rule	For the class with the initial of class beginning with Sample, the method with the initial of method name is beginning with Init is defined.

3. Append a condition

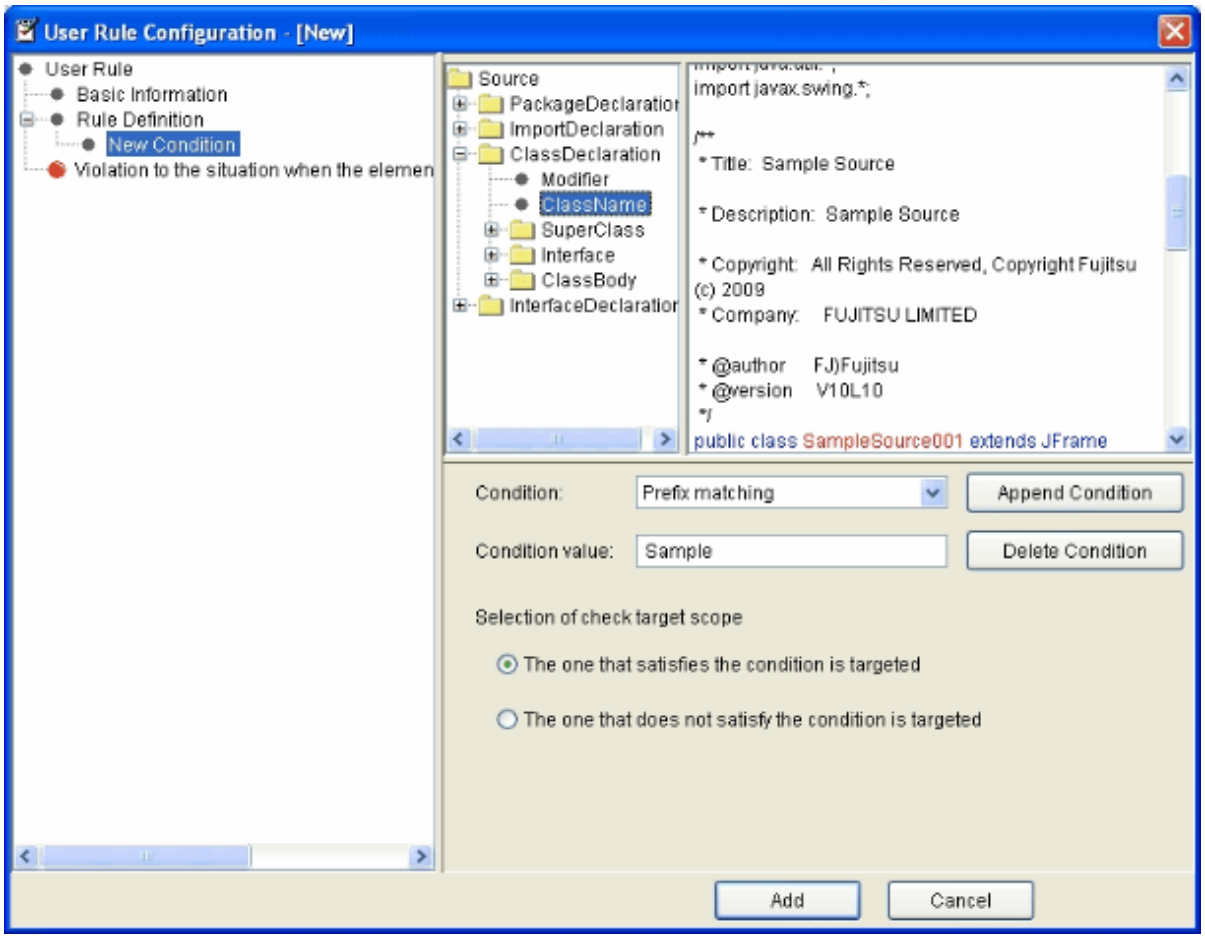
Select "Rule definition" and select the [Append condition] from the pop-up menu displayed by right-click.



"New Condition" is added under "Rule definition". The condition element setting view will be displayed on the right of the dialog.



In the condition element setting view, set the condition for limiting the Java source that becomes the target.



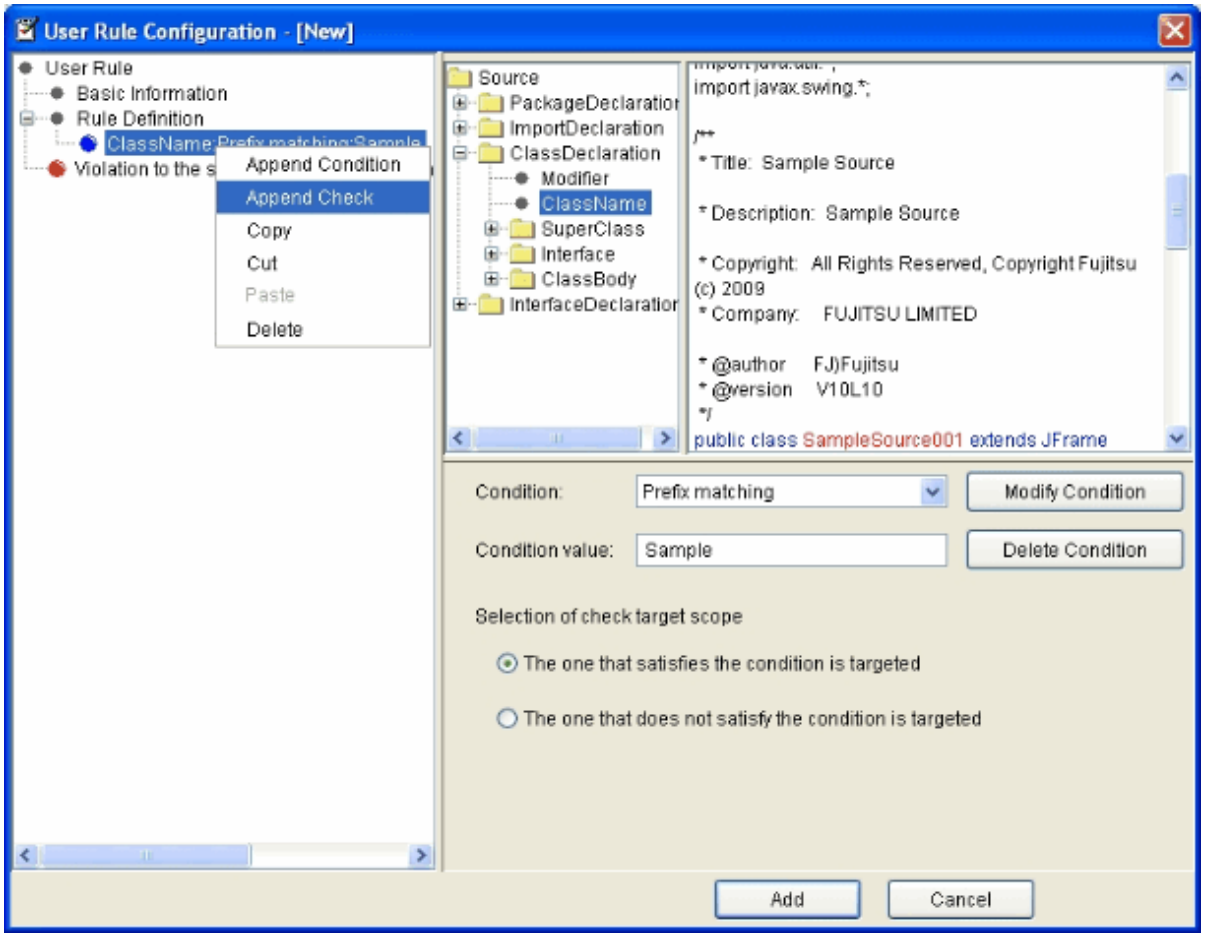
Set the following items according to the condition element setting view.

Configuration Items	Value
Java element	ClassName
Condition	Prefix matching
Condition value	Sample
Selection of check target scope	The one that satisfies the condition is targeted

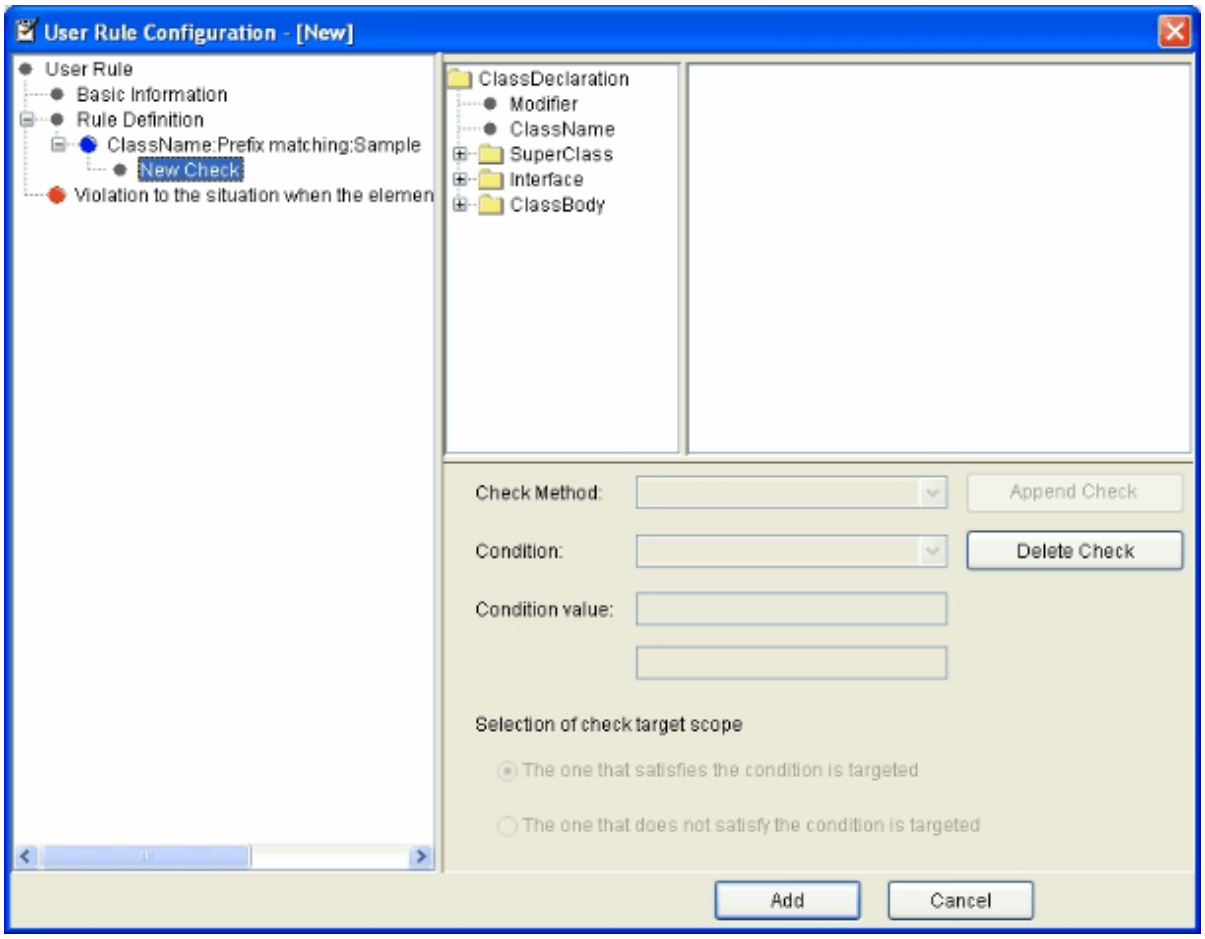
Click the [Append Condition] button to register the content that has been edited here.

4. Append a check

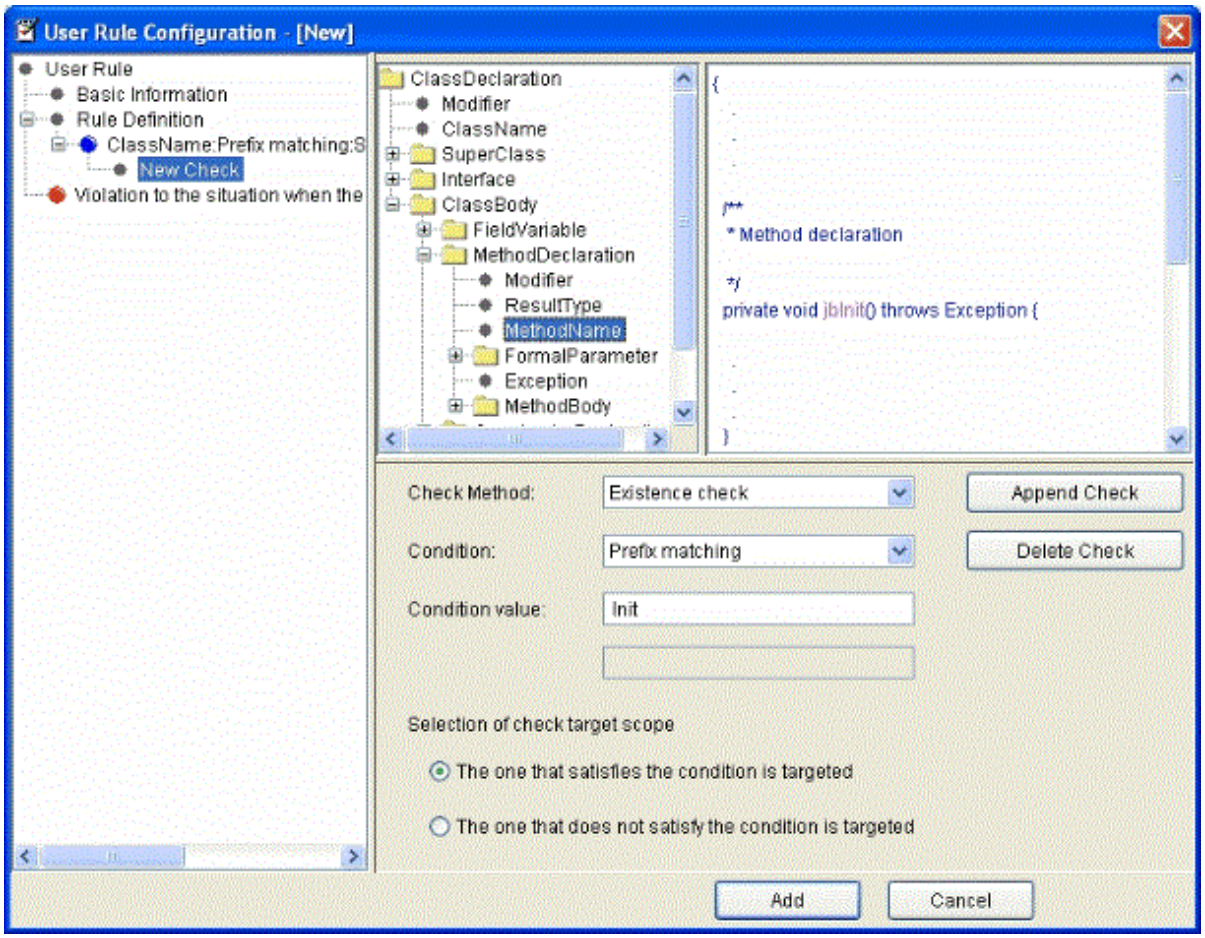
   Select "ClassName:Prefix matching:Sample" condition that has been added in step 3, and select [Append check] from the pop-up menu displayed by right-click.



"New Check" is added under the condition "ClassName:Prefix matching:Sample", and the check element configuration view is displayed on the right of the dialog.



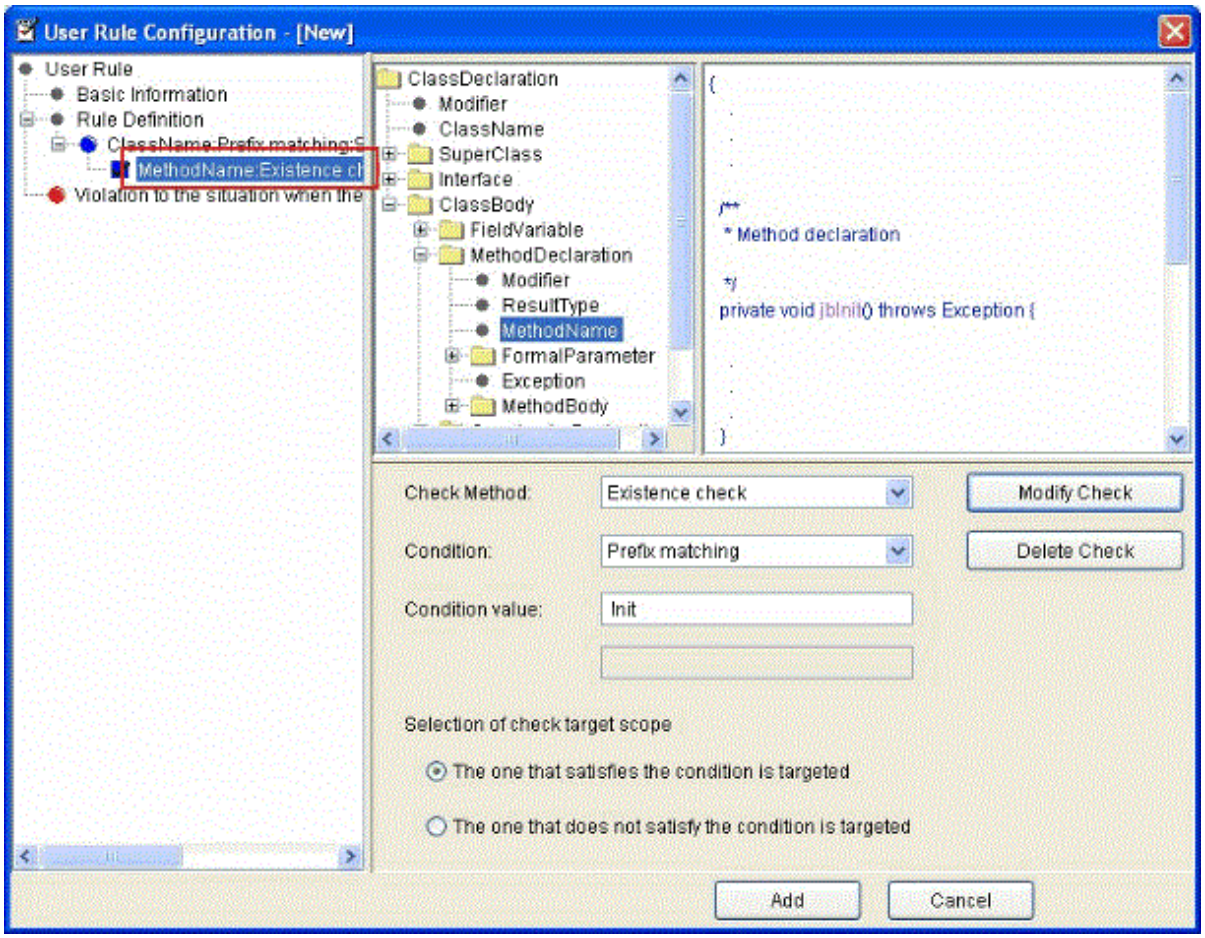
Set the following items in the check element configuration view.



Please set the items according to the following contents.

Configuration Items	Value
Java element	MethodName
Check Method	Existence check
Condition	Prefix matching
Condition value	Init
Select the check target scope	The one that satisfies the condition is targeted

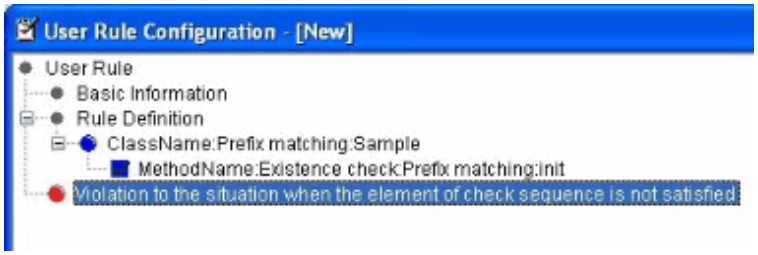
Click the [Append check] button to register the content that has been edited here.



The check part "whether the method name starts with Init" can be created.

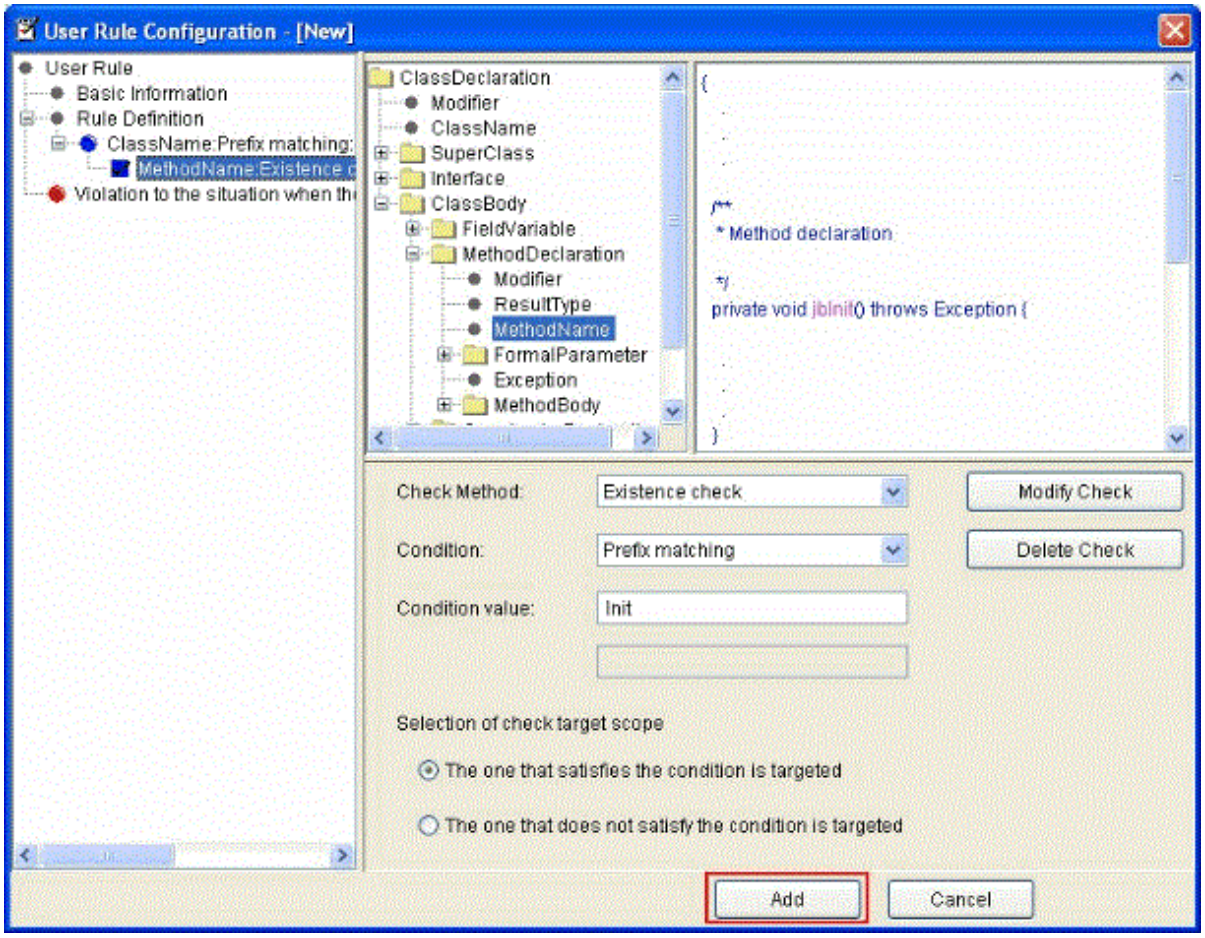
5. Set rule violation judgment.

   When the method with the initial of method name beginning with Init does not exist in the class with the initial of class name beginning with Sample, click the rule violation judgment that turns out to be violation, and set "Violation to the situation when the element of check sequence is not satisfied".

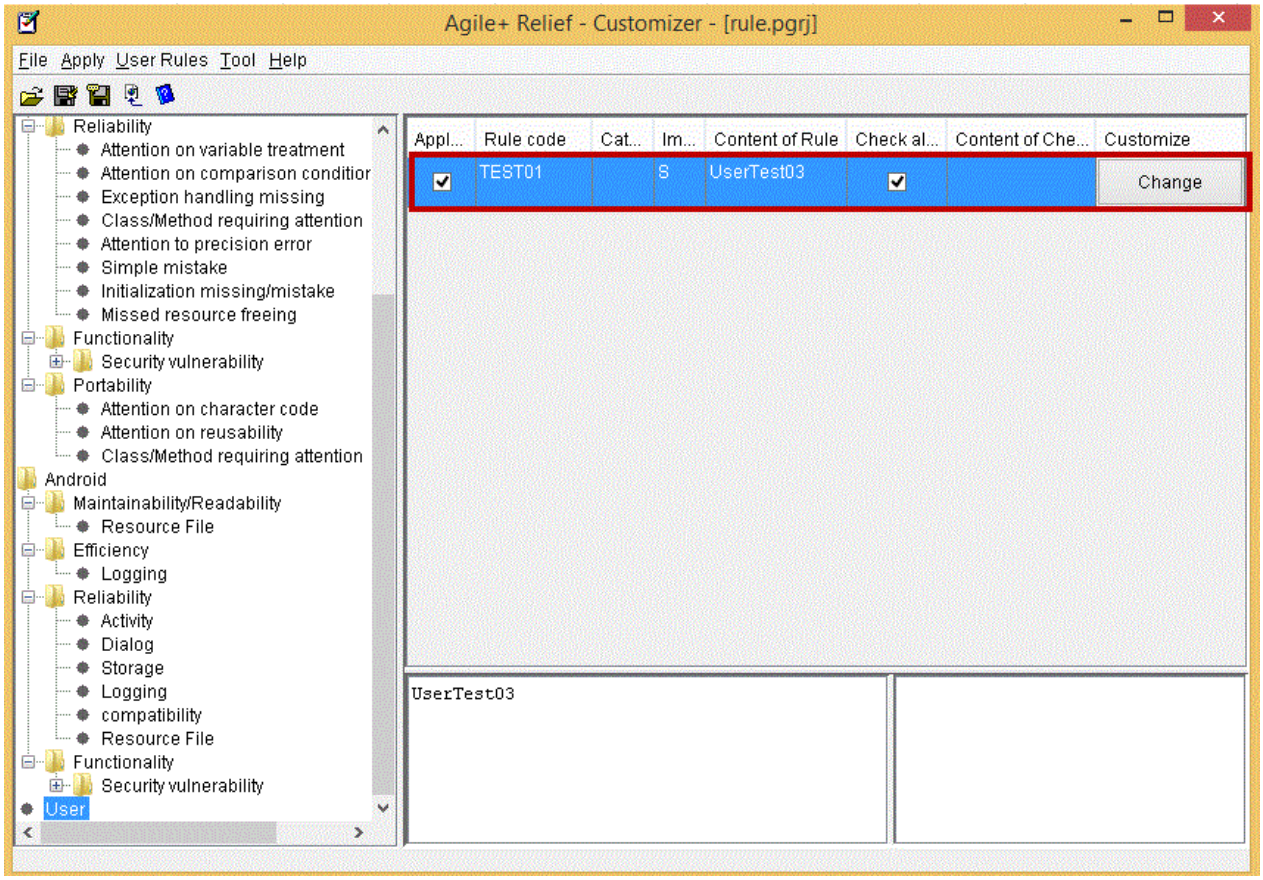


6. Register a user rule

Click the [Add] button at the lower-right of the [User Rule Configuration] dialog to register the created user rule.



The created user rule is added to the rule list when returning to the main screen.

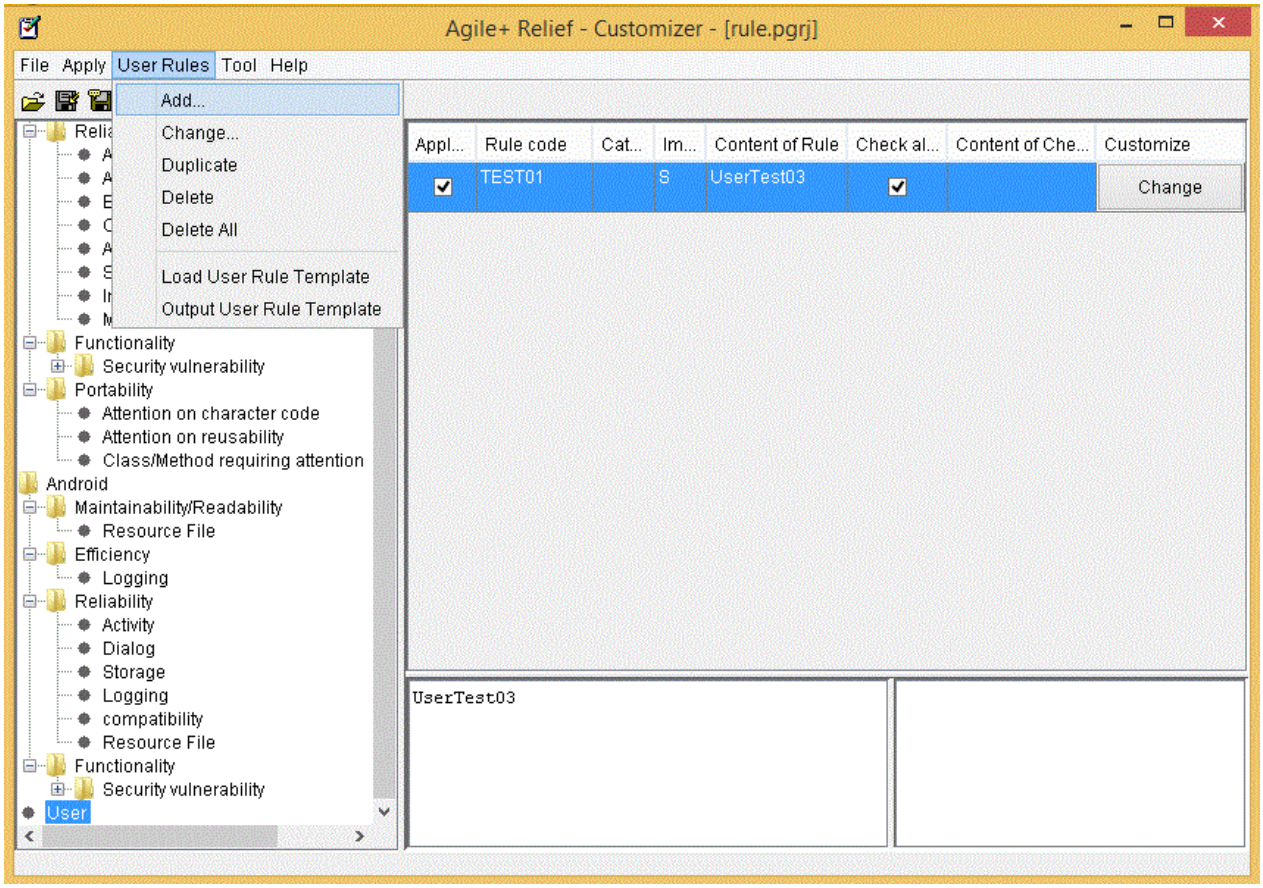


3.5.9.2 Creation Example 2

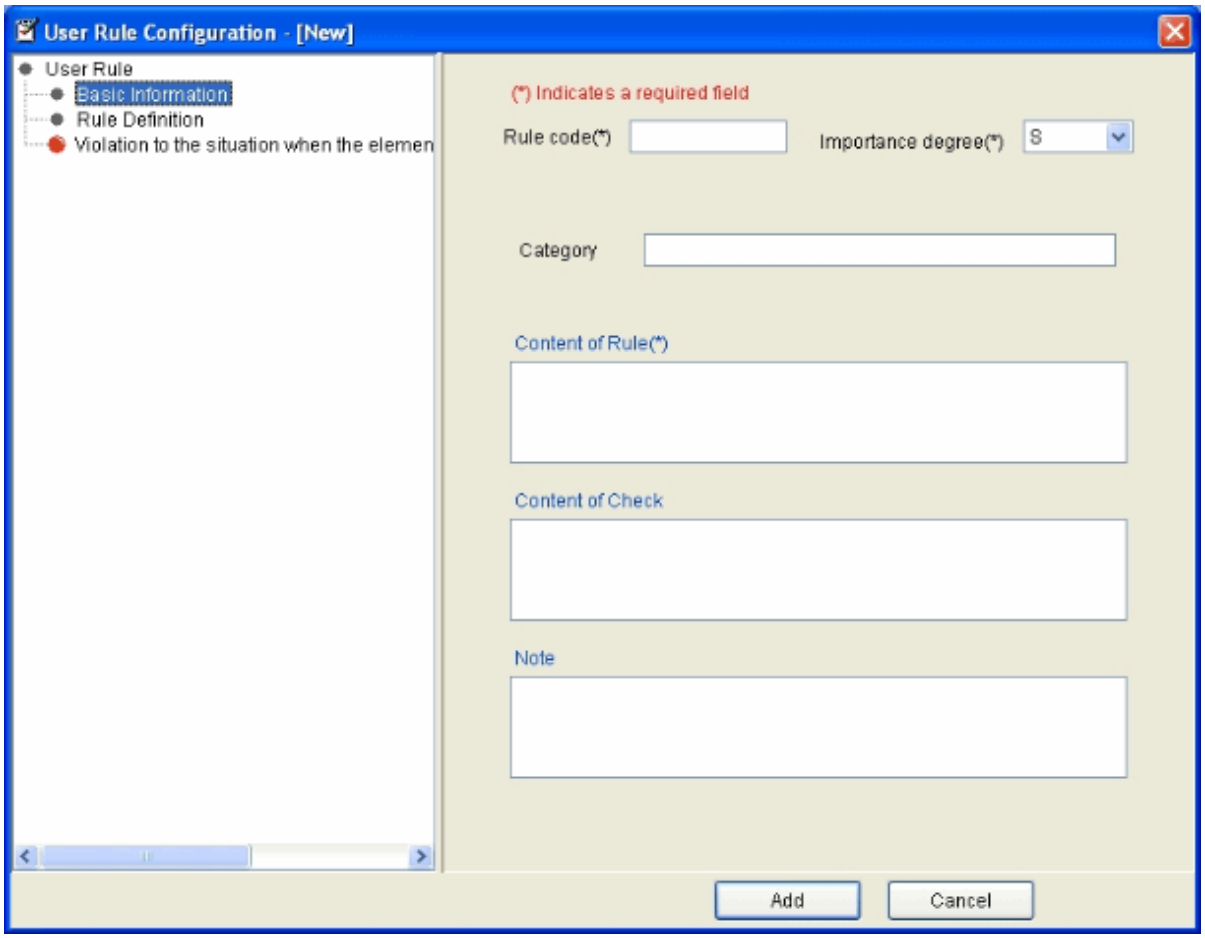
Do not invoke the public method defined its own class (the class to which the method belongs) in method.

1. Start [User Rule Configuration] dialog

   Select [User rules] menu > [Add] of Customizer.

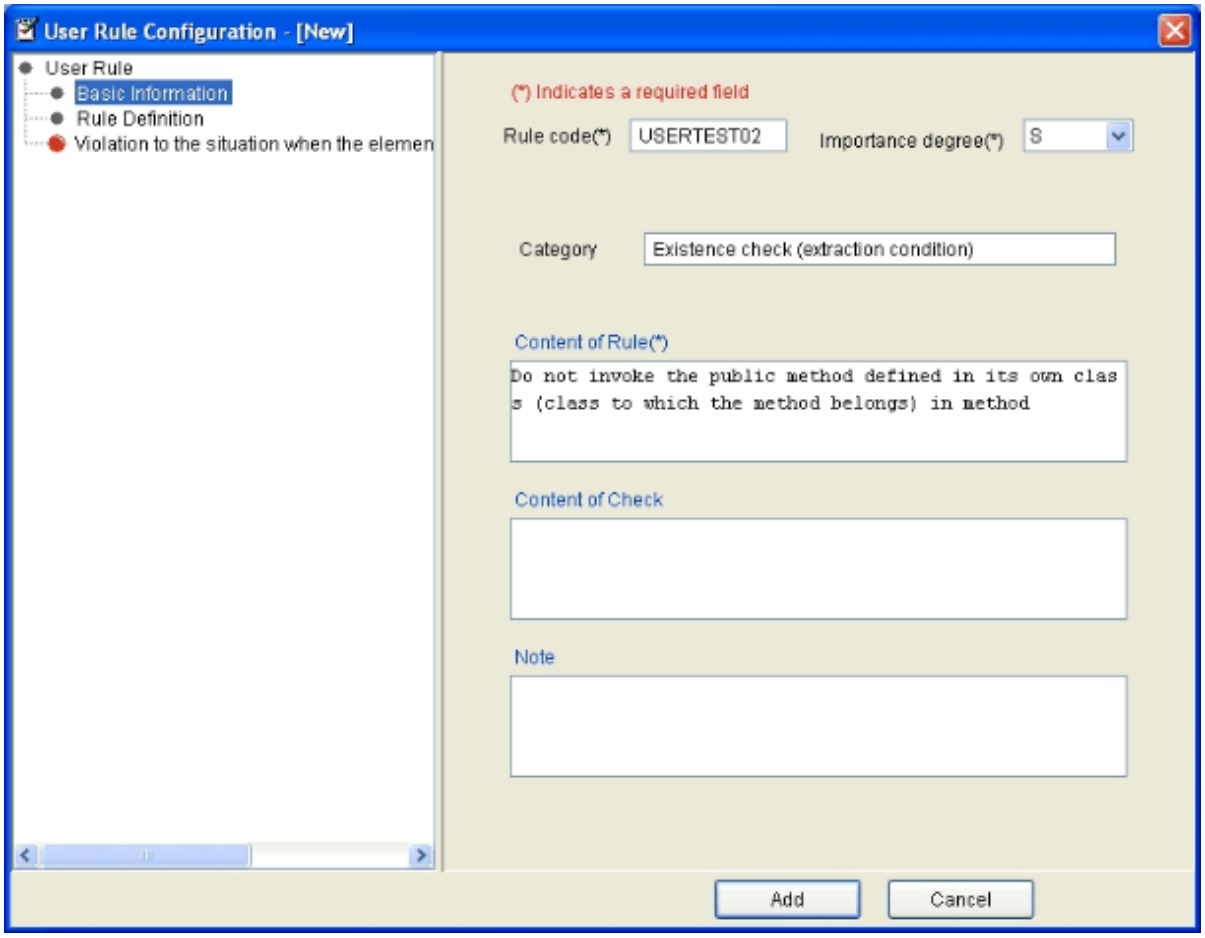


The [User Rule Configuration] dialog is displayed.



2. Basic information

Set the basic information of user rule in the [User Rule Configuration] dialog.

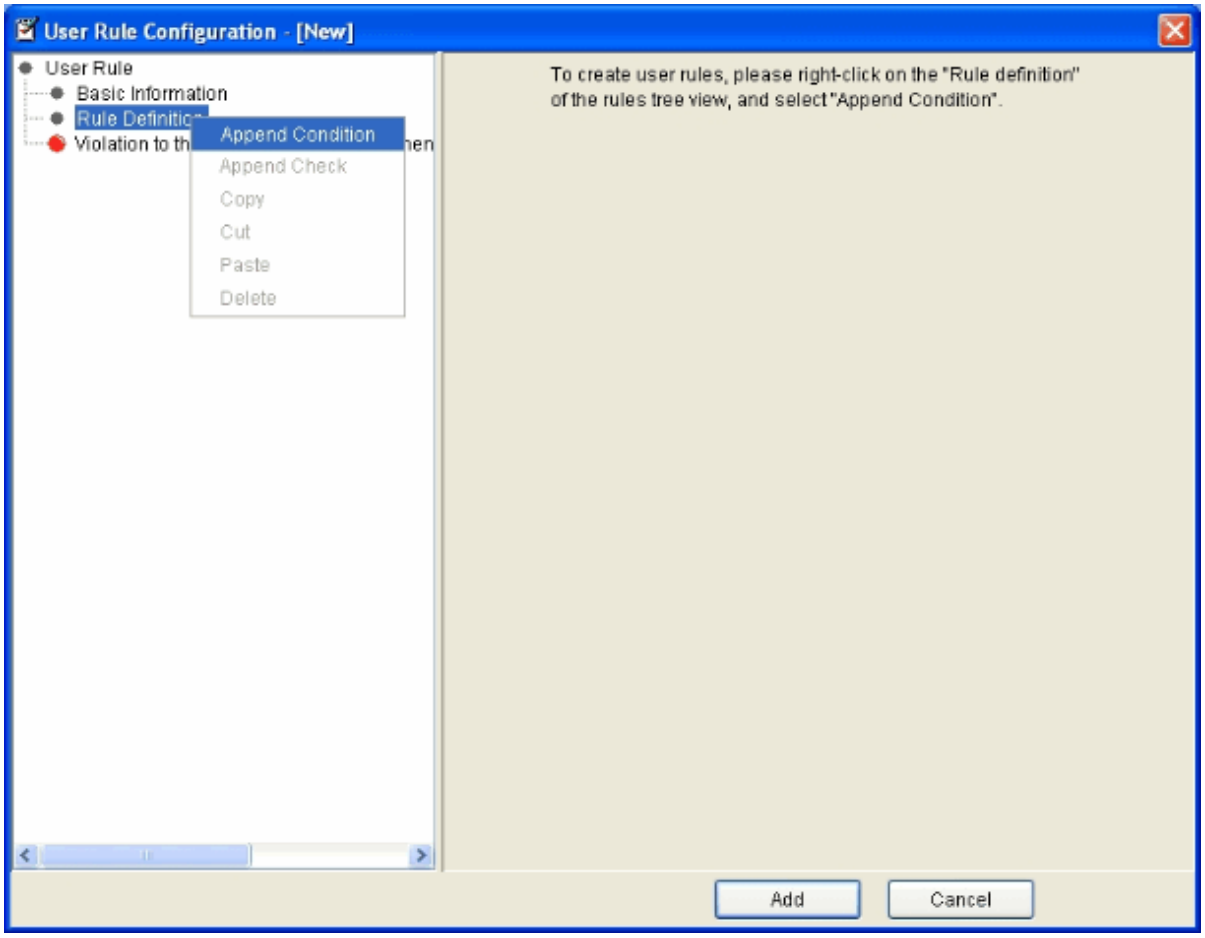


Set the following items as basic information. (In this example, only the mandatory items marked with * are set.)

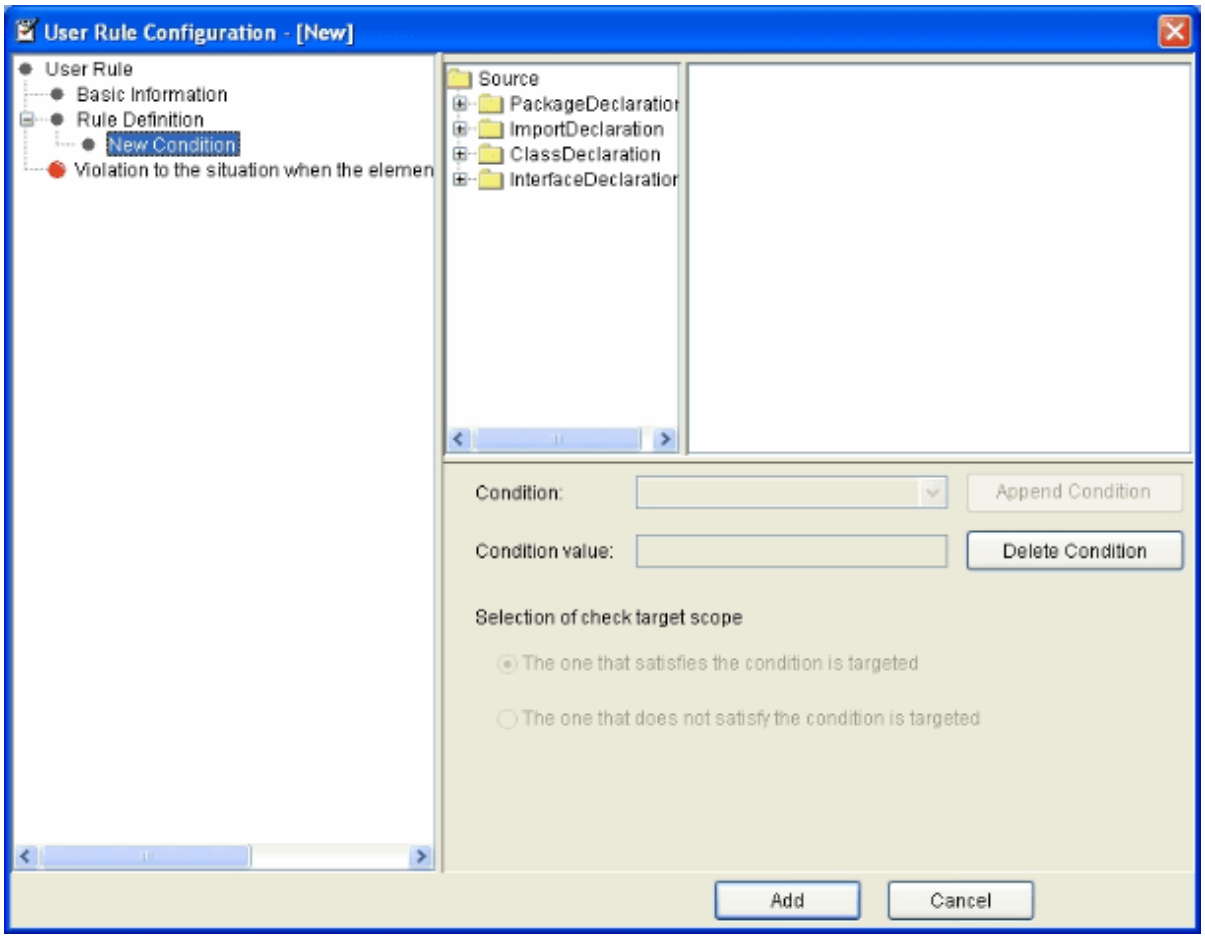
Configuration Items	Value
Rule code	USERTEST02
Importance degree	S
Category	Existence check (extraction condition)
Rule content	Do not invoke the public method defined in its own class (class to which the method belongs) in method

3. Append a condition

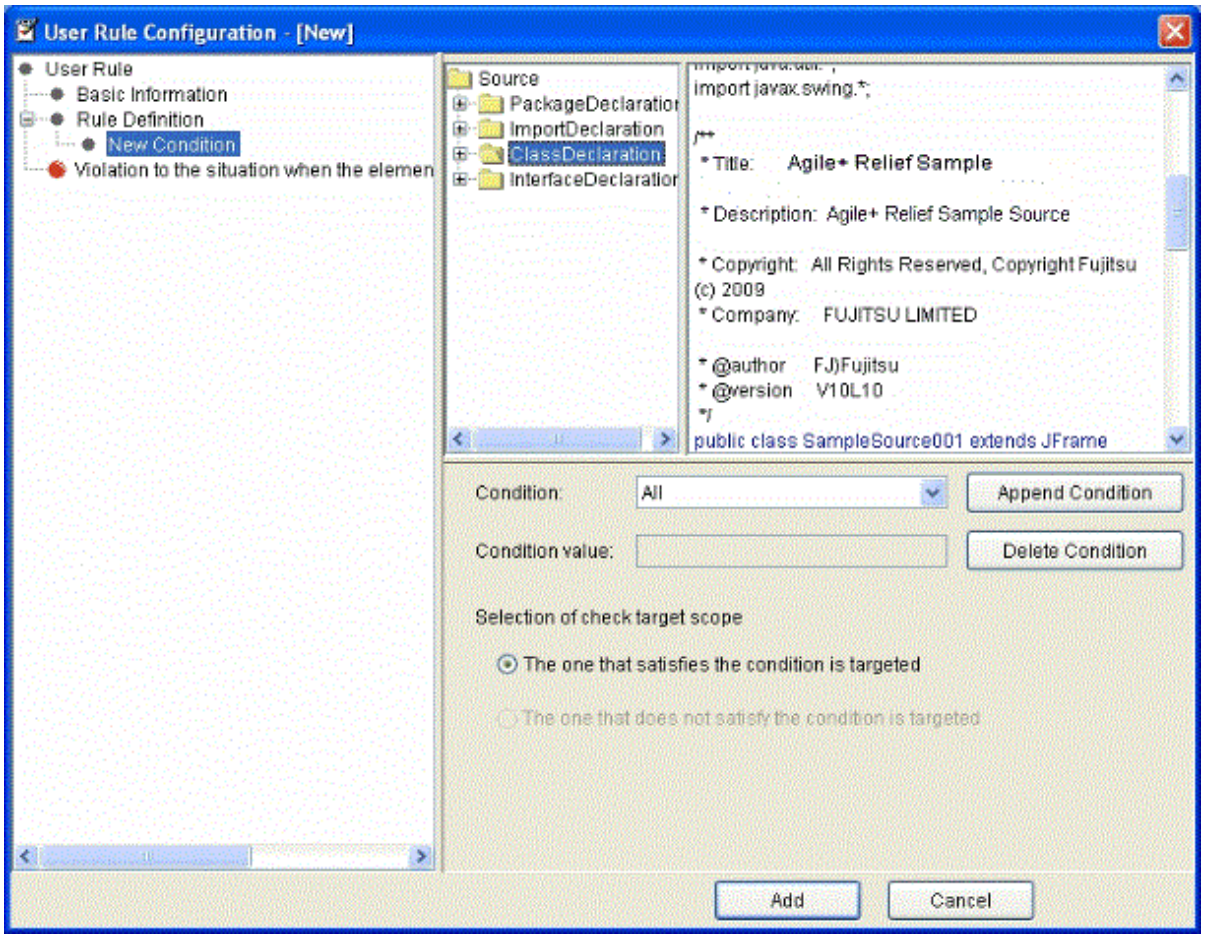
Right-click Rule Definition in the selected status and select [Append condition] from the pop-up menu displayed by right-click.



"New condition" is added under "Rule Definition", and the check element setting view is displayed on the right of the dialog.



In the condition element setting view, set the condition to limit the Java source that becomes the target.



Set each item of the condition element setting view according to the following contents.

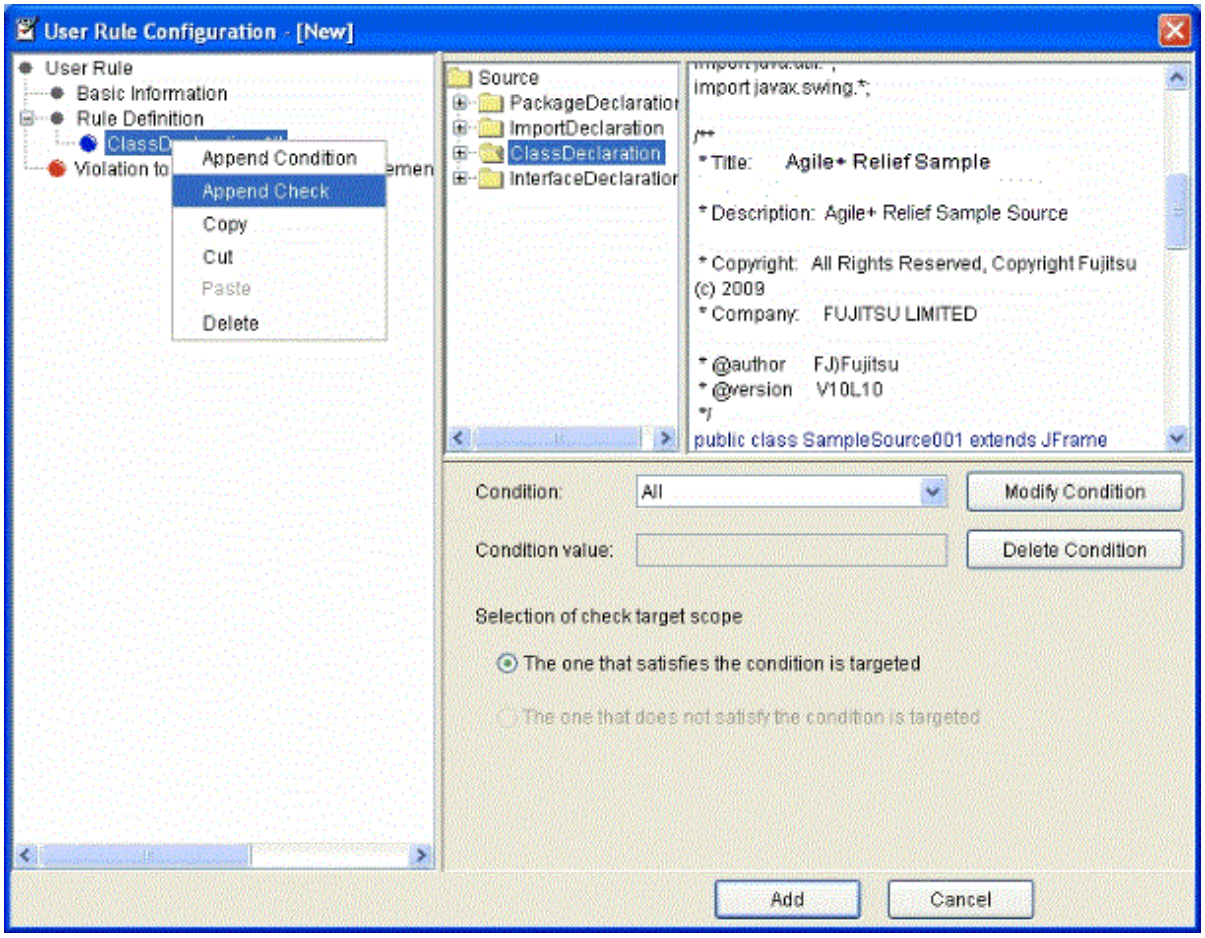
Configuration Items	Value
Java element	ClassDeclaration
Condition	All
Condition value	-
Selection of the check target scope	The one that satisfies the condition is targeted

Click the [Append Condition] button to register the condition part that has been set.

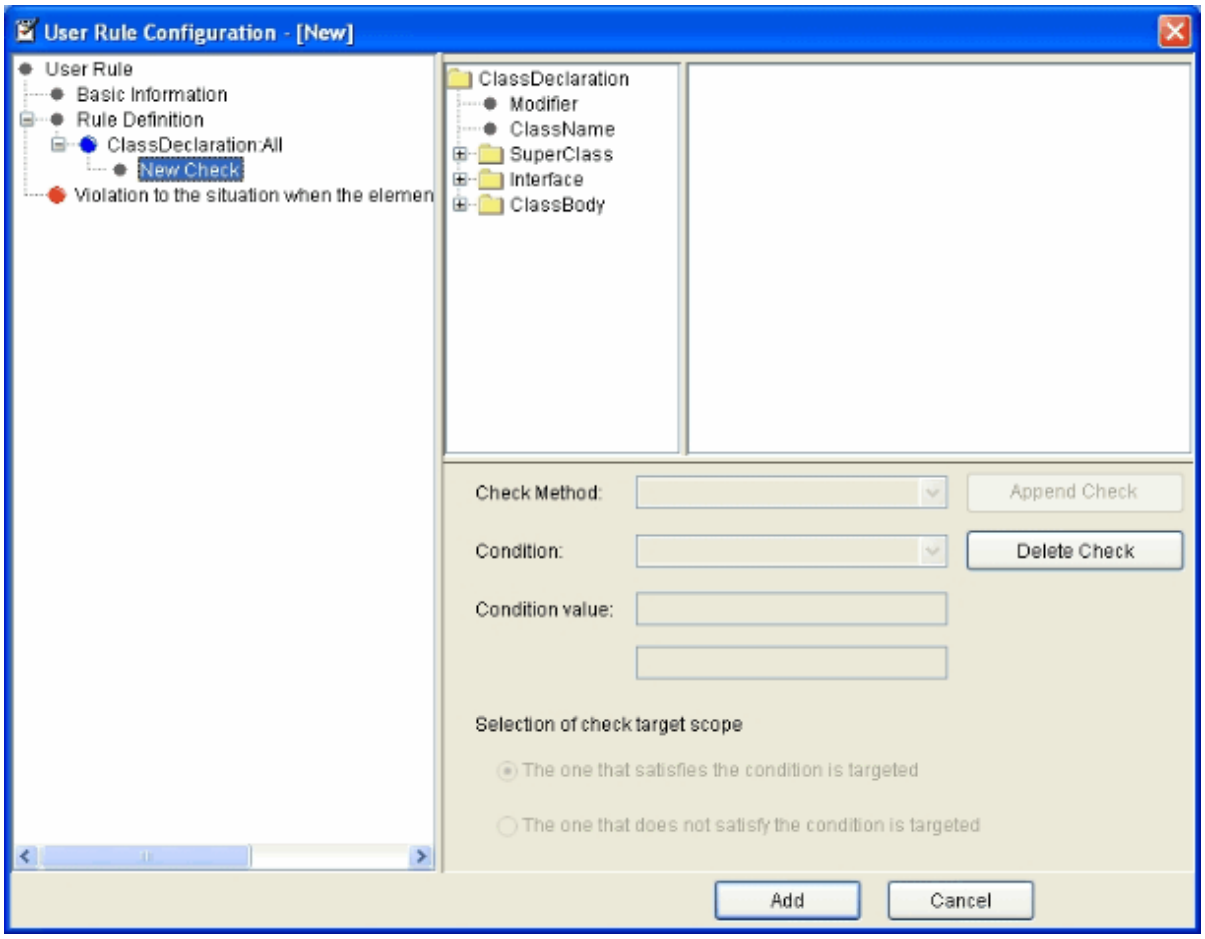
The setting of "ClassDeclaration:All" as the condition element is added.

4. Append a check

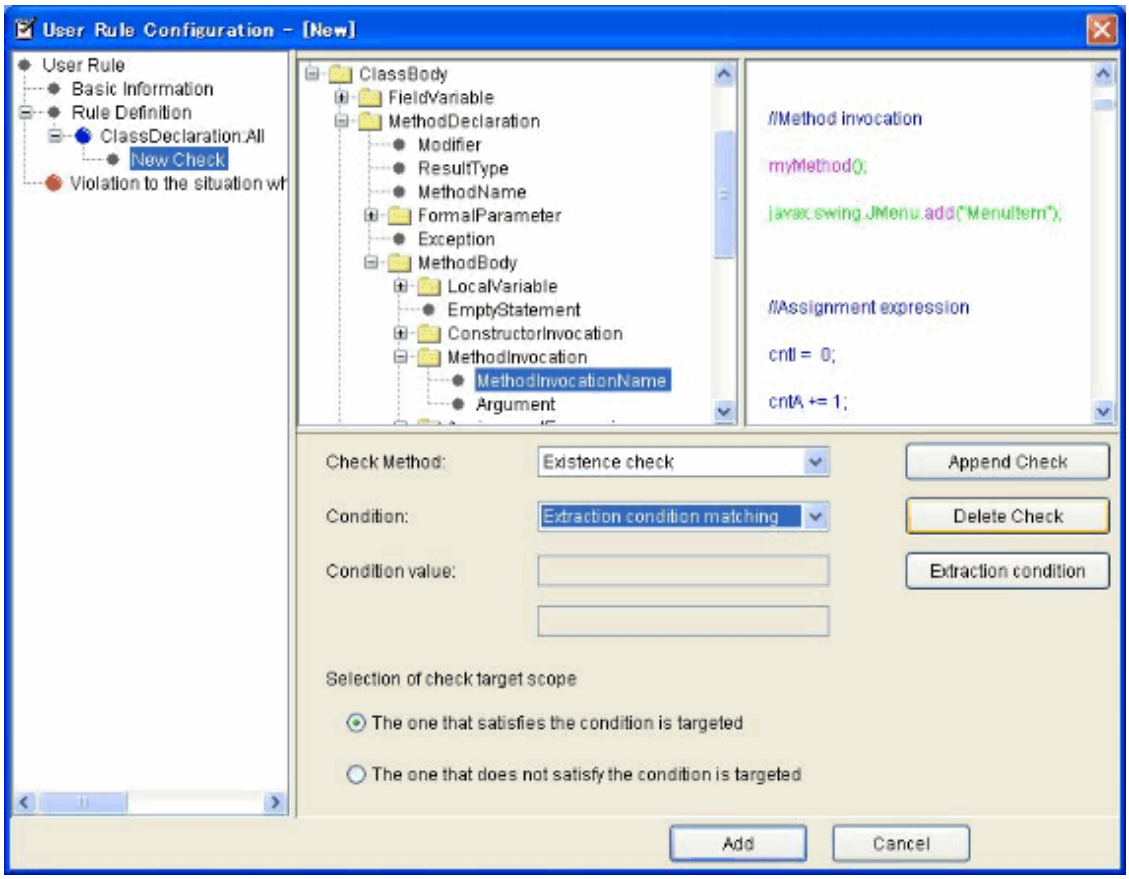
Select [Append Check] from the pop-up menu of "ClassDeclaration:All" condition element that is added in the rule tree view.



"New check" is added under the condition element "ClassDeclaration:All", and the check element configuration view is displayed on the right of the dialog.



Set the following check elements in the check element configuration view.

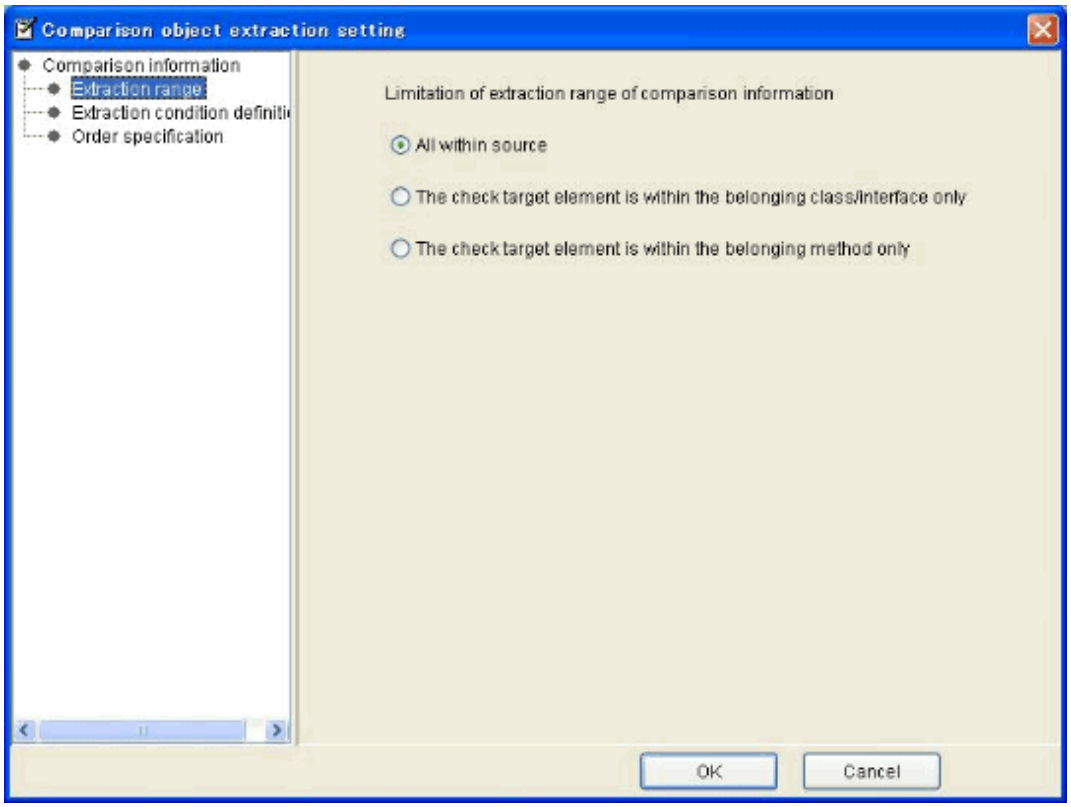


List of configuration items

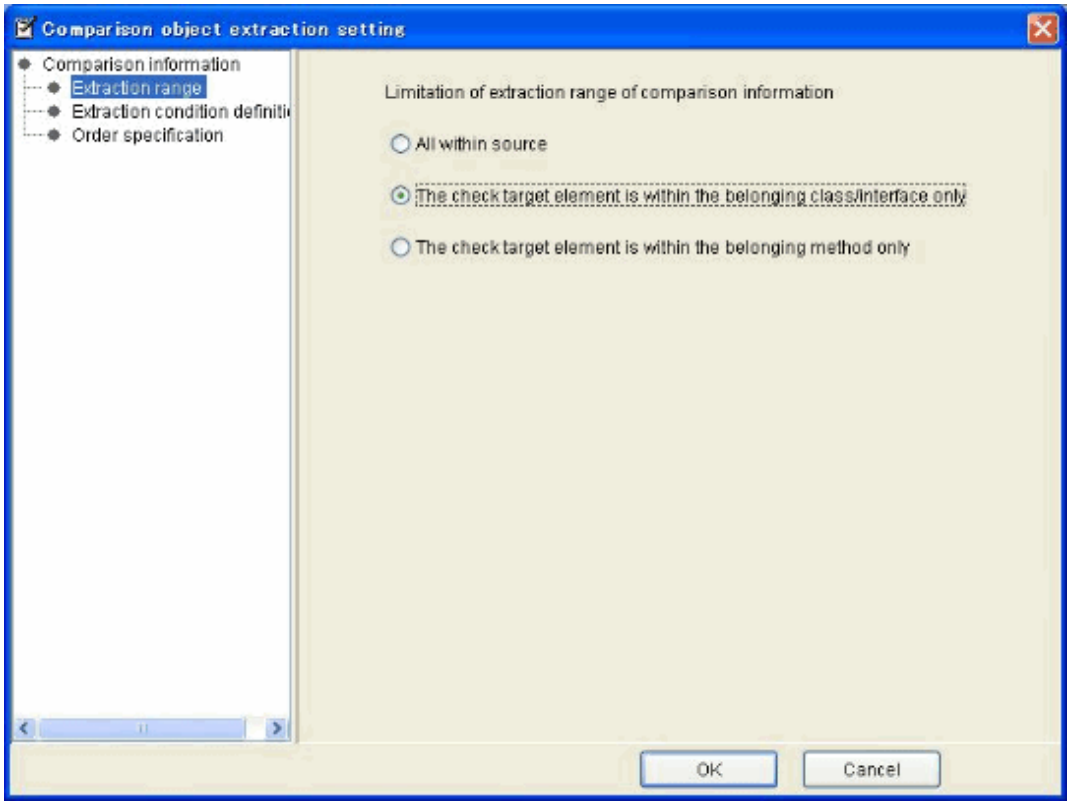
Configuration Items	Value
Java element	MethodInvocationName
Check content	Existence check
Condition	Extraction condition matching
Condition value	-
Selection of the check target scope	The one that satisfies the condition is targeted

The [Extraction condition] option button is displayed.

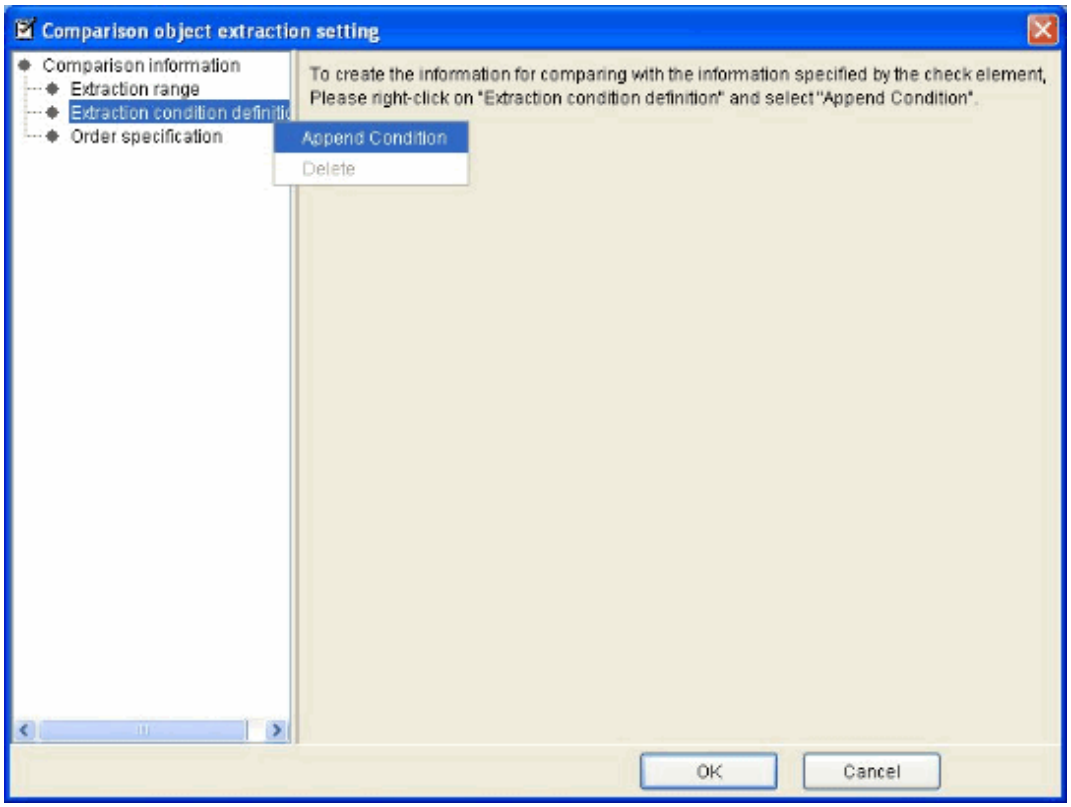
Click the [Extraction condition] option button, the Comparison object extraction setting dialog is displayed.



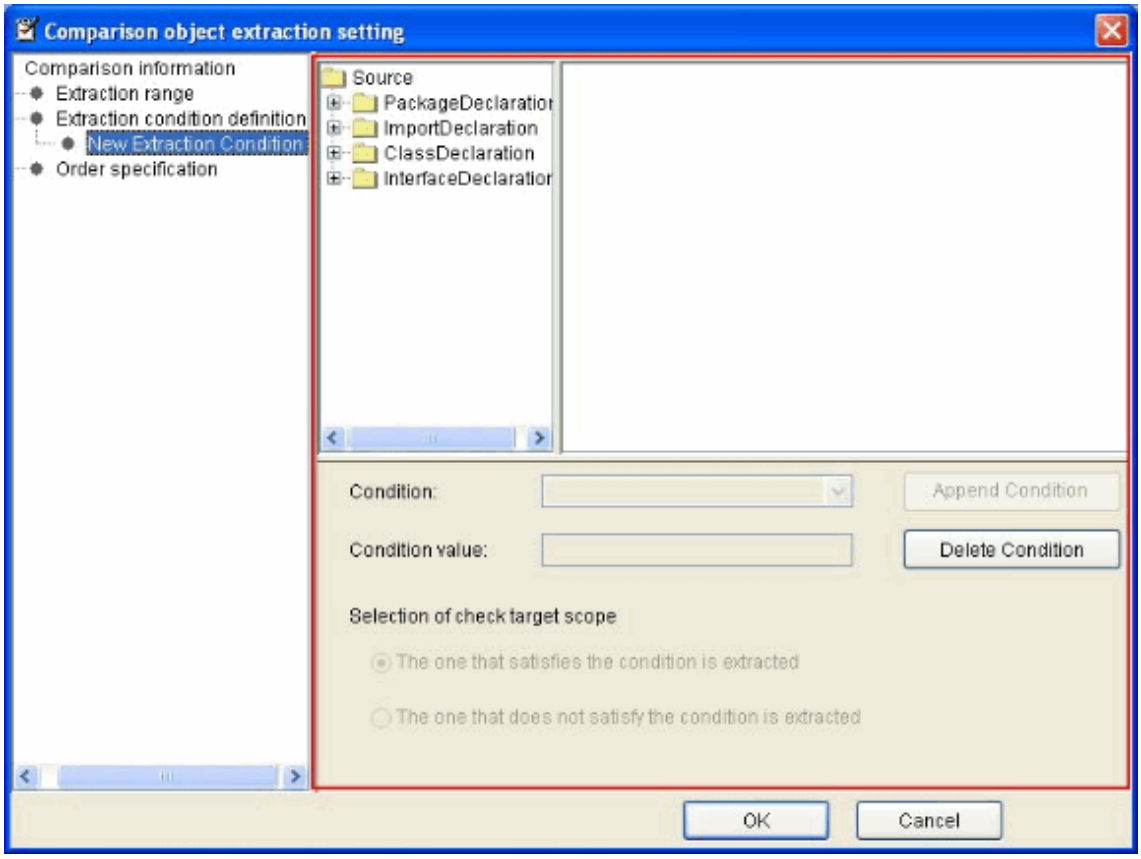
Select "The check target element is within the belonging class/interface only" in "Extraction range" of the Comparison object extraction setting dialog.



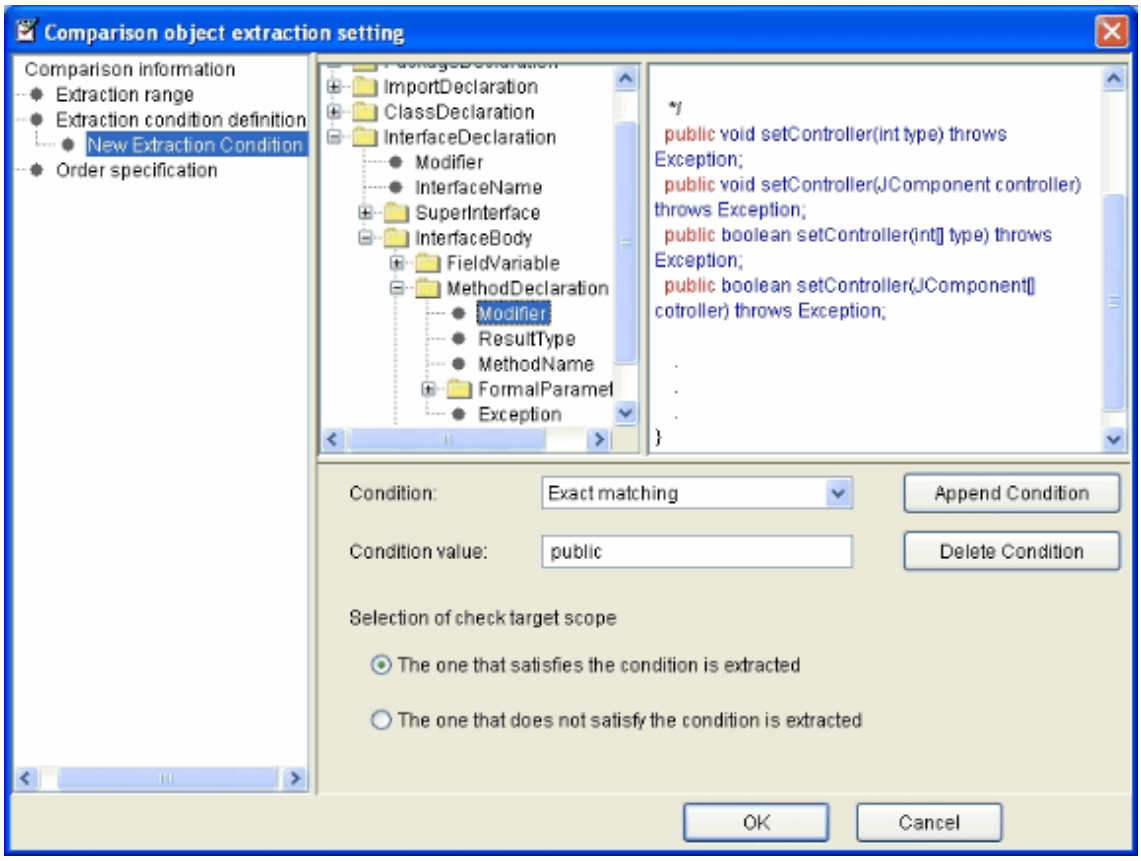
Right-click [Extraction condition definition] of the comparison object extraction setting dialog and select [Append Condition].



The extraction element setting view is displayed.



Perform the following settings in the extraction element setting view.



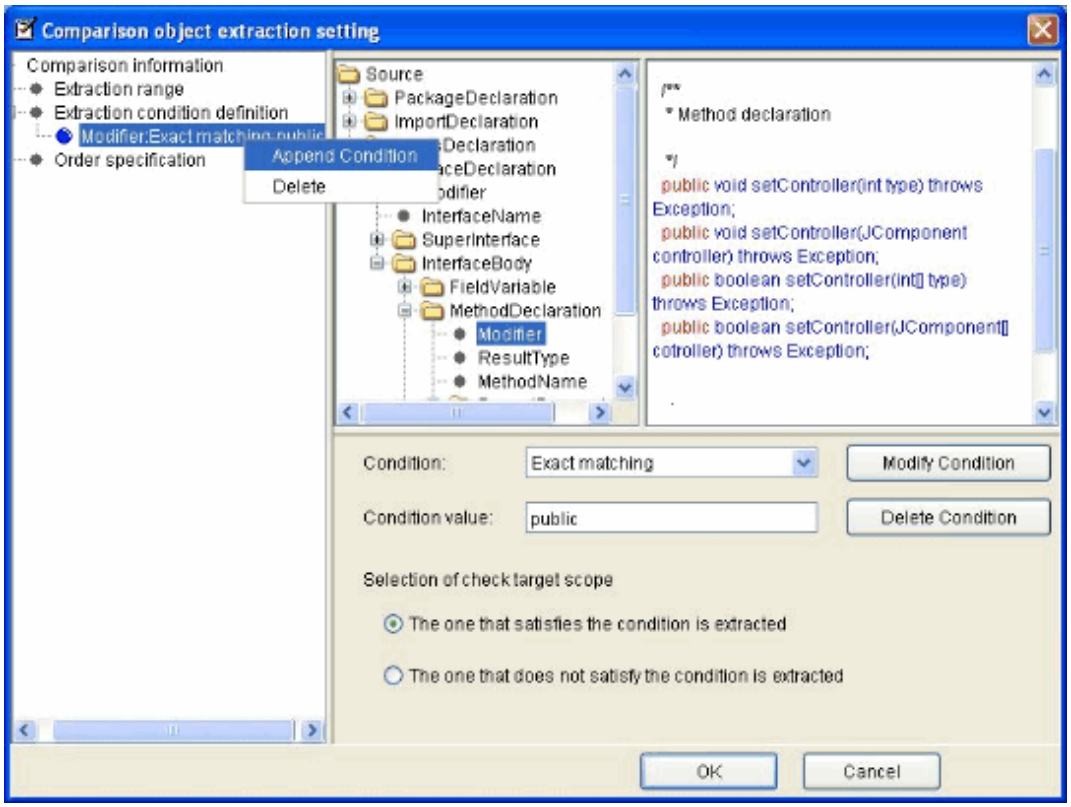
Set each item of the extraction element setting view according to the following contents.

Configuration Items	Value
Java element	Modifier
Condition	Exact matching
Condition value	public
Selection of the check target scope	The one that satisfies the condition is targeted

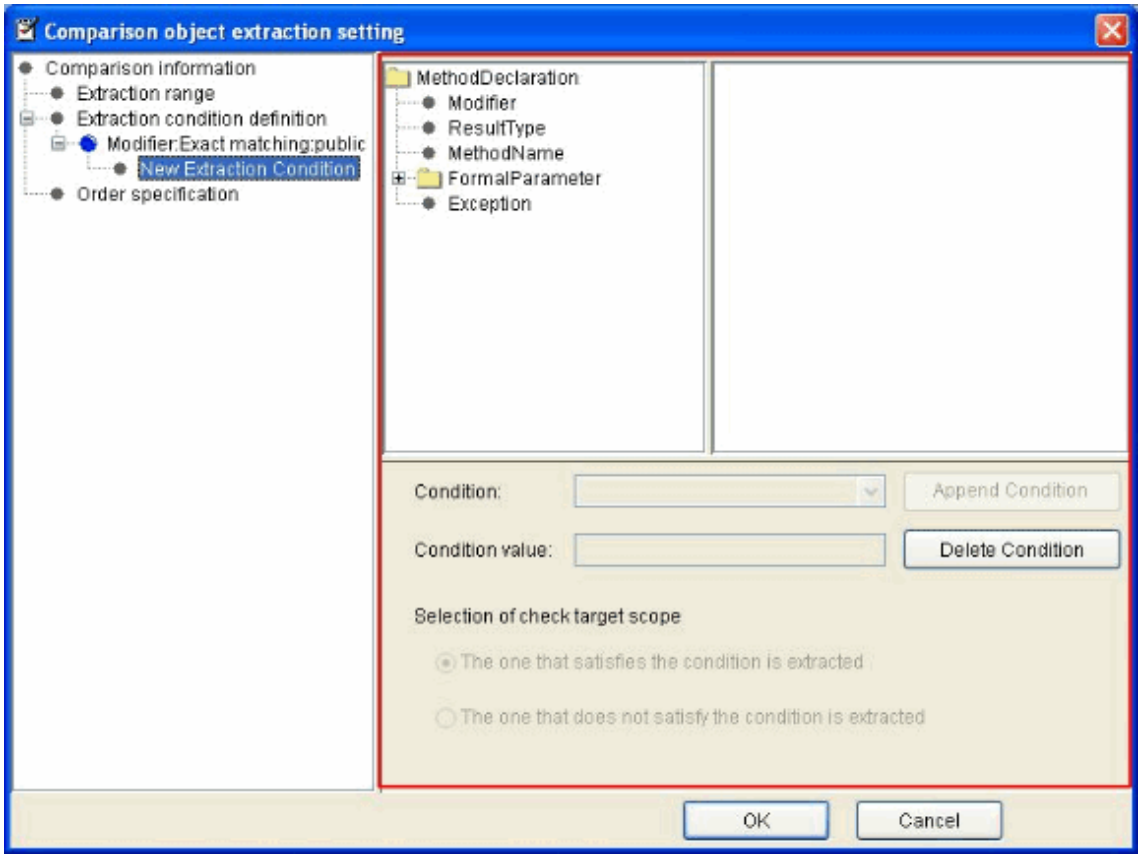
Select "Append Condition" of the check element setting view.

Then, define the extraction condition.

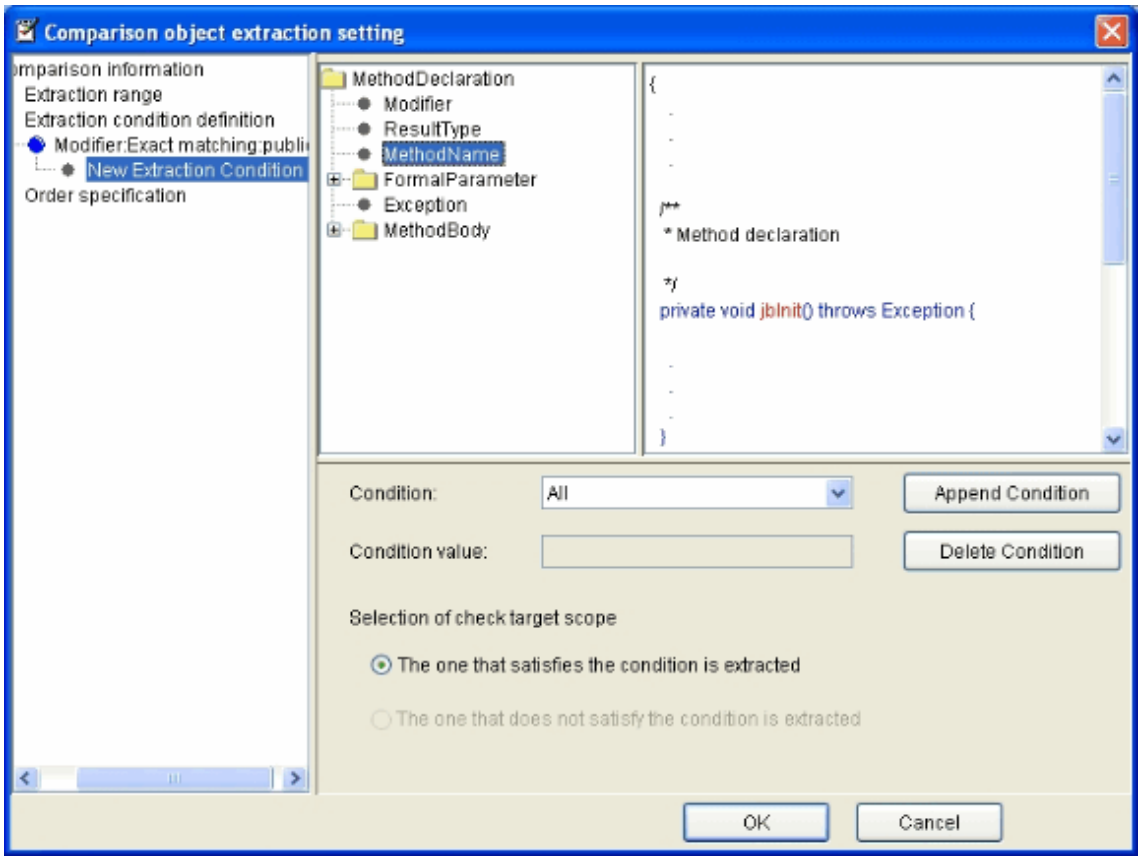
Right-click "Modifier:Exact matching:pubic" added in "Extraction condition definition" and select "Append Condition".



The extraction element setting view is displayed.



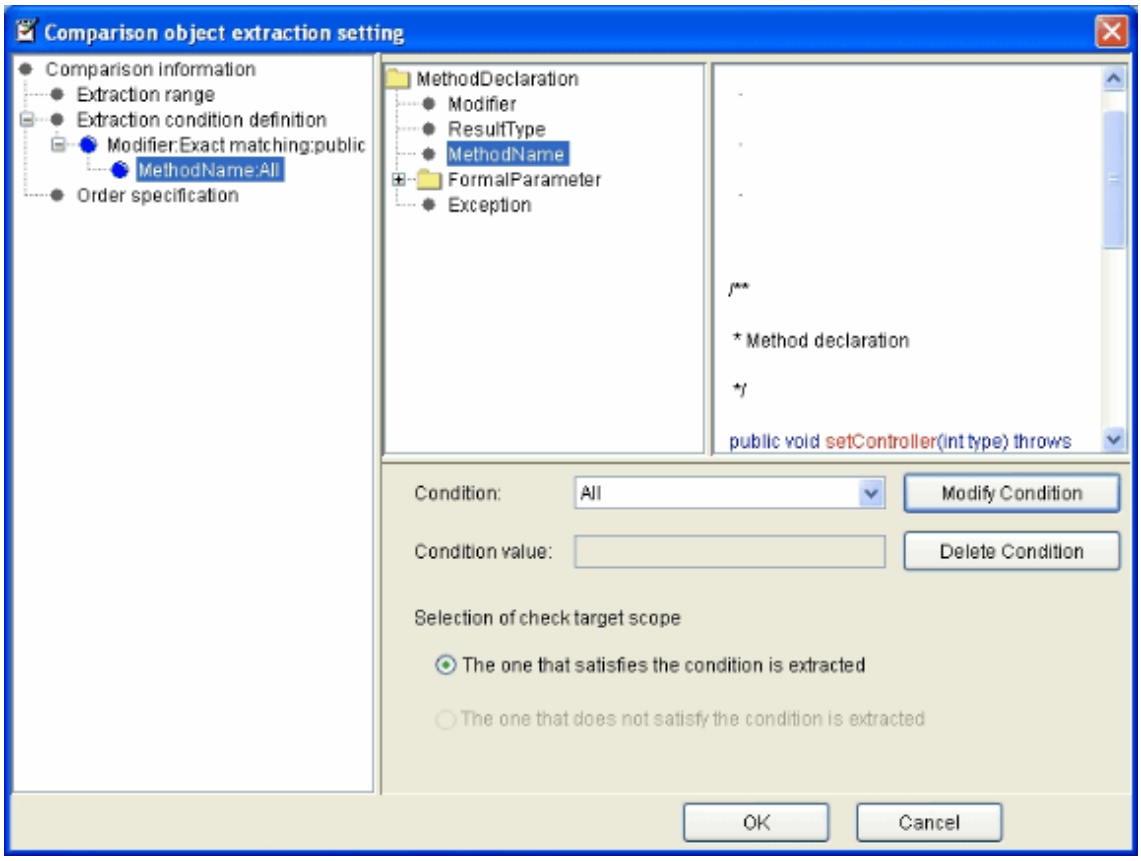
Perform the following settings in the extraction element setting view.



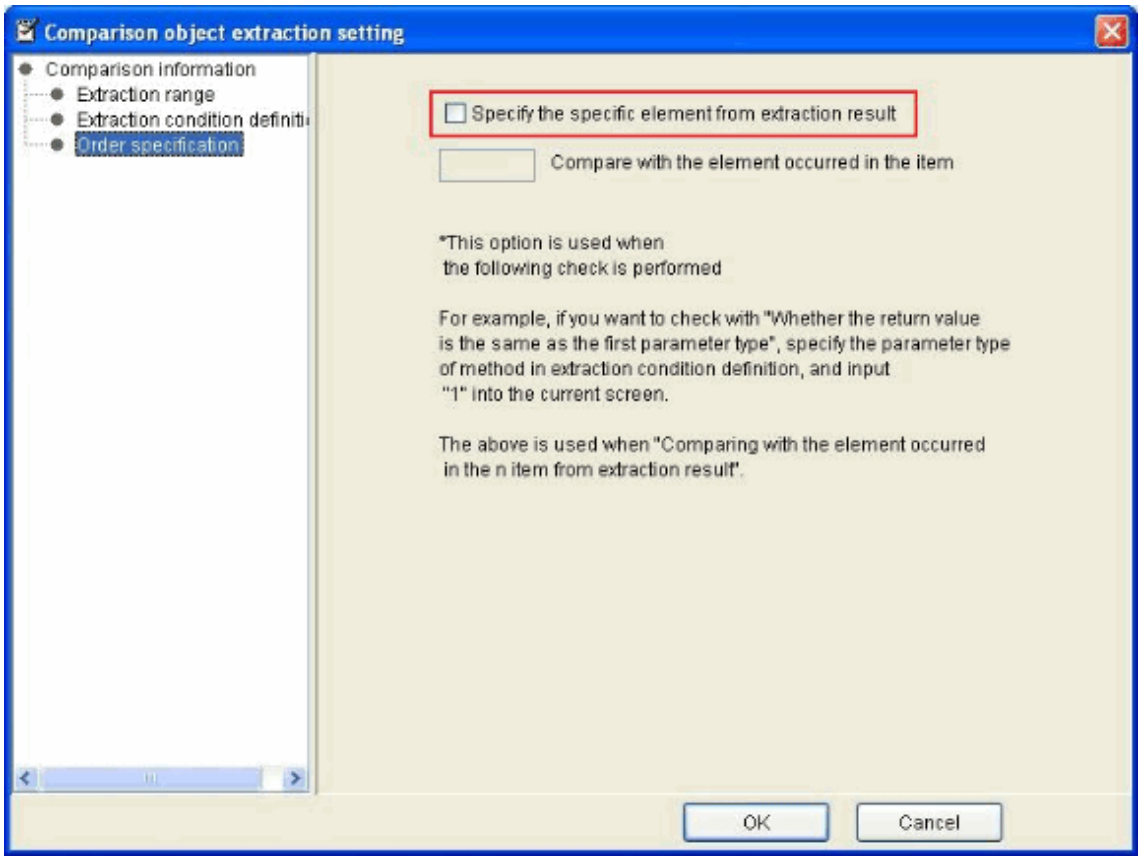
Set each item of the check element configuration view according to the following contents.

Configuration Items	Value
Java element	MethodNname (of MethodDeclaration)
Condition	All
Condition value	-
Selection of the check target scope	The one that satisfies the condition is extracted

Select "Append condition" of the check element configuration view.



In "Order specification" of the comparison object extraction setting dialog, do not select "Specify the specific element from extraction result".



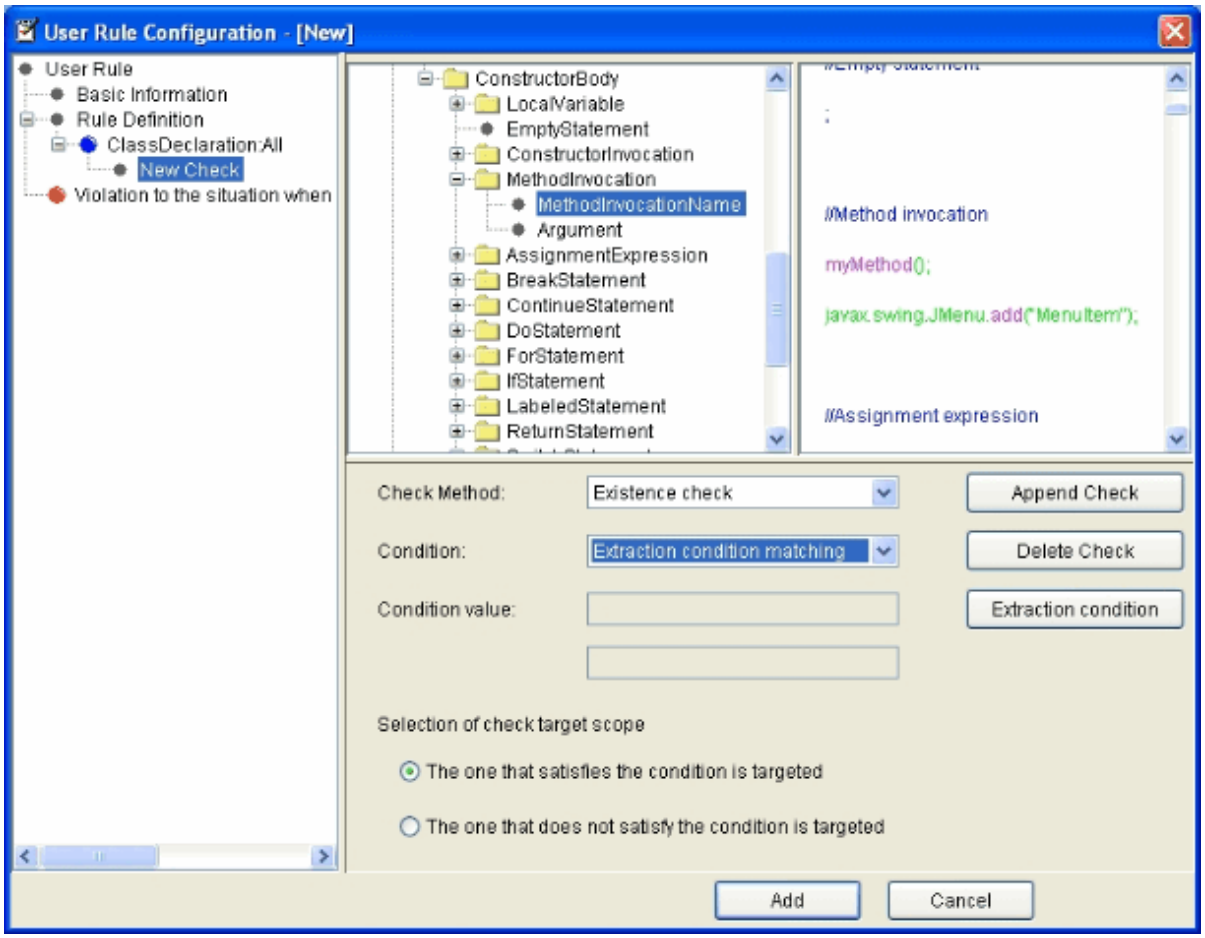
Once the settings of comparison object extraction setting dialog is complete, click the [OK] button.

(Hint) The information extracted from specification of the above extraction condition and the Java element of check target has the following meanings.

When the Extraction range is "The check target element is within the belonging class/interface only", it will turn to perform with in the class/interface which is "check target element = MethodInvocation".

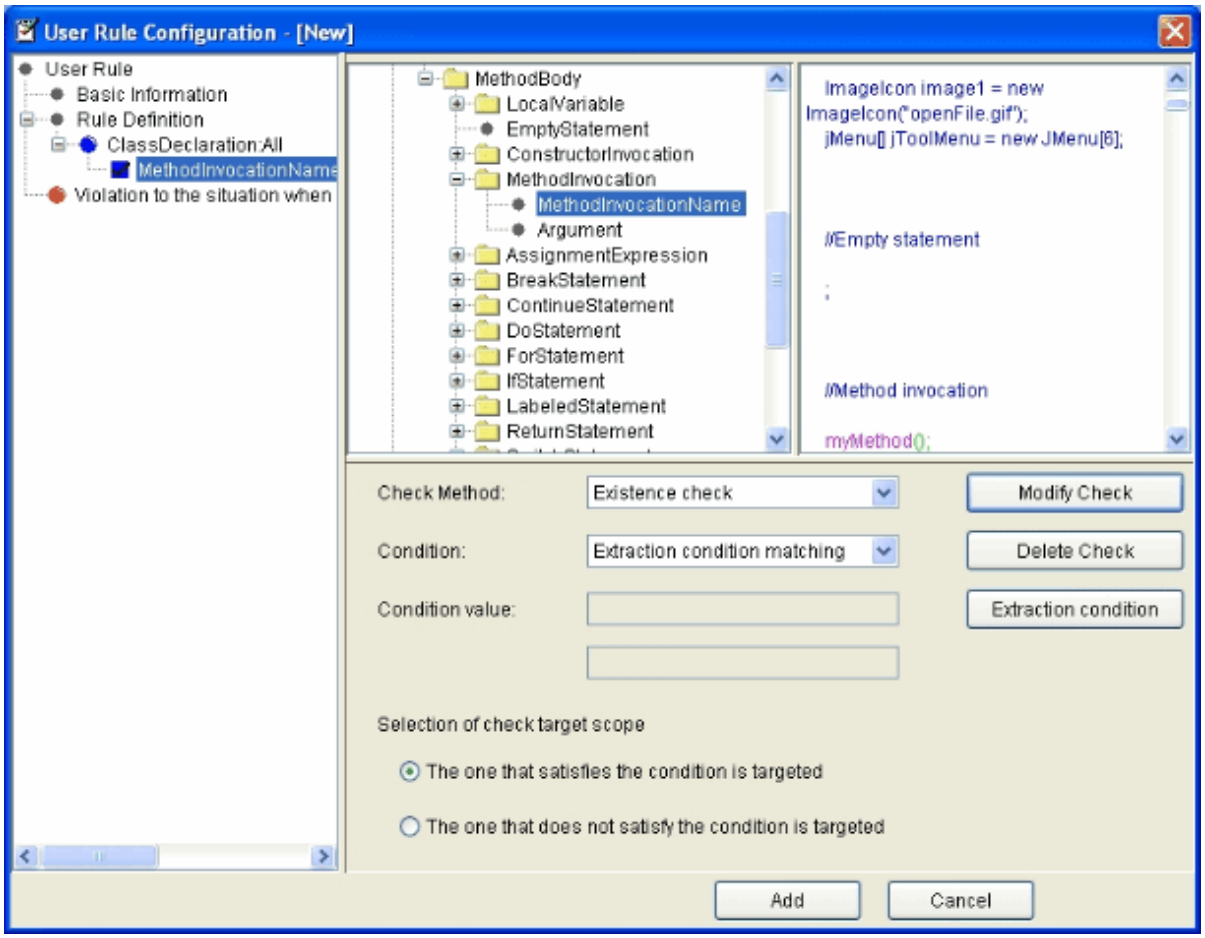
Extracting any information from the Extraction range, it means extracting all the method names with qualifier (of method declaration) being "public" as information.

Return to the [User Rule Configuration] dialog.



The setting of check element configuration view is complete.

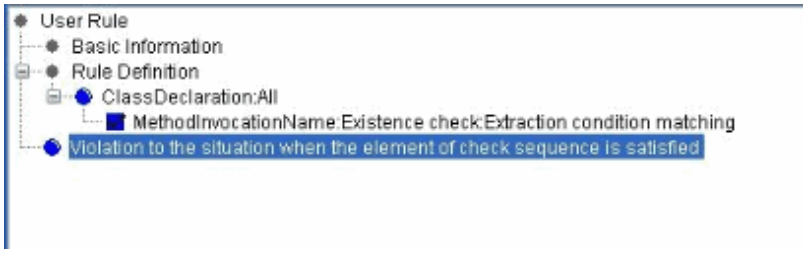
Click the [Append Check] button to register the check element.



The check part "whether to invoke the public method defined in its own class (to which the method belongs) in method" can be created.

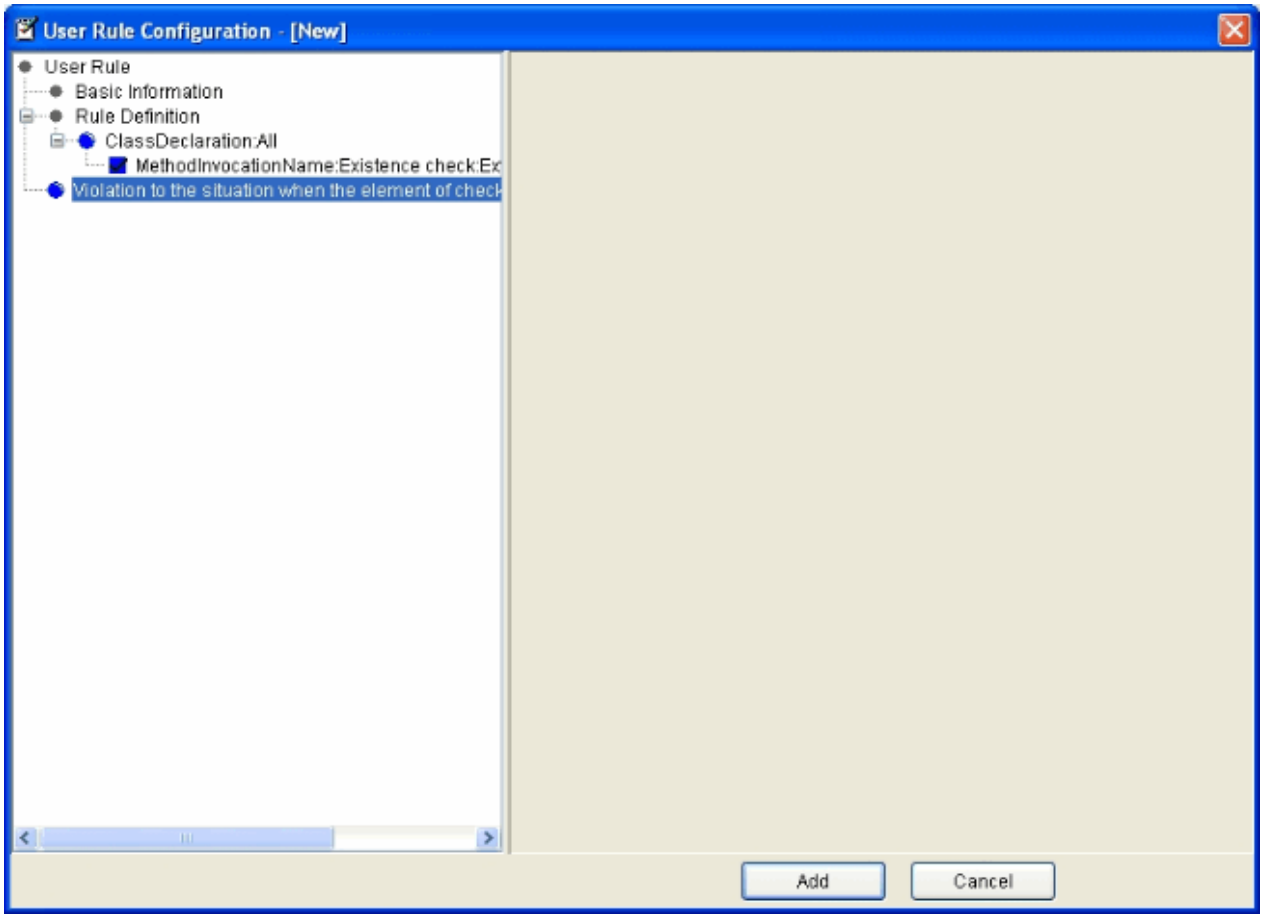
5. Judge rule violation

Click rule violation judgment in the violation to the situation when invoking the public method defined in its own class (to which the method belongs) in method, and set "Violation to the situation when the element of check sequence is satisfied".

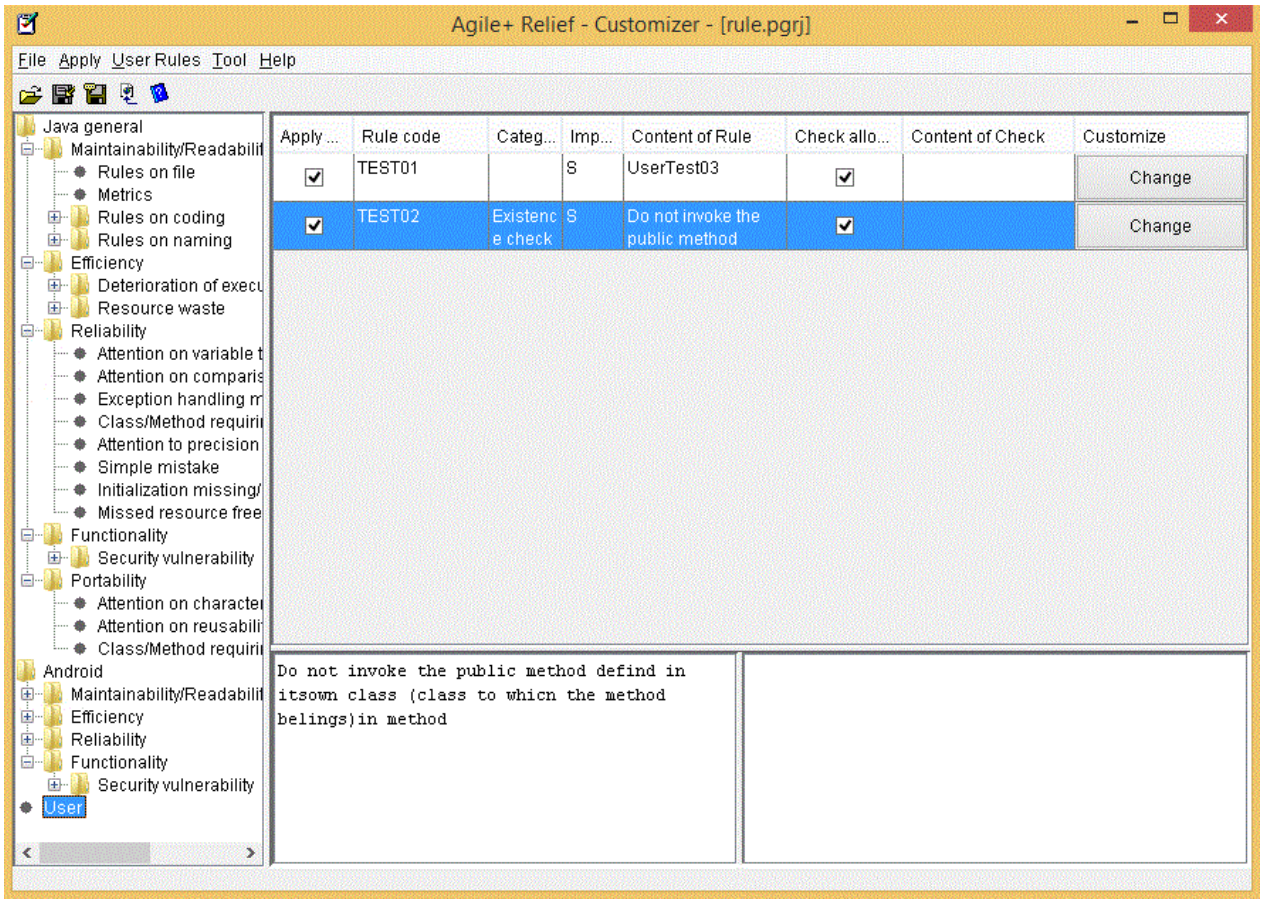


6. Add

Finally, select the [Add] button at the lower-right of [User Rule Configuration] dialog.



Return to the main page, and the created user rule is added.



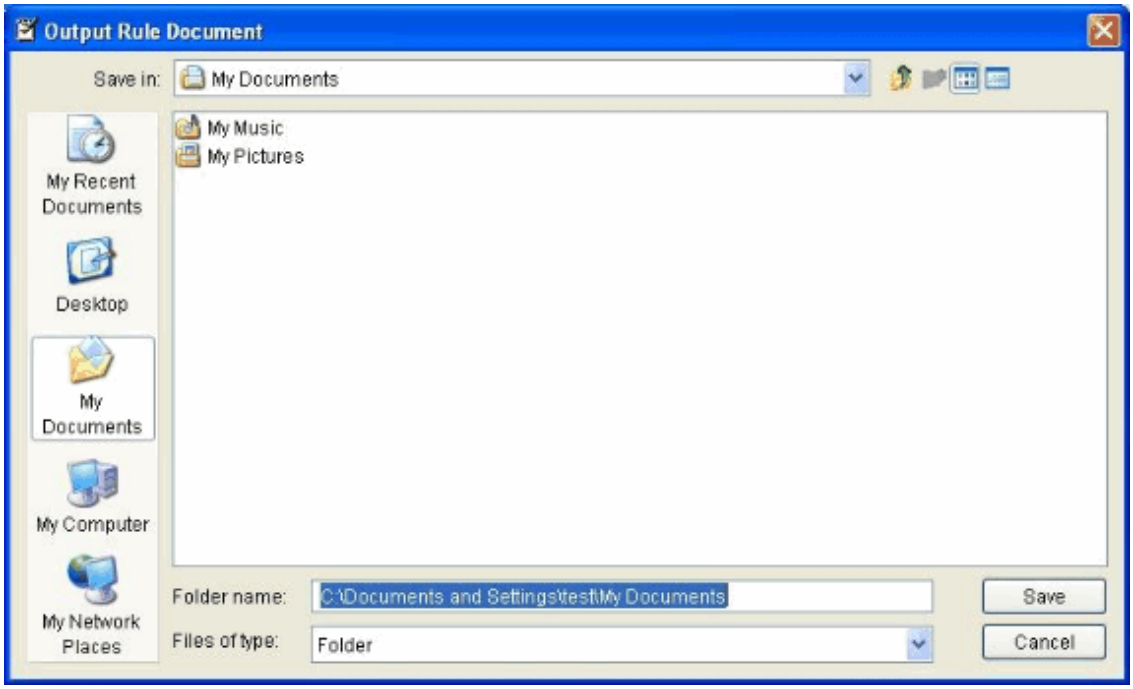
3.6 Outputting Rule Document

Output the coding rules (check rule definition file) as a file in html format.

<Operations>

1. Select [Tool] > [Output Rule Document].

2. Select the output folder of rule document in the folder selection dialog box, and click the [Save] button.

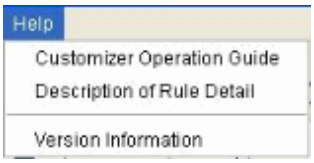


3. Click the [Save] button, and the generation of rule document is started.
   To cancel the processing, please click the [Cancel] button.

To browse the rule document, please open index.htm in the output folder in the Browser.

3.7 Help

This section describes how to use help.



- Customizer operation procedure

  Show "Agile+ Relief J Customizer Operation Guide".

- Description of Rule Detail

  Show "Agile+ Relief J Rule Detail Manual".

- Version information

  Show the version information of this product.

Chapter 4 Cautions

This chapter describes the cautions of using the customizer.

4.1 All

- During check application, rules are required for the setting of customizer.

  The applicable rules are as follows.

  <Applicable Rules>

  pgj10097, pgj10025

- Please do not use other applications to open and update the check rule definition file used in customization.
- Please do not start the customizer or update the same check rule definition file for multiple times. (The post saving becomes valid).
- As the operating environment of customizer, please use the resolution of display that is greater than 1024 x 768.
- According to the customization of the rule that allows numeric settings, when the value is greater than 2, 147, 483, or 647, or a minus value is set, the following message will be displayed.

  "The non-numerical information is specified. This project does not allow the input of information except numbers. Please confirm the input information."

- About the check rule definition file

  The maximum number of user rules in the check rule definition file is 500. The check rule definition file over this maximum number cannot be guaranteed.

- Please set required access privilege for the check rule definition file and document output folder.
- The environment variable JAVA_HOME must be registered. Please register the path of JDK being used to JAVA_HOME.

4.2 When using Interstage Studio

- If the applet of the Interstage Studio is used in Java development tool, customization of the following rules is required.

  pgj10090 : The available character type of class name is customized to _ and $.

  pgj10109, pgj10116, pgj10125 : The available character type of the variable is customized to _.

  pgj10134 : The available character type of the method name is customized to _ and $.

- When the applet of the "Interstage Studio" is used, please do not apply the rule pgj10021.