

# **FUJITSU Software**

## **Agile+ Relief J V1.1.1**

A horizontal band featuring a red abstract graphic with flowing, overlapping lines and bright light spots, creating a sense of motion and energy.

# Improvement Reference

B1WD-3619-01ENZ0(00)  
January 2023

# Preface

---

The Agile+ Relief J is a product supports the quality improvement for the coding project in the development process. Through static parsing of the programs written by Java language, the problems that are ignored visually can be detected. Thus, the product quality can be improved and the development cost can be reduced.

## About This Manual

This manual explains the improvements and changes that have been made from "PGRelief J 2009 autumn" to "Agile+ Relief J V1.1.1".

## 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.

All rights reserved, Copyright(C) 2009-2023 FUJITSU LIMITED.

# Contents

---

Chapter 1 Improvement for PGRelief J 2010.....	1
1.1 Solutions for version upgrade of Java development environment.....	1
1.2 Improvements of code check operation in Eclipse.....	1
1.3 Addition of cooperation tool.....	1
1.3.1 Solutions for version upgrade of FindBugs.....	1
1.3.2 Addition of PMD cooperation.....	1
1.4 Improvements of Pgr items.....	1
1.4.1 Improvement of pgj10034.....	1
1.4.2 Improvement of pgj10056.....	1
1.4.3 Improvement of pgj10212.....	2
1.4.4 Improvement of pgj10221.....	2
1.4.5 Improvement of pgj10226.....	2
1.4.6 Improvement of pgj10235.....	2
1.4.7 Improvement of pgj10412.....	2
Chapter 2 Improvement for PGRelief J 2010 autumn.....	3
2.1 Correspondence to the latest OS/IDE environment.....	3
2.2 Improvements of Pgr items.....	3
2.2.1 Improvement of the indicate regarding security vulnerability.....	3
2.2.2 An optional constant is added to pgj10410.....	3
Chapter 3 Improvement for PGRelief J 2011.....	5
3.1 Correspondence to the latest OS/MW/IDE environment.....	5
3.2 Add Pgr items.....	5
3.2.1 Pgr items relating to Android Application.....	5
3.2.2 Pgr items relating to Java general.....	6
3.3 Improvements of Pgr items.....	6
3.3.1 Improvement of pgj10124.....	6
3.3.2 Improvement of pgj10204.....	6
3.3.3 Improvement of pgj10214.....	7
3.3.4 Improvement of pgj10309.....	7
3.3.5 Improvement of pgj10323.....	7
3.3.6 Improvement of pgj10327.....	7
3.3.7 Improvement of pgj10328.....	7
3.3.8 Improvement for static import function.....	8
Chapter 4 Improvement for PGRelief J 2011 autumn.....	9
4.1 Correspondence to the latest OS/MW/IDE environment.....	9
4.2 Pgr items relating to Java general.....	9
4.3 Improvements of Pgr items.....	9
4.3.1 Improvement of pgj10233.....	9
4.4 Improvement of Operability/Convenience.....	9
Chapter 5 Improvement for PGRelief J 2012.....	10
5.1 Correspondence to the latest OS/MW/IDE environment.....	10
5.2 Add Pgr items.....	10
5.2.1 Pgr items relating to Android Application.....	10
5.2.2 Pgr items relating to Java general.....	10
5.3 Improvements of Pgr items.....	11
5.3.1 Improvement of pgj10214.....	11
5.4 Improvement of Operability/Convenience.....	11
5.4.1 Addition of check rule lock function.....	11
5.4.2 Reinforcement of operation environment error check.....	11
5.4.3 Addition of option in pgrjava command (floating license version).....	12
5.4.3.1 -qm option.....	12
5.4.3.2 -am option /-ar option /-arf option.....	12

Chapter 6 Improvement for PGRelief J 2012 autumn.....	13
6.1 Correspondence to the latest OS/MW/IDE environment.....	13
6.2 Improvements of Pgr items.....	13
6.2.1 Improvement of pgj10061.....	13
6.2.2 Improvement of pgj10065.....	13
6.2.3 Improvement of pgj10117.....	13
6.2.4 Improvement of pgj10120.....	13
6.2.5 Improvement of pgj10221.....	13
6.2.6 Improvement of pgj10300.....	14
Chapter 7 Improvement for PGRelief J 2013.....	15
7.1 Correspondence to the latest OS/MW/IDE environment.....	15
7.2 Add Pgr items.....	15
7.3 Improvements of Pgr items.....	16
7.3.1 For Java7.....	16
7.3.1.1 Improvement of pgj10044.....	16
7.3.1.2 Improvement of pgj10045.....	16
7.3.1.3 Improvement of pgj10235.....	16
7.3.1.4 Improvement of pgj10247.....	16
7.3.1.5 Improvement of pgj10301.....	16
7.3.1.6 Improvement of pgj10405.....	17
7.3.1.7 Improvement of pgj10414.....	17
7.3.2 Other improvements.....	17
7.3.2.1 Improvement of pgj10117.....	17
7.3.2.2 Improvement of pgj10120.....	17
7.4 Improvement of Operability/Convenience.....	17
7.4.1 Addition of evidence output function.....	17
7.4.2 Addition of Diagnosis report generation tool.....	18
Chapter 8 Improvement for PGRelief J 2013 autumn.....	19
8.1 Correspondence to the latest OS/MW/IDE environment.....	19
8.2 Improvements of Pgr items.....	19
8.2.1 Improvement of pgj10053.....	19
8.2.2 Improvement of pgj10059.....	19
8.2.3 Improvement of pgj10208.....	19
8.2.4 Improvement of pgj10210.....	20
8.2.5 Improvement of pgj10231.....	20
8.2.6 Improvement of pgj10237.....	20
8.2.7 Improvement of pgj10238.....	20
Chapter 9 Improvement for PGRelief J 2014.....	21
9.1 Correspondence to the latest OS/MW/IDE environment.....	21
9.2 Improvements of Pgr items.....	21
9.2.1 Improvement of pgj10003.....	21
9.2.2 Improvement of pgj10008.....	21
9.2.3 Improvement of pgj10010.....	21
9.2.4 Improvement of pgj10034.....	21
9.2.5 Improvement of pgj10084.....	22
9.2.6 Improvement of pgj10085.....	22
9.2.7 Improvement of pgj10221.....	22
9.3 Improvement of Operability/Convenience.....	22
9.3.1 Improvement of evidence output function.....	22
Chapter 10 Improvement for PGRelief J 2014 autumn.....	23
10.1 Correspondence to the latest OS/MW/IDE environment.....	23
Chapter 11 Improvement for PGRelief J 2015.....	24
11.1 Correspondence to the latest OS/MW/IDE environment.....	24

11.2 Add Pgr items.....	24
11.3 Improvements of Pgr items.....	25
11.3.1 Improvement of pgj10221.....	25
<b>Chapter 12 Improvement for PGRelief J 2015 autumn.....</b>	<b>26</b>
12.1 Correspondence to the latest OS/MW/IDE environment.....	26
12.2 Correspondence to the 64bit development environment.....	26
<b>Chapter 13 Improvement for PGRelief J 2016.....</b>	<b>27</b>
13.1 Correspondence to the latest OS/MW/IDE environment.....	27
13.2 Add Pgr items.....	27
13.3 Improvements of Pgr items.....	28
13.3.1 Improvement of pgj10219.....	28
13.3.2 Improvement of pgj10433.....	28
13.4 Improvement of Operability/Convenience.....	28
13.5 Offer of CI tool (Jenkins) Plugin(floating license version).....	28
<b>Chapter 14 Improvement for PGRelief J 2016 autumn.....</b>	<b>29</b>
14.1 Correspondence to the latest OS/MW/IDE environment.....	29
14.2 Improvements of Pgr items.....	29
14.2.1 Improvement of pgj10315.....	29
14.2.2 Improvement of pgj10316.....	29
14.2.3 Improvement of pgj10334.....	29
14.3 Correspondence to the large-scale project.....	29
14.3.1 Add pgrjava command's -Xmx option(floating license version).....	29
14.3.2 Add settings of the maximum heap size for Code check.....	30
<b>Chapter 15 Improvement for PGRelief J 2017.....</b>	<b>31</b>
15.1 Correspondence to the latest OS/MW/IDE environment.....	31
15.2 Improvements of Pgr items.....	31
15.3 Correspondence to the latest OS/MW/IDE environment.....	31
15.3.1 Improvement of the type resolve.....	31
15.3.1.1 Improvement of generics.....	31
15.3.1.2 Improvement of variable arguments .....	32
15.3.1.3 Improvement of boxing/unboxing conversion.....	32
15.3.2 Improvement of pgj10315.....	33
15.4 Improvement of Operability/Convenience.....	33
15.4.1 Addition of sharing function of code check result.....	33
<b>Chapter 16 Improvement for PGRelief J 2017 autumn.....</b>	<b>34</b>
<b>Chapter 17 Improvement for PGRelief J 2018.....</b>	<b>35</b>
17.1 Correspondence to the latest OS/MW/IDE environment.....	35
<b>Chapter 18 Improvement for PGRelief J 2018 autumn.....</b>	<b>36</b>
18.1 Correspondence to the latest IDE environment.....	36
<b>Chapter 19 Improvement for PGRelief J 2019.....</b>	<b>37</b>
19.1 Correspondence to the latest OS/MW/IDE environment.....	37
19.2 Change of CI Tool Collaboration Jenkins Plugin provision method.....	37
<b>Chapter 20 Improvement for Agile+ Relief J V1.1.1.....</b>	<b>38</b>
20.1 Correspondence to the latest OS/MW/IDE environment.....	38

# Chapter 1 Improvement for PGRelief J 2010

This chapter explains the improvements and changes that have been made from "PGRelief J 2009 autumn" to "PGRelief J 2010". In addition, the following improvements will be inherited in later versions.

## 1.1 Solutions for version upgrade of Java development environment tool

---

The following has been added as the Java development environment where PGRelief J works.

- Eclipse 3.5.1, 3.5.2
- Interstage Studio V9.2

## 1.2 Improvements of code check operation in Eclipse

---

Even if a project without a file to be checked is included when the code check is performed with multiple projects being selected on Eclipse, code check would continue.

## 1.3 Addition of cooperation tool

---

### 1.3.1 Solutions for version upgrade of FindBugs

---

FindBugs 1.3.9 has been added as a cooperation tool.

### 1.3.2 Addition of PMD cooperation

---

The PMD which is the static parsing tool of Java program has been added as a new cooperation tool. The target version is PMD 4.2.5.

## 1.4 Improvements of Pgr items

---

### 1.4.1 Improvement of pgj10034

---

In addition to "break" statement and "return" statement, when the "throw" statement is also described in the "case" block of the "switch" statement, it does not have to indicate.

### 1.4.2 Improvement of pgj10056

---

The rule title has been changed.

[New]

The number of array dimensions should not exceed the specified size

[Old]

The array size should not exceed the specified size

### 1.4.3 Improvement of pgj10212

---

As a check object for initialization of field variable, InstanceInitializer has been added. Therefore it will not indicate when a field variable is initialized by a Constructor or an InstanceInitializer.

### 1.4.4 Improvement of pgj10221

---

The comparison between same type of Enum is not indicated.

### 1.4.5 Improvement of pgj10226

---

The rule title has been changed.

[New]

When overriding the equals() method, the hashCode() method should also be overridden, and vice versa

[Old]

When overriding the hashCode() method, the equals() method should also be overridden, and vice versa

### 1.4.6 Improvement of pgj10235

---

The rule title has been changed.

[New]

Do not use the equals() method to compare objects with different types

[Old]

The equals() method that always returns false should be avoided

### 1.4.7 Improvement of pgj10412

---

The following has been added as a method invocation to be checked.

- java.lang.instrument.ClassDefinition

  - ClassDefinition(Class theClass, byte[] theClassFile)

- java.lang.instrument.Instrumentation

  - addTransformer(ClassFileTransformer transformer)

  - addTransformer(ClassFileTransformer transformer, boolean canRetransform)

  - redefineClasses(ClassDefinition... definitions)

## Chapter 2 Improvement for PGRelief J 2010 autumn

This chapter explains the improvements and changes that have been made from "PGRelief J 2010" to "PGRelief J 2010 autumn". In addition, the following improvements will be inherited in later versions.

### 2.1 Correspondence to the latest OS/IDE environment

---

The following has been added as the OS/IDE environment that PGRelief J operates.

[OS Environment]

- Windows(R) 7

Note) When you use Windows(R) 7 64 bit version, it operates by 32 bit emulation.

[IDE Environment]

- Eclipse 3.6.1 (32 bit version)

### 2.2 Improvements of Pgr items

---

#### 2.2.1 Improvement of the indicate regarding security vulnerability

---

The following two have been improved regarding the security vulnerability rule.

[Target Rule]

pgj10402、pgj10403、pgj10404、pgj10405、pgj10406、pgj10407、pgj10408、pgj10409、pgj10410

- 1) Improvement when the method and the field variable of the StringBuilder type are specified for optional constant.

When the method and the field of the StringBuilder type were specified for an optional constant, the method and the field were not treated as the constant but occasionally indicated.

When the method and the field of the StringBuilder type are specified for an optional constant, they are no longer indicated.

- 2) Improvement for treatment of the primitive wrapper class

Primitive wrapper classes except Character are treated as constants.

[Target of the wrapper class]

Boolean、Byte、Short、Integer、Long、Float、Double

#### 2.2.2 An optional constant is added to pgj10410

---

The following options have been added as an optional constant.

- java.io.File#separator
- java.io.File#separatorChar
- java.lang.System#mapLibraryName(java.lang.String)
- javax.servlet.ServletContext#getContextPath()
- javax.servlet.ServletContext#getRealPath(java.lang.String)



- javax.servlet.ServletRequest#getRealPath(java.lang.String)

The check rule definition file (rule.pgrj) that their effective optional constants are located under the PGRelief J installation folder.

For more information about configuration settings, please open the rule.pgrj by customizer, and refer to definition information in the dialog [Detailed Setting- pgj10410] that is displayed by [Detailed setting] button of pgj10410. If necessary, please add a similar setting to your check rule definition file.

For the operational procedure of the optional constant, please refer to "3.4.5.17 Specify Security Rules" in "Customizer Operation Guide".

# Chapter 3 Improvement for PGRelief J 2011

This chapter explains the improvements and changes that have been made from "PGRelief J 2010 autumn" to "PGRelief J 2011". In addition, the following improvements will be inherited in later versions.

## 3.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[OS Environment]

- Windows Server(R) 2008 R2

Note) When you use Windows Server(R) 2008 R2, it operates by 32 bit emulation.

[IDE Environment]

- Eclipse 3.6.2 (32 bit version)

## 3.2 Add Pgr items

### 3.2.1 Pgr items relating to Android Application

The following 18 rules have been added for Android application. For details of each rule, please refer to "Rule Detail Manual".

Table 3.1 List of added Android rules

Rule code	Rule content	Category
pgj20200	onCreate() method must be implemented	Android - Reliability
pgj20202	onPause() method must be implemented	Android - Reliability
pgj20203	The super class version should be invoked at the beginning of Activity lifecycle method	Android - Reliability
pgj20204	Do not perform persistent data writing in onStop(), onDestroy() and onSaveInstanceState()	Android - Reliability
pgj20205	BroadcastReceiver is registered through onResume(), and released through onPause()	Android - Reliability
pgj20206	Create dialog through onCreateDialog()	Android - Reliability
pgj20207	Do not register repeating buttons in the AlertDialog	Android - Reliability
pgj20208	When creating instance of Toast class, setView() method should be invoked before show() method	Android - Reliability
pgj20209	Commit() must be invoked last when save Preference	Android - Reliability
pgj20210	The resources opened by openFileOutput() or openFileInput() must be closed	Android - Reliability
pgj20211	Do not operate UI component outside UI thread	Android - Reliability
pgj20212	Do not use Object.wait() or Thread.sleep() to wait for the end of processing of sub-thread	Android - Reliability
pgj20213	Catch ActivityNotFoundException When calling startActivity() method	Android - Reliability
pgj20214	Do not describe an active logic in the logging block	Android - Reliability
pgj20300	Do not output the log record of DEBUG level to the release version	Android - Efficiency
pgj20301	Do not output the log record of VERBOSE level to the release version	Android - Efficiency

Rule code	Rule content	Category
pgj20302	Do not record log that is above the INFORMATIVE level during the disconnection of Network	Android - Efficiency
pgj20303	Do not record the log that is above the INFORMATIVE level in the log record relating to file system exception	Android - Efficiency

### 3.2.2 Pgr items relating to Java general

The following 5 rules have been added as Java general rules. For details of each rule, please refer to "Rule Detail Manual".

Table 3.2 List of added Java General Rules

Rule code	Rule content	Category
pgj10153	Declare the field variable at the beginning of class or before the method being used	Java general - Maintainability/ Readability
pgj10154	Set local variable to the minimum scope	Java general - Maintainability/ Readability
pgj10155	@Override annotation should be attached for the override method	Java general - Maintainability/ Readability
pgj10156	"TODO" in the TODO comment must be written with capital letters	Java general - Maintainability/ Readability
pgj10245	Do not use finalize() method	Java general - Reliability

## 3.3 Improvements of Pgr items

### 3.3.1 Improvement of pgj10124

In the default option of local variable with single character, StringBuilder has been added as the "p,q,r,s" type.

### 3.3.2 Improvement of pgj10204

When the finally block does not exist, it is unable to indicate the situation when super.finalize() method is not invocated in the finally block. Through the improvement, now it can be indicated.

<Example>

```

1 public class Sample {
2     public void finalize() throws Throwable {
3         try { // [Indication] super.finalize() is not invocated in finally block
4             :
5             super.finalize();
6         } catch (IOException e) {
7             :
8         }
9     }
10 }
```

### 3.3.3 Improvement of pgj10214

---

The static initializer has been added as the check target for initialization of static field variable. As a result, when the static field variable is being declared or it is being initialized by the static initializer, no indication will occur.

### 3.3.4 Improvement of pgj10309

---

The StringBuilder type has been added as the type of local variable which becomes the check target. As a result, since the type of local variable is StringBuilder type, it will be indicated when the value is not changed.

Through this improvement, the rule title will be changed as follows.

[New]

Do not declare string constant by StringBuffer type and StringBuilder type

[Old]

Do not declare string constant by StringBuffer type

### 3.3.5 Improvement of pgj10323

---

The StringBuilder has been added as detected object. As a result, when the StringBuilder object was created without specifying the initial size, the situation will be indicated.

Through this improvement, the rule title will be changed as follows.

[New]

Specify the initial size when StringBuffer/StringBuilder is generated

[Old]

Specify the initial size when StringBuffer is generated

### 3.3.6 Improvement of pgj10327

---

The rule title has been changed.

[New]

Use StringBuffer or StringBuilder for frequent string linkage

[Old]

Use StringBuffer for frequent string linkage

### 3.3.7 Improvement of pgj10328

---

The rule title has been changed.

[New]

Use StringBuffer or StringBuilder when strings are being replaced frequently

[Old]

Use StringBuffer when strings are being replaced frequently

### 3.3.8 Improvement for static import function

---

Improvement has been made to the check when the description related to the static import function exists.

In the following program example, since the static import has been done to `java.lang.System.gc` at the first line, the method invocation at the fourth line is unable to recognize the invocation of `System.gc()`. Therefore, the violation to the rule `pgj10300` "Do not invoke `System.gc()`" cannot be indicated. After improvement has been made, it can be indicated properly.

<Example>

```
1  import static java.lang.System.gc ;
2  class MyClass {
3      void method(){
4          gc() ;    // Unable to indicate violation of rule pgj10300.
5      }
6  }
```

In addition, when the static import function is being used, `pgj10244` "Do not perform standard output or the output that may cause standard error" cannot be detected correctly. For details, please refer to "Chapter 6 Cautions" of "Code check Operation Guide".

# Chapter 4 Improvement for PGRelief J 2011 autumn

This chapter explains the improvements and changes that have been made from "PGRelief J 2011" to "PGRelief J 2011 autumn". In addition, the following improvements will be inherited in later versions.

## 4.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 3.7.0 (32 bit version)
- Android SDK 3.1, 3.2

## 4.2 Pgr items relating to Java general

The following 4 rules have been added as Java general rules. For details of each rule, please refer to "Rule Detail Manual".

Table 4.1 List of added Java General Rules

Rule code	Rule content	Category
pgj10246	Do not use equals(class type) method to overload equals(Object) method	Java general - Reliability
pgj10247	Field variable that does not use equals() method to compare cannot be used for generating hash code	Java general - Reliability
pgj10248	Do not describe if statement, which conditional part is logic type literal only	Java general - Reliability
pgj10249	Do not perform meaningless assignment operation	Java general - Reliability

## 4.3 Improvements of Pgr items

### 4.3.1 Improvement of pgj10233

As a return statement of detect section, a return statement that does not return a value is added.

## 4.4 Improvement of Operability/Convenience

When Eclipse performed Code check of PGRelief, the Code check result was always output under the Eclipse project folder. In this version, any folder can be specified as the output folder of Code check result.

For detailed information on how to specify an output folder for code check results, refer to "2.2.3 Set output folder for Code check result" of "Code check Operation Guide".

# Chapter 5 Improvement for PGRelief J 2012

This chapter explains the improvements and changes that have been made from "PGRelief J 2011 autumn" to "PGRelief J 2012". In addition, the following improvements will be inherited in later versions.

## 5.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 3.7.1, 3.7.2 (32 bit version)
- Interstage Studio V10.0, V10.1
- Android SDK 4.0

[Cooperation Tools]

- FindBugs 2.0
- PMD 4.2.6, 4.3

## 5.2 Add Pgr items

### 5.2.1 Pgr items relating to Android Application

The following 6 rules have been added for Android application. For details of each rule, please refer to "Rule Detail Manual".

Given these additions, options for Android rule check have been added in the customizer. For more information, please refer to "3.2.3 Set Android Options" in "Customizer Operation Guide".

Table 5.1 List of added Android rules

Rule code	Rule content	Category
pgj20000	Prepare external resources for onscreen character strings	Android - Maintainability/ Readability
pgj20001	Prepare external resources for onscreen character strings (resource file)	Android - Maintainability/ Readability
pgj20215	When supporting a lower API Level, pay attention to the use of the added API	Android - Reliability
pgj20216	When supporting a higher API Level, pay attention to the use of the deleted API	Android - Reliability
pgj20217	Pay attention to any changes in API between API Level	Android - Reliability
pgj20218	Always define minSdkVersion	Android - Reliability

### 5.2.2 Pgr items relating to Java general

The following 6 rules have been added as Java general rules. For details of each rule, please refer to "Rule Detail Manual".

Table 5.2 List of added Java General Rules

Rule code	Rule content	Category
pgj10157	When rethrowing an exception, keep the original exception information	Java general - Maintainability/ Readability
pgj10250	Do not write an empty if block	Java general - Reliability
pgj10251	Do not write an empty try block	Java general - Reliability
pgj10252	Do not write an empty constructor	Java general - Reliability
pgj10413	Pay attention to the injection to search criteria in the directory service	Java general - Functionality
pgj10414	Pay attention to the injection to an regular expression	Java general - Functionality

## 5.3 Improvements of Pgr items

---

### 5.3.1 Improvement of pgj10214

---

You can now select either "all" or "reference types" as the type of a static field variable for a detected object. Due to this change, the rule title has been changed.

Rule Title

[New]

Explicitly initialize the static field variables

[Old]

Explicitly initialize all the static field variables

## 5.4 Improvement of Operability/Convenience

---

### 5.4.1 Addition of check rule lock function

---

A function to lock applied rules has been added to the customizer 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. For more information, please refer to "3.4.7 Lock / Unlock Applied Rules" in "Customizer Operation Guide".

### 5.4.2 Reinforcement of operation environment error check

---

Class files are analyzed in the rule checking for PGR relief security vulnerability. Hence when the class files are older than the Java files, checking will not be executed properly.

In the previous version, checking was done to see whether the class files exist. Adding to this feature, in this version, a function was added to compare the last updated dates of the Java files and the class files to check whether the class files are the latest version. This function outputs a message either ERR104 or ERR106 when an error occurs. For more information on errors, please refer to "Chapter 5 Operating Environment Error" in "Code Check Operation Guide".



This function gets activated when a rule for security vulnerability has the check allowed status, or when FindBugs cooperation is specified. The Security vulnerability rule applies to the following rules.

<Security Vulnerability Rules>

pgj10402, pgj10403, pgj10404, pgj10405, pgj10406, pgj10407, pgj10408, pgj10409, pgj10410, pgj10412, pgj10413, pgj10414

## **5.4.3 Addition of option in pgrjava command (floating license version)**

---

### **5.4.3.1 -qm option**

In response to the "analysis result consolidation mode" function that was added in the "PGRelief 2012 Quality Management Option", "-qm" option has been added to the pgrjava command.

For more information on "analysis result consolidation mode", please refer to "2.3.2 Procedure of using under analysis result consolidation mode" in "Quality Management Option Manual".

For more information on the -qm option, please refer to "Chapter 1 pgrjava Command" in "Command Manual".

### **5.4.3.2 -am option /-ar option /-arf option**

In response to the addition of Android rules pgj20001 or pgj20218, -am option/-ar option/-arf option have been added.

For details of each option, please refer to "Chapter 1 pgrjava Command" in "Command Manual".

# Chapter 6 Improvement for PGRelief J 2012 autumn

This chapter explains the improvements and changes that have been made from "PGRelief J 2012 " to "PGRelief J 2012 autumn". In addition, the following improvements will be inherited in later versions.

## 6.1 Correspondence to the latest OS/MW/IDE environment

---

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 4.2.0 (32 bit version)
- Android SDK 4.1

## 6.2 Improvements of Pgr items

---

### 6.2.1 Improvement of pgj10061

---

You can now select either "all" or "Do not detect while/do-while" as the hidden assignment for a detected object.

When "Do not detect while/do-while" is selected, the hidden assignment will not be detected even if it is used in while/do-while condition part. However, the nest assignment will be detected.

### 6.2.2 Improvement of pgj10065

---

As a method of detect section, the following methods whose classes inherit from classes that implement Serializable interfaces will not be detected.

- void readObject(java.io.ObjectInputStream)
- void writeObject(java.io.ObjectOutputStream)
- void readObjectNoData()
- java.lang.Object readResolve()
- java.lang.Object writeReplace()

### 6.2.3 Improvement of pgj10117

---

As a constant of detect section, the serialVersionUID whose class implements a Serializable interface will not be detected.

### 6.2.4 Improvement of pgj10120

---

As a constant of detect section, the serialVersionUID whose class implements a Serializable interface will not be detected.

### 6.2.5 Improvement of pgj10221

---

The comparison expression that uses the "Auto boxing function" will not be detected. The rule title has been changed.

Rule Title

[New]

Do not use "==" or "!=" when comparing with object

[Old]

Do not use "==" or "!=" while comparing between objects

## **6.2.6 Improvement of pgj10300**

---

As a method of detect section, Runtime.gc() has been added. Due to this change, the rule title has been changed.

Rule Title

[New]

Do not explicitly invoke garbage collection

[Old]

Do not invoke System.gc()

# Chapter 7 Improvement for PGRelief J 2013

This chapter explains the improvements and changes that have been made from "PGRelief J 2012 autumn" to "PGRelief J 2013". In addition, the following improvements will be inherited in later versions.

## 7.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[OS Environment]

- Windows(R) 8
- Windows Server(R) 2012

Note) When you use Windows(R) 8 64 bit version or Windows Server(R) 2012 64 bit version, it operates by 32 bit emulation.

Note) Does not correspond to Modern UI design.

[IDE Environment]

- Eclipse 4.2.1, 4.2.2 (32 bit version)
- Interstage Studio V11.0
- Android SDK 4.2
- Java 7

[Cooperation Tools]

- FindBugs 2.0.1
- PMD 5.0.0, 5.0.1

## 7.2 Add Pgr items

The following 12 rules have been added for Android application. For details of each rule, please refer to "Rule Detail Manual".

Table 7.1 List of added Android rules

Rule code	Rule content	Category
pgj20400	Do not save sensitive information in external storage	Android - Functionality
pgj20401	Local file is made with MODE_PRIVATE	Android - Functionality
pgj20402	SQLite database file is made with MODE_PRIVATE	Android - Functionality
pgj20403	Limit the range of opening to the public of Activity	Android - Functionality
pgj20404	Limit the range of opening to the public of Service	Android - Functionality
pgj20405	Limit the range of opening to the public of BroadcastReceiver	Android - Functionality
pgj20406	Limit the range of opening to the public of ContentProvider	Android - Functionality

Rule code	Rule content	Category
pgj20407	Do not output sensitive information to the log	Android - Functionality
pgj20408	Set the parameter of Intent by the putExtra() method	Android - Functionality
pgj20409	Use the permitted permission	Android - Functionality
pgj20410	Do not use the permission that has not been permitted	Android - Functionality
pgj20411	Do not use WebView#addJavascriptInterface(Object, String)	Android - Functionality

## 7.3 Improvements of Pgr items

---

### 7.3.1 For Java7

---

#### 7.3.1.1 Improvement of pgj10044

At try-with-resources, it came to point it out when there was no blank between "try(resources)" and "{".

#### 7.3.1.2 Improvement of pgj10045

At try-with-resources, it came to point it out when there was no blank between "try(resources)" and "{".

#### 7.3.1.3 Improvement of pgj10235

When objects with different types were compared by using java.util.Objects#equals(Object, Object), it came to be detected.

#### 7.3.1.4 Improvement of pgj10247

java.util.Objects#equals(Object, Object) was added as a comparison means of the field variable.

#### 7.3.1.5 Improvement of pgj10301

When the String object and empty characters were compared by using java.util.Objects#equals(Object, Object), it came to be detected.

Due to this change, the rule title has been changed.

##### Rule Title

[New]

Do not compare String object and empty characters

[Old]

Do not pass "" to the parameter of String.equals()

Moreover, the following description pattern was added as a detection target.

```
"".equals(String object)
```

### 7.3.1.6 Improvement of pgj10405

The following has been added as a method invocation to be checked.

- java.nio.file.FileSystem  
  getPath(java.lang.String, java.lang.String[])
- java.nio.file.Path  
  resolve(java.lang.String)  
  resolveSibling(java.lang.String)
- java.nio.file.Paths  
  get(java.lang.String, java.lang.String[])

### 7.3.1.7 Improvement of pgj10414

The following has been added as a method invocation to be checked.

- java.nio.file.FileSystem  
  getPathMatcher(java.lang.String)

## 7.3.2 Other improvements

---

### 7.3.2.1 Improvement of pgj10117

As a constant of detect section, the serialPersistentFields whose class implements a Serializable interface will not be detected.

### 7.3.2.2 Improvement of pgj10120

As a constant of detect section, the serialPersistentFields whose class implements a Serializable interface will not be detected.

## 7.4 Improvement of Operability/Convenience

---

### 7.4.1 Addition of evidence output function

---

The function to output the result of Code check as an evidence was added.

A similar function already includes the function to output the file of Comma Separated Value from the [PGRelief Message List] view etc. of Eclipse. However, it has the possibility of causing the user's output mistake as the output result is different etc. according to the state of the filter of the view.

On the other hand, such a mistake does not happen so that the evidence output function added in this version may output all the results regardless of the state of the display of the view. Moreover, metrics information on the number of lines and the complexity etc. of the program is newly added.

For details, please refer to "3.5 Output Code check Evidence" in "Code check Operation Guide".

## **7.4.2 Addition of Diagnosis report generation tool**

---

The diagnosis report generation tool that generated the report of the source diagnosis from the result of Code check of PGRelief J was added. Please use "Diagnosis report generation tool" from the start menu or the start screen(since Windows(R) 8, Windows Server(R) 2012) of Windows(R).

In PGRelief J, these evidence file is input, and Diagnosis report generation tool that outputs the result of analyzing the information as a report of the Book-type of Microsoft(R) Office Excel(R) is being offered.

# Chapter 8 Improvement for PGRelief J 2013 autumn

This chapter explains the improvements and changes that have been made from "PGRelief J 2013" to "PGRelief J 2013 autumn". In addition, the following improvements will be inherited in later versions.

## 8.1 Correspondence to the latest OS/MW/IDE environment

---

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 4.3.0 (32 bit version)
- Android SDK 4.3

[Cooperation Tools]

- FindBugs 2.0.2
- PMD 5.0.2 - 5.0.5

## 8.2 Improvements of Pgr items

---

### 8.2.1 Improvement of pgj10053

---

A labeled continue will be detected. Due to this change, the rule title has been changed.

Rule Title

[New]

Do not jump to the label

[Old]

Do not jump to the label by break command

### 8.2.2 Improvement of pgj10059

---

The following has been added as a class to be checked.

- inner class
- local class
- interface
- inner interface
- enum
- annotation

### 8.2.3 Improvement of pgj10208

---

Enhanced for loops was added as a check target.



## 8.2.4 Improvement of pgj10210

---

The assignment operator to be detected can be selected from the following assignment operators. In default, "=" is checked.

=, +=, -=, \*=, /=, %=, &=, |=, ^=, <<=, >>=, >>>=

## 8.2.5 Improvement of pgj10231

---

When the Exception class was declared to the throws phrase of the override method, it can be selected whether to detect it.

## 8.2.6 Improvement of pgj10237

---

The following methods has been added as detected object.

- Object#wait(long)
- Object#wait(long, int)

## 8.2.7 Improvement of pgj10238

---

The following methods has been added as detected object.

- Object#wait(long)
- Object#wait(long, int)

# Chapter 9 Improvement for PGRelief J 2014

This chapter explains the improvements and changes that have been made from "PGRelief J 2013autumn" to "PGRelief J 2014 ". In addition, the following improvements will be inherited in later versions.

## 9.1 Correspondence to the latest OS/MW/IDE environment

---

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[OS Environment]

- Windows(R) 8.1
- Windows Server(R) 2012 R2

Note) When you use Windows(R) 8.1 64 bit version or Windows Server(R) 2012 R2, it operates by 32 bit emulation.

Note) Does not correspond to Modern UI design.

[IDE Environment]

- Eclipse 4.3.1, 4.3.2 (32 bit version)
- Interstage Studio V11.1
- Android SDK 4.4
- Java 8

[Cooperation Tools]

- FindBugs 2.0.3

## 9.2 Improvements of Pgr items

---

### 9.2.1 Improvement of pgj10003

---

The annotation type and the enum type has been added as a detected target.

### 9.2.2 Improvement of pgj10008

---

The do-while have been added as a detected target.

### 9.2.3 Improvement of pgj10010

---

The annotation type and the enum type has been added as a detected target.

### 9.2.4 Improvement of pgj10034

---

In case of the switch statement, when the continue statement in addition to the break statement, the return statement, and the throw statement was described, it was not detected.

Moreover, case without processing came to be able to control detection by customizer. For details, please refer to "3.4.5.19 case without processing is excluded from the check" in "Customizer Operation Guide".

## 9.2.5 Improvement of pgj10084

---

The annotation type and the enum type has been added as a detected target.

## 9.2.6 Improvement of pgj10085

---

The enum constants have been added as detected object.

## 9.2.7 Improvement of pgj10221

---

Even if the method invocation exists, the comparison expression that uses the "Auto boxing function" and the comparison expression that compares the same enum type will not be detected. Due to this change, the message has been changed.

### Message

[New]

"==" or "!=" has been used in the comparison of object.

[Old]

"==" or "!=" has been used in the comparison of object (<variable name>).

## 9.3 Improvement of Operability/Convenience

---

### 9.3.1 Improvement of evidence output function

---

The evidence came to be able to output to the unit of the file. The evidence can be output by selecting one or two or more files on eclipse.

For details, please refer to "3.5.1 Output Evidence" in "Code check Operation Guide".

## Chapter 10 Improvement for PGRelief J 2014 autumn

This chapter explains the improvements and changes that have been made from "PGRelief J 2014" to "PGRelief J 2014 autumn ". In addition, the following improvements will be inherited in later versions.

### 10.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 4.4.0 (32 bit version)
- Android SDK 4.4W

[Cooperation Tools]

- FindBugs 3.0.0
- PMD 5.1.0 - 5.1.2

# Chapter 11 Improvement for PGRelief J 2015

This chapter explains the improvements and changes that have been made from "PGRelief J 2014 autumn" to "PGRelief J 2015". In addition, the following improvements will be inherited in later versions.

## 11.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 4.4.1, 4.4.2 (32 bit version)
- Android SDK 5.0

[Cooperation Tools]

- PMD 5.1.3 - 5.2.3

## 11.2 Add Pgr items

The following 19 rules have been added as Java general rules. For details of each rule, please refer to "Rule Detail Manual".

Table 11.1 List of added Java general Rules

Rule code	Rule content	Category
pgj10415	Do not use the hard-coding ID or the hard-coding password to connect to the database	Java general - Functionality
pgj10416	Invalidate the session before the invocation of LoginContext.login()	Java general - Functionality
pgj10417	Specify the character-set for Set-Content	Java general - Functionality
pgj10418	Do not specify the domain attribute of the cookie	Java general - Functionality
pgj10419	Enable the secure attribute of the cookie	Java general - Functionality
pgj10420	Do not specify the expires attribute of the cookie	Java general - Functionality
pgj10421	Do not specify the route for a value of the path attribute of the cookie	Java general - Functionality
pgj10422	Enable the HttpOnly attribute of the cookie	Java general - Functionality
pgj10423	Do not output the exception information	Java general - Functionality
pgj10424	Encrypt the socket communication as much as possible	Java general - Functionality
pgj10425	Invoke the check method of the SecurityManager class before the invocation of the sensitive processing	Java general - Functionality
pgj10426	Do not allocate the dangerous combinations of permissions	Java general - Functionality

Rule code	Rule content	Category
pgj10427	Use the encryption algorithm with high safety	Java general - Functionality
pgj10428	Use the hash function algorithm with high safety	Java general - Functionality
pgj10429	Use the pseudorandom numbers generate algorithm with high safety	Java general - Functionality
pgj10430	Disable the cache, when an important content is treated	Java general - Functionality
pgj10431	Be coherent in the user authentication processing	Java general - Functionality
pgj10432	Be coherent in the access control processing	Java general - Functionality
pgj10433	Escape in the value passed to the response header	Java general - Functionality

## 11.3 Improvements of Pgr items

---

### 11.3.1 Improvement of pgj10221

---

This rule as in the following cases, it does not have to indicate.

- Comparison of Class types
- The instance(this) and the argument are compared by implements the equals(Object) method.

## Chapter 12 Improvement for PGRelief J 2015 autumn

This chapter explains the improvements and changes that have been made from "PGRelief J 2015" to "PGRelief J 2015 autumn". In addition, the following improvements will be inherited in later versions.

### 12.1 Correspondence to the latest OS/MW/IDE environment

---

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[OS Environment]

- Windows(R) 10

[IDE Environment]

- Eclipse 4.4.2 (64 bit version), 4.5.0 (32/64 bit version)
- Android SDK 5.1

[Cooperation Tools]

- FindBugs 3.0.1
- PMD 5.3.0 - 5.3.3

[Server virtualization software] \* Only the floating license version corresponds.

- Microsoft(R) Windows Server(R) 2012 R2 Hyper-V(R)
- VMware vSphere(R) 5.0

### 12.2 Correspondence to the 64bit development environment

---

It came to be able to analyze by the development environment of 64bit(Java (64bit version) and Eclipse (64bit version)) by installing PGRelief J (x64 edition).

# Chapter 13 Improvement for PGRelief J 2016

This chapter explains the improvements and changes that have been made from "PGRelief J 2015 autumn" to "PGRelief J 2016". In addition, the following improvements will be inherited in later versions.

## 13.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 4.5.1, 4.5.2 (32/64 bit version)
- Android SDK 6.0

[Cooperation Tools]

- PMD 5.3.4 - 5.4.1

## 13.2 Add Pgr items

The following 17 rules have been added as Java general rules. For details of each rule, please refer to "Rule Detail Manual".

Table 13.1 List of added Java general Rules

Rule code	Rule content	Category
<a href="#">pgj10434</a>	Do not output the file to the place where the access control is not done	Java general - Functionality
<a href="#">pgj10435</a>	Do not hide the menubar of the browser view	Java general - Functionality
<a href="#">pgj10436</a>	Do not hide the statusbar of the browser view	Java general - Functionality
<a href="#">pgj10437</a>	Do not hide the addressbar of the browser view	Java general - Functionality
<a href="#">pgj10438</a>	Do not use the deprecated protocol when Web Socket communication is used	Java general - Functionality
<a href="#">pgj10439</a>	Do not execute the authentication, the session management, or the access control using only the information that can misrepresent	Java general - Functionality
<a href="#">pgj10440</a>	Encrypt the Web Socket communication when the private information or the sensitive information are treated	Java general - Functionality
<a href="#">pgj10441</a>	Create the new session, if it succeeded in the authentication	Java general - Functionality
<a href="#">pgj10442</a>	Control the number of the running processes	Java general - Functionality
<a href="#">pgj10443</a>	Do not pass the important data to the cookie	Java general - Functionality
<a href="#">pgj10444</a>	Verify the value of the input parameter	Java general - Functionality
<a href="#">pgj10445</a>	Do not compare of part to the Origin header	Java general - Functionality
<a href="#">pgj10446</a>	Set the reliable domain to the Access-Control-Allow-Origin header	Java general - Functionality



Rule code	Rule content	Category
<a href="#">pgj10447</a>	Escape the HTML special character	Java general - Functionality
<a href="#">pgj10448</a>	Do not store the important information in the shared scope	Java general - Functionality
<a href="#">pgj10449</a>	Initialize safely the pseudorandom number	Java general - Functionality
<a href="#">pgj10450</a>	Do not use the hard-coding encryption key	Java general - Functionality

## 13.3 Improvements of Pgr items

---

### 13.3.1 Improvement of pgj10219

---

This rule came to be able to select the type of the argument that is detected.

### 13.3.2 Improvement of pgj10433

---

The following method was added to be detected. The message has been changed.

- HttpServletResponse#sendRedirect(String)

Message

[New]

There is danger of the HTTP header injection.

[Old]

There is danger of the cross site request forgeries.

## 13.4 Improvement of Operability/Convenience

---

It dealt with the problem that the error occurs when a lot of reference class libraries were specified by the PMD cooperation or the FindBugs cooperation.

## 13.5 Offer of CI tool (Jenkins) Plugin(floating license version)

---

Plugin that cooperated with Jenkins of the CI(Continuous Integration) tool was added.

The execution result of the PGRelief J came to be able to be consolidated on Jenkins, and to be displayed by applying plugin.

Please refer to 'CI Tool Collaboration Jenkins Plugin Manual' ("Etc\ci-plugin\pgrelief-jenkins-plugin\_en.pdf") stored in the installation medium for details.

# Chapter 14 Improvement for PGRelief J 2016 autumn

This chapter explains the improvements and changes that have been made from "PGRelief J 2016" to "PGRelief J 2016 autumn". In addition, the following improvements will be inherited in later versions.

## 14.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 4.6.0 (32/64 bit version)

[Cooperation Tools]

- PMD 5.3.7, 5.4.2, 5.5.0, 5.5.1

## 14.2 Improvements of Pgr items

### 14.2.1 Improvement of pgj10315

It will not be detected when there was the invocation of `FileInputStream#getChannel()` in the same method.

### 14.2.2 Improvement of pgj10316

It will not be detected when there was the invocation of `FileOutputStream#getChannel()` in the same method.

### 14.2.3 Improvement of pgj10334

It will not be detected when becoming the ambiguous invocation by the overload method because the cast expression of the argument is removed.

<Example>

```
1 public void test(){
2     if (doSomething((String)null)){ // Because it is a description necessary to clarify the
invocation of the method, it does not be detected.
3         :
4     }
5 }
6
7 private boolean doSomething (String s){
8     :
9 }
10
11 private boolean doSomething (Integer s){
12     :
13 }
```

## 14.3 Correspondence to the large-scale project

### 14.3.1 Add pgrjava command's -Xmx option(floating license version)

The pgrjava command's -Xmx option is added. It will be able to specify the maximum heap size for Code check. Please refer to "Chapter 1 pgrjava Command" of "Command Manual" for details.

## **14.3.2 Add settings of the maximum heap size for Code check**

---

It will be able to specify the maximum heap size for Code check. Please refer to "2.2.5 Set Maximum Heap Size for Code check" of "Code check Operation Guide" for details of the maximum heap size setting method.

# Chapter 15 Improvement for PGRelief J 2017

This chapter explains the improvements and changes that have been made from "PGRelief J 2016 autumn" to "PGRelief J 2017". In addition, the following improvements will be inherited in later versions.

## 15.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[OS Environment]

- Windows Server(R) 2016

[IDE Environment]

- Eclipse 4.6.1, 4.6.2 (32/64 bit version)
- Android SDK 7.0

[Cooperation Tools]

- PMD 5.3.8, 5.4.3, 5.4.4, 5.5.2, 5.5.3

[Server virtualization software] \* Only the floating license version corresponds.

- Microsoft(R) Windows Server(R) 2016 Hyper-V(R)

## 15.2 Improvements of Pgr items

## 15.3 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 4.6.3, 4.7.0 (32/64 bit version)
- Android SDK 7.1

[Cooperation Tools]

- PMD 5.4.5, 5.4.6, 5.5.4, 5.5.5, 5.5.6, 5.5.7, 5.6.0, 5.6.1, 5.7.0, 5.8.0, 5.8.1

### 15.3.1 Improvement of the type resolve

#### 15.3.1.1 Improvement of generics

It will be resolve correctly the method invocation with the type variable in the argument.

<Example >

```
01 public class MyClassA01
02 {
03     public void caller10(){
04         String s = "";
```

```

05     MyClassA02<Integer> obj = new MyClassA02();
06     for ( int i = 0 ; i < 10; i++ ) {
07         obj.caller20(new Integer(1), s) ; // The method invocation with the type variable in the
arguments.
08     }
09 }
10 }
11
12 class MyClassA02<T>
13 {
14     public void caller20(T t, String s)
15     {
16         s.concat( "b" ) ; // pgj10327 is detected, because the method invocation of
17     } // the 7th line is resolved.
18 }

```

### 15.3.1.2 Improvement of variable arguments

It will be resolve correctly the method invocation with the variable arguments in the argument.

<Example>

```

01 public class MyClassB01
02 {
03     public void method01()
04     {
05         for (int i = 0; i < 10; i++) {
06             method02("a", "b"); // method invocation with variable arguments
07         }
08     }
09     private void method02(String ...ss)
10     {
11         ss[0].concat("b"); // pgj10327 is detected, because the method invocation of
12     } // the 6th line is resolved.
13 }

```

### 15.3.1.3 Improvement of boxing/unboxing conversion

It will be resolve correctly the method invocation with the boxing/unboxing conversion in the argument.

<Example>

```

01 public class MyClassC01
02 {
03     public void method01()
04     {
05         for (int i = 0; i < 10; i++) {
06             method02(new Integer(1), "b"); // The unboxing conversion works by the 1st argument.
07         }
08     }
09     private void method02(int i, String s)
10     {
11         s.concat("b"); // pgj10327 is detected, because the method invocation of
12     } // the 6th line is resolved.
13 }
14

```

```

15 class MyClassC02
16 {
17     public void method03()
18     {
19         for (int i = 0; i < 10; i++) {
20             method04(1, "b"); // The boxing conversion works by the 1st argument.
21         }
22     }
23     private void method04(Integer i, String s)
24     {
25         s.concat("b"); // pgj10327 is detected, because the method invocation of
26     } // the 20th line is resolved.
27 }

```

## 15.3.2 Improvement of pgj10315

---

It will not be detected when the close() method is invoked by parent's try statement.

<Example>

```

01 public void doPost(HttpServletRequest request, HttpServletResponse response) {
02     String result = "";
03     PrintWriter pw = null;
04     try {
05         try {
06             pw = new PrintWriter(osw); // It is not detected, because the close() method is invoked
by the 13th line.
07         } catch (Exception e) {
08         }
09     } catch ( NullPointerException e){
10     } finally {
11         // close
12         if ( pw != null ) {
13             pw.close();
14         }
15     }
16 }

```

## 15.4 Improvement of Operability/Convenience

---

### 15.4.1 Addition of sharing function of code check result

---

The function to import/export of the code check result was added. The code check result (include the state of the confirmation and the confirmation memo) can be shared among the developers. For details, please refer to "3.6 Share Code check result" in "Code check Operation Guide".

## **Chapter 16 Improvement for PGRelief J 2017 autumn**

This chapter explains the improvements and changes that have been made from "PGRelief J 2017" to "PGRelief J 2017 autumn". In addition, the following improvements will be inherited in later versions.

# Chapter 17 Improvement for PGRelief J 2018

This chapter explains the improvements and changes that have been made from "PGRelief J 2017 autumn" to "PGRelief J 2018". In addition, the following improvements will be inherited in later versions.

## 17.1 Correspondence to the latest OS/MW/IDE environment

---

The following has been added as the OS/MW/IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 4.7.1, 4.7.2, 4.7.3 (32/64 bit version)

[Cooperation Tools]

- PMD 6.0.0, 6.0.1, 6.1.0



## Chapter 18 Improvement for PGRelief J 2018 autumn

This chapter explains the improvements and changes that have been made from "PGRelief J 2018" to "PGRelief J 2018 autumn". In addition, the following improvements will be inherited in later versions.

### 18.1 Correspondence to the latest IDE environment

---

The following has been added as the IDE environment that PGRelief J operates.

[IDE Environment]

- Eclipse 4.8.0, 4.9.0 (64 bit version)
- Java11 \* Only the operating environment corresponds. It does not correspond to the syntax added since Java9.

# Chapter 19 Improvement for PGRelief J 2019

This chapter explains the improvements and changes that have been made from "PGRelief J 2018 autumn" to "PGRelief J 2019". In addition, the following improvements will be inherited in later versions.

## 19.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the IDE environment that PGRelief J operates.

[OS Environment] \* Only the floating license version corresponds.

- Windows Server(R) 2019

[IDE Environment]

- Eclipse 4.10.0, 4.11.0 (64 bit version)

[Server virtualization software] \* Only the floating license version corresponds.

- Microsoft(R) Windows Server(R) 2019 Hyper-V(R)

[Microsoft(R) Office Excel(R)] \* Using "Diagnosis report generation tool".

- Microsoft(R) Office Excel(R) 2016
- Microsoft(R) Office Excel(R) 2019

## 19.2 Change of CI Tool Collaboration Jenkins Plugin provision method

The "CI Tool Collaboration Jenkins Plugin" that was provided in the floating license version installation media has been changed to be available as a download from the product HP. For details on the plug-in, please refer to "CI Tool Collaboration Jenkins Plugin" on the following download site.

<https://www.fujitsu.com/jp/products/software/resources/technical/agilerelief/search/>

\* When problems such as EOL (End-of-life) and security vulnerabilities occur in the open source software used in "CI Tool Collaboration Jenkins Plugin", the provision method has been changed so that it can be supported as needed.

## Chapter 20 Improvement for Agile+ Relief J V1.1.1

This chapter explains the improvements and changes that have been made from "PGRelief J 2019" to "Agile+ Relief J V1.1.1". In addition, the following improvements will be inherited in later versions.

### 20.1 Correspondence to the latest OS/MW/IDE environment

The following has been added as the OS/MW/IDE environment that Agile+ Relief J operates.

[Microsoft(R) Office Excel(R)] \* Using "Diagnosis report generation tool".

- Microsoft(R) Excel(R) for Office 365
- Microsoft(R) Office Excel(R) 2021 (64 bit version)

[OS Environment]

- Windows(R) 11 (64 bit version)
- Windows Server(R) 2022 \* Only the floating license version for 64bit OS corresponds.

[Server virtualization software] \* Only the floating license version corresponds.

- Microsoft(R) Windows Server(R) 2022 Hyper-V(R)
- VMware vSphere(R) 7.0