

Square 15-A-SP2

Eclipse Plugin Guide

Reference : SUM_Square_ECL_PLG
Version : 15-A-SP2
Date : 07/10/2015

Eclipse Plugin Guide

Copyright © 2015 Squoring Technologies

Abstract

This edition of the Eclipse Plugin Guide applies to Squore 15-A-SP2 and to all subsequent releases and modifications until otherwise indicated in new editions.

Licence

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of the copyright owner, Squoring Technologies.

Squoring Technologies reserves the right to revise this publication and to make changes from time to time without obligation to notify authorised users of such changes. Consult Squoring Technologies to determine whether any such changes have been made.

The terms and conditions governing the licensing of Squoring Technologies software consist solely of those set forth in the written contracts between Squoring Technologies and its customers.

All third-party products are trademarks or registered trademarks of their respective companies.

Warranty

Squoring Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Squoring Technologies shall not be liable for errors contained herein nor for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Table of Contents

Typographical Conventions	iv
Acronyms and Abbreviations	v
1. Introduction	1
1.1. Foreword	1
1.2. About This Document	1
1.3. Contacting Squoring Technologies Product Support	1
1.4. Responsibilities	1
1.5. Getting the Latest Version of this Manual	1
2. Introducing the Squore Eclipse Plugin	2
2.1. Squore Eclipse Plugin in the Squore ecosystem	2
2.2. Who is the Squore Eclipse Plugin destined to?	3
3. Installing and setting up the Squore Eclipse Plugin	4
3.1. Squore Eclipse Plugin prerequisites	4
3.2. Squore Eclipse Plugin installation	5
3.3. Squore Eclipse Plugin setup	5
3.3.1. Activation and server status section	6
3.3.2. Server connection section	6
3.3.3. Rating Configuration section	7
3.3.4. Configuration procedure	7
4. Getting Started With the Squore Eclipse Plugin	8
Index	13

Typographical Conventions

The following conventions are used in this manual.

Typeface or Symbol	Meaning
Bold	Book titles, important items, or items that can be selected including buttons and menu choices. For example: Click the Next > button to continue
<i>Italic</i>	A name of a user defined textual element. For example: Username : <i>admin</i>
Courier New	Files and directories; file extensions, computer output. For example: Edit the <code>config.xml</code> file
Courier Bold	Commands, screen messages requiring user action. For example: Username : <i>admin</i>
>	Menu choices. For example: Select File > Open . This means select the File menu, then select the Open command from it.
<...>	Generic terms. For example: <INSTALLDIR> refers to the Squore installation directory.

Notes

Screenshots displayed in this manual may differ slightly from the ones in the actual product.

Acronyms and Abbreviations

The following acronyms and abbreviations are used in this manual.

CI	Continuous Integration
CLI	Command Line Interface
DP	Data Provider, a Squore module capable of handling input from various other systems and import information into Squore
RC	Repository Connector, a Squore module capable of extracting source code from source code management systems.

1. Introduction

1.1. Foreword

This document was released by Squoring Technologies.

It is part of the user documentation of the Squore software product edited and distributed by Squoring Technologies.

1.2. About This Document

This document is the Eclipse Plugin Guide for Squore.

Its objective is to guide you through various aspects of the Squore Eclipse Plugin:

- Chapter 2, *Introducing the Squore Eclipse Plugin* positions the Squore Eclipse Plugin inside the Squore ecosystem, explaining its main features.
- Chapter 3, *Installing and setting up the Squore Eclipse Plugin* describes the steps to follow in order to install the Squore Eclipse Plugin.
- Chapter 4, *Getting Started With the Squore Eclipse Plugin* walks you through the features provided by the Squore Eclipse Plugin.


1.3. Contacting Squoring Technologies Product Support

If the information provided in this manual is erroneous or inaccurate, or if you encounter problems during your installation, contact Squoring Technologies Product Support: <http://support.squoring.com/>

You will need a valid Squore customer account to submit a support request. You can create an account on the support website if you do not have one already.

For any communication:

 **support@squoring.com**

 **Squoring Technologies Product Support**
76, allées Jean Jaurès / 31000 Toulouse - FRANCE

1.4. Responsibilities

Approval of this version of the document and any further updates are the responsibility of Squoring Technologies.

1.5. Getting the Latest Version of this Manual

The version of this manual included in your Squore installation may have been updated. If you would like to check for updated user guides, consult the Squoring Technologies documentation site to consult or download the latest Squore manuals at <http://support.squoring.com//documentation/15-A-SP2>. Manuals are constantly updated and published as soon as they are available.

2. Introducing the Squore Eclipse Plugin

2.1. Squore Eclipse Plugin in the Squore ecosystem

Squore analyzes projects (in a broad sense) to provide high level indicators and produce personalised dashboards and tailored action lists for each kind of connected user. The whole process can be automated, and integrated in a continuous integration framework.

Information generated in this fashion can be refreshed with any desired rate, and the results will be available each time the Squore process concludes a new build. However, this scenario is best suited to a daily cycle at most, where all input files are the result of a daily commit, and results are generated along with the nightly build.

The Squore Eclipse Plugin is destined to operate at a faster rate, the objective being to deliver results during the development process, before files are committed and fed to the continuous integration process. Tightening the processing loop has several consequences for the Squore Eclipse Plugin behavior:

- *It doesn't need to be connected to a central Squore server to rate a project, unless for project activation / licence validation purposes.*

By limiting communication with the server to a minimum, we make sure projects can evolve outside the global rating cycle, but still offer the rating service.

Should the server be unreachable, rating service is still available for a whole week, before being deactivated. Once a connection is available again, and the credentials are validated, the rating capabilities are restored.

- *It must be able to process and rate a project locally*

The Squore Eclipse Plugin embeds the Squore rating module, much like Squore CLI.

This allows us to retrieve data from Data Providers, assemble them, and launch the Squore Analysis and Decision Engines.

Since the core module used for these phases are the exact same one used by Squore Server, the results are identical.

- *It must use a configuration (in terms of Analysis Model and Data Providers) synchronised with the one installed on Squore Server*

Since the Squore Eclipse Plugin uses a Squore rating module, it needs a rating environment, which must be synchronised with the server's, to ensure results equivalence.

The Squore Eclipse Plugin will perform such synchronisation during the product activation, which is a feature available in the Squore Eclipse Plugin preferences page.

- *It must be able to manage projects using other languages than Java (e.g. C, C++, C# Ada)*

The Squore Eclipse Plugin benefits from the Squore source code analyzer, which provides support for several languages (see Squore documentation for more details).

Analysis results will be available at folder and file level.

- *It must be integrated into the Eclipse IDE, taking advantage of its extendability*

The Eclipse plugin development framework offers a number of mechanisms to enhance various parts of the IDE. The added features have been designed to be both unobtrusive and informative.

- *It must enhance a project's available information by displaying ratings, Action Items, Findings for each compatible object*

The added information is assigned to each artefact, which means at project, file, module, class and function level. Each of these elements will have a rating, as well as a list of Action Items and Findings, depending on the model. A search view offering project artefacts sorted by rating is also available.

→ *It must be able to manage local versions, which are dissociated from the source code control versioning mechanism*

One of the differences between the Squore Eclipse Plugin and Squore is that Squore Server uses a database to store the project builds across multiple versions, whereas the Squore Eclipse Plugin uses a local file storage mechanism, to maintain a local history.

This allows functions such as `DELTA_VALUE` or `PREVIOUS_VALUE` to work as expected, and produce trend analysis indicators.

The data used to produce that trend are these of the current version, and the last rated version.

2.2. Who is the Squore Eclipse Plugin destined to?

As hinted by its name, the Squore Eclipse Plugin is a developer tool. Its objective is to provide the user with enough information to help him:

- Process the generated action items list
- Check the evolution of the source code rating as it is being produced
- Find what artefacts are responsible for a poor rating, and understand why
- Anticipate and address problems that will appear in the project's daily Squore server build

As the Squore Eclipse Plugin seamlessly integrates Squore results in the IDE, and uses the same model as the one on Squore Server, the developer will gain visibility over the evaluation process, and improve his development practices to match the standard defined by the model.

The integrated rating feedback also gives the developer the opportunity to actively participate to the global effort on quality improvement.

3. Installing and setting up the Squore Eclipse Plugin

3.1. Squore Eclipse Plugin prerequisites

Just like Squore CLI, the Squore Eclipse Plugin is based on a modular architecture using scripts. These scripts being written in Perl and Tcl, the system running the Squore Eclipse Plugin must contain installed and accessible versions of:

→ **Perl** version 5.10.1 or greater including the following extra-modules:

→ Mandatory packages:

- **Archive::Zip**
- **Algorithm::Diff**
- **DBI**
- **DBD::Pg**
- **Digest::SHA**
- **HTTP::Request**
- **JSON**
- **LWP::UserAgent**
- **LWP::Protocol::https**
- **Time::HiRes**
- **XML::Parser**

→ Optional packages for working with Microsoft Excel:

- **Spreadsheet::XLSX**
- **Spreadsheet::WriteExcel**
- **Spreadsheet::ParseExcel**
- **Spreadsheet::BasicRead**
- **OLE::Storage_Lite::PPS**
- **Digest::Perl::MD5**
- **Crypt::RC4**

→ Optional packages for working with OSLC systems:

- **HTML::TreeBuilder**
- **HTTP::Server::Simple**
- **HTTP::Server::Simple::CGI**
- **WWW::Mechanize**
- **Date::Parse**
- **Date::Language**
- **Date::Format**
- **Time::Zone**

- Tcd version 8.5 or greater

3.2. Squore Eclipse Plugin installation

The Squore Eclipse Plugin complies with the Eclipse requirements, and is an OS-independent package to be deployed in the Eclipse plugins directory. Installation is done through an update site.

For a successful installation of Squore Eclipse Plugin, you will need:

- An operational Eclipse IDE. Currently supported versions are:
 - 3.7 32 bits (Indigo)
 - 3.7 64 bits (Indigo)
 - 4.2 32 bits (Juno)
 - 4.2 64 bits (Juno)
 - 4.3 32 bits (Kepler)
 - 4.3 64 bits (Kepler)

- At least 1 GB of RAM

This amount of required memory will depend on what external tools are connected to the Squore Eclipse Plugin, and does not take into account any memory used by other installed plugins. Please adjust your Eclipse memory settings accordingly.

- Create a new update site within Eclipse
 - Select **Help > Install new software;**
 - Click on **Available Software Sites;**
 - Add a new site;
 - Give it a name (for example: "Squore Eclipse Plugin");
 - Set the location as follows:

```
http://eclipse.squoring.com
```

- Go back to the "**Install**" window with the "**OK**" buttons;
- Select the newly created update site;
- Proceed with the plugin installation as with any Eclipse plugin.

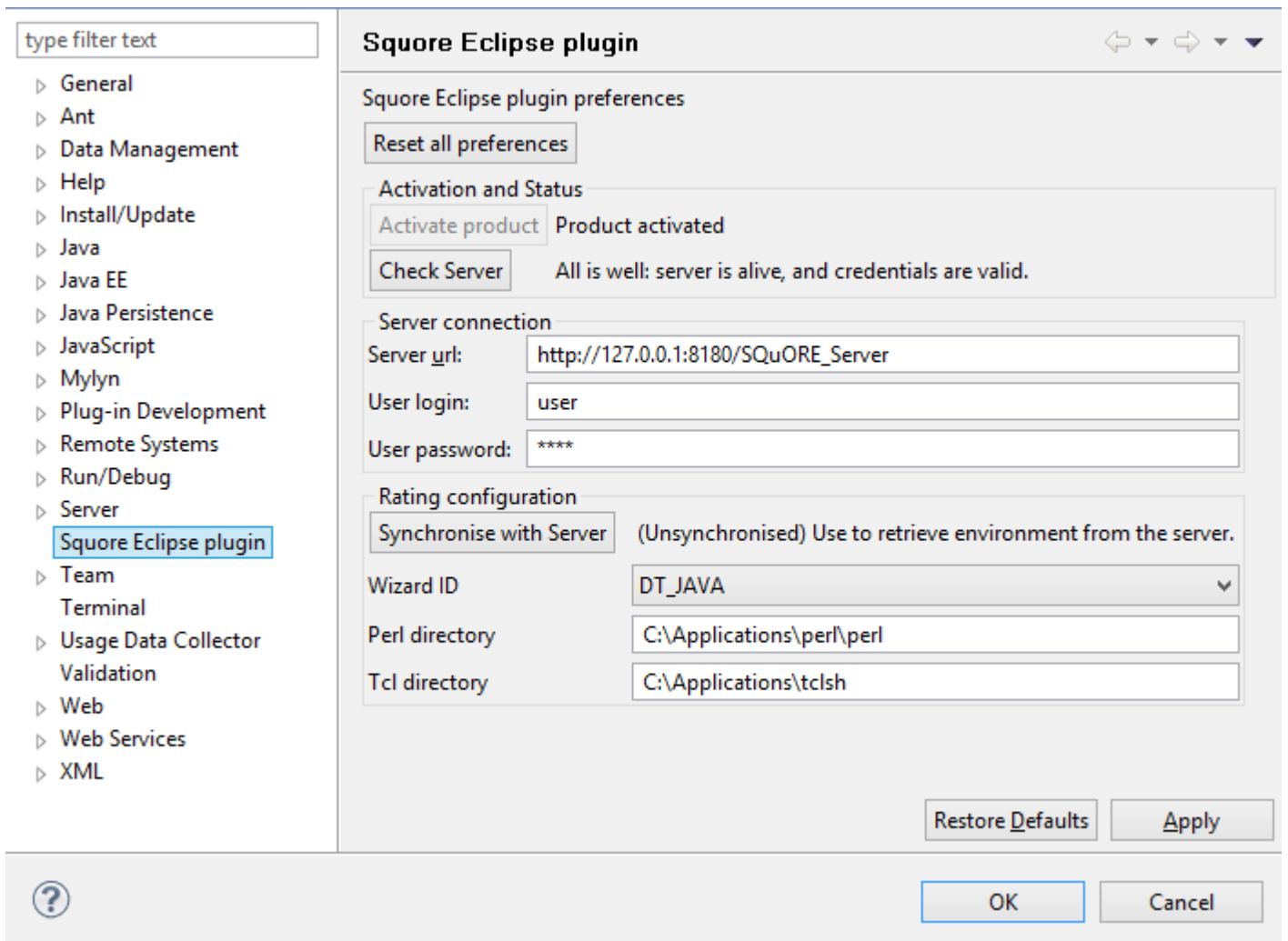
Notes:

- If the Squore Eclipse Plugin does not appear, please uncheck "Group items by category".
- To speed-up installation procedure, it is possible to uncheck "Contact all update sites during install to find required software", as no external software is needed by the Squore Eclipse Plugin.
- Should the plugin fail to deploy, Eclipse will notify the user of the problem cause, and will start its operations as usual. If necessary, copy and send the generated Eclipse log to support@squoring.com.

3.3. Squore Eclipse Plugin setup

To use the Squore Eclipse Plugin, it must be activated, the server must be reachable with proper credentials, and rating preferences must be provided.

All these operations are accessible through the Windows>Preferences>Squore Eclipse Plugin page. Note that all provided information uses the Eclipse secure storage.



The Squore Eclipse Plugin preferences page

3.3.1. Activation and server status section

This section serves two purposes: provide status informations on the Squore Eclipse Plugin and the associated Squore server.

- The **Activate product** button launches a product activation on the server, using the provided credentials;
- The **Check server** button launches a server reachability check, and display the returned status.

3.3.2. Server connection section

This section contains all data required to connect to a server.

- The server URL;
- The user login;
- The user password.

3.3.3. Rating Configuration section

The reference rating environment is the one used by the server. The Squore Eclipse Plugin must work with the same environment. This section allows the user to synchronize with the Squore server, and specify the Squore Eclipse Plugin rating parameters.

- The **Synchronise with server** button launches a full synchronization with the server's rating environment. This can be a lengthy operation (depending on the server configuration), and the Eclipse status bar will inform you of its progress. You can close the preferences page during that operation.
- A reference to Perl must be provided in this section. Perl being a prerequisite for Squore Eclipse Plugin, the location of the perl binary must be defined here. Typical values are:
 - `/usr/bin/perl` on Unix
 - `C:\Applications\perl\perl` on Windows
- A reference to Tcl must be provided in this section. Tcl being a prerequisite for Squore Eclipse Plugin, the location of the Tcl binary must be defined here. Typical values are:
 - `/usr/bin/tclsh` on Unix
 - `C:\Applications\tclsh` on Windows

3.3.4. Configuration procedure

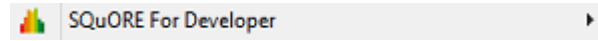
To successfully activate the Squore Eclipse Plugin, the following steps must be taken:

- Provide the URL of a running server in the **Server url** field;
- Provide valid credentials using the **User login** and **User password** fields;
- Click on the **Apply** button;
- Click on the **Check server** button. The associated text will display a status of the connection with the server;
- Click on the **Activate product** button. The associated text should change to "Product activated";
- Click on the **Synchronize with server** button;
- Once the synchronisation is complete, select the Wizard with which the rating will be performed;
- Set the Perl directory;
- Set the Tcl directory.

4. Getting Started With the Squore Eclipse Plugin

Once the Squore Eclipse Plugin is installed in Eclipse, obtaining a Java project rating is rather straightforward.

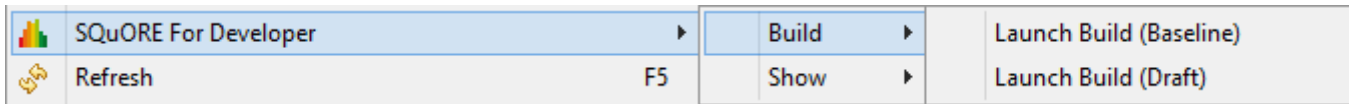
First locate the contextual menu entry for the Java project you want to rate. This menu is only available on Java Projects.



Contextual menu on a Java project

Expanding this menu will offer you rating actions and display options.

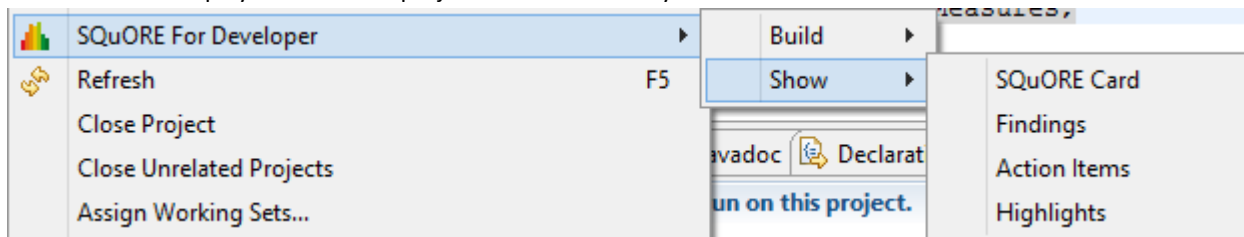
Rating (or "building") a project with Squore Eclipse Plugin can either be performed as a baseline build or a draft build.



Rating actions

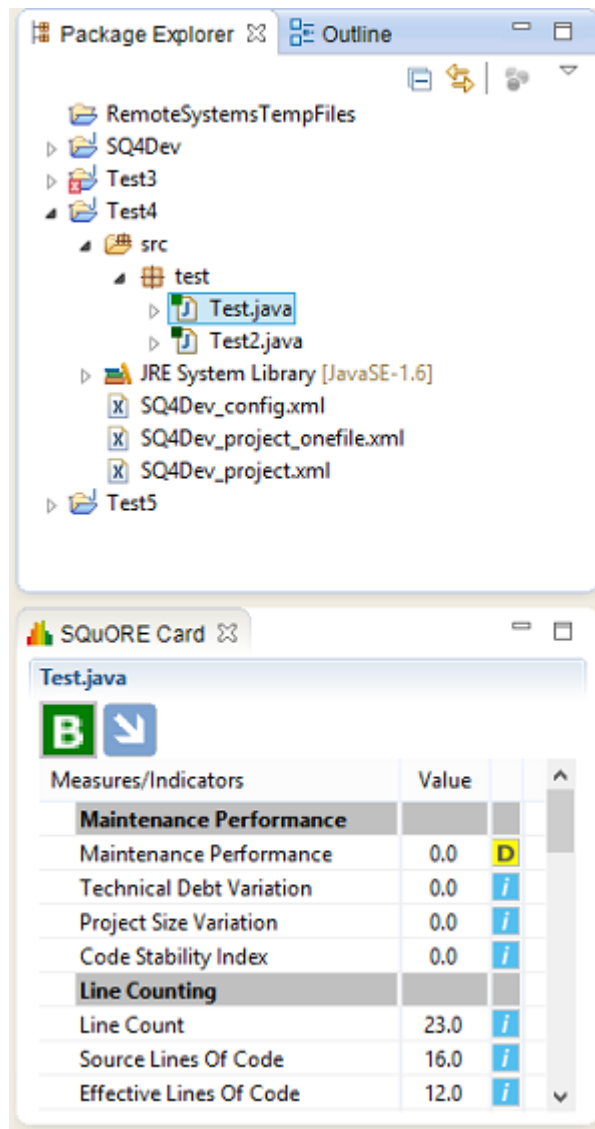
- When choosing baseline build, the project rating will produce a new reference version, with which all subsequent ratings will be compared to. This will affect rating computations based on metrics previous values, as well as all trends displayed in the Squore Eclipse Plugin views.
The baseline mode is useful when a stable state of the project has been reached, and a development phase is about to begin.
- When choosing draft build, the project is rated as usual, but the reference version is not modified. Multiple draft builds will always compare to the last defined baseline.
The draft mode is useful during a development phase, when the project can undergo several back and forth changes, and the user wants to keep comparing the current version to the last stable one.

The "Show" sub-menu will allow selection of several views providing rating-related information. Each one of these views displays data for the project artefact currently selected.

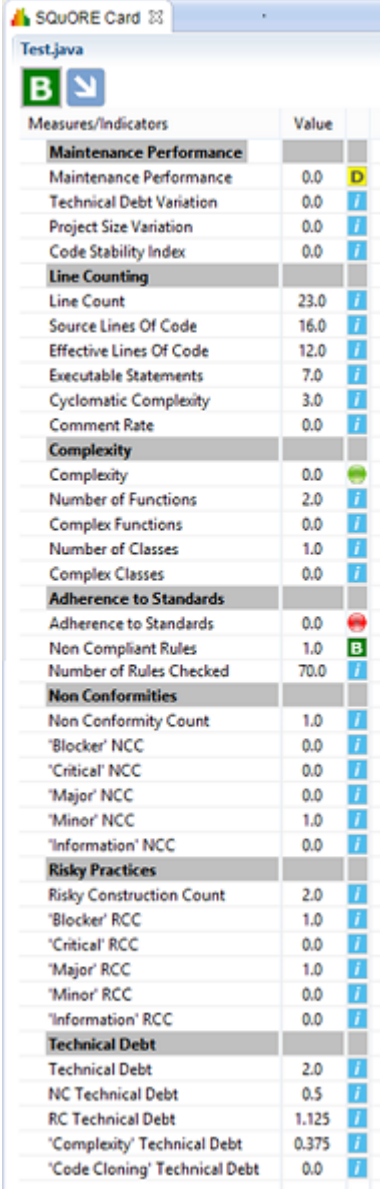


Display options

- The Squore Card view displays the overall rating for the current artefact, and its trend. It also lists the tables of interests for this artefact, as defined in the Analysis Model used to rate the project.



Rating card

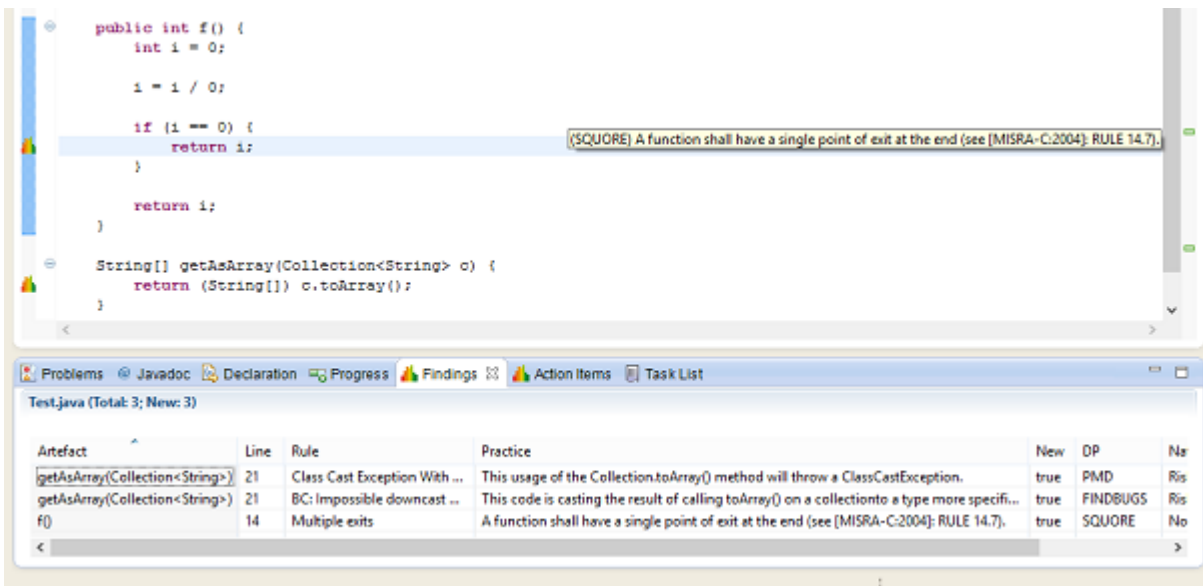


Measures/Indicators	Value	
Maintenance Performance		
Maintenance Performance	0.0	D
Technical Debt Variation	0.0	I
Project Size Variation	0.0	I
Code Stability Index	0.0	I
Line Counting		
Line Count	23.0	I
Source Lines Of Code	16.0	I
Executable Lines Of Code	12.0	I
Executable Statements	7.0	I
Cyclomatic Complexity	3.0	I
Comment Rate	0.0	I
Complexity		
Complexity	0.0	G
Number of Functions	2.0	I
Complex Functions	0.0	I
Number of Classes	1.0	I
Complex Classes	0.0	I
Adherence to Standards		
Adherence to Standards	0.0	R
Non Compliant Rules	1.0	B
Number of Rules Checked	70.0	I
Non Conformities		
Non Conformity Count	1.0	I
'Blocker' NCC	0.0	I
'Critical' NCC	0.0	I
'Major' NCC	0.0	I
'Minor' NCC	1.0	I
'Information' NCC	0.0	I
Risky Practices		
Risky Construction Count	2.0	I
'Blocker' RCC	1.0	I
'Critical' RCC	0.0	I
'Major' RCC	1.0	I
'Minor' RCC	0.0	I
'Information' RCC	0.0	I
Technical Debt		
Technical Debt	2.0	I
NC Technical Debt	0.5	I
RC Technical Debt	1.125	I
'Complexity' Technical Debt	0.375	I
'Code Cloning' Technical Debt	0.0	I

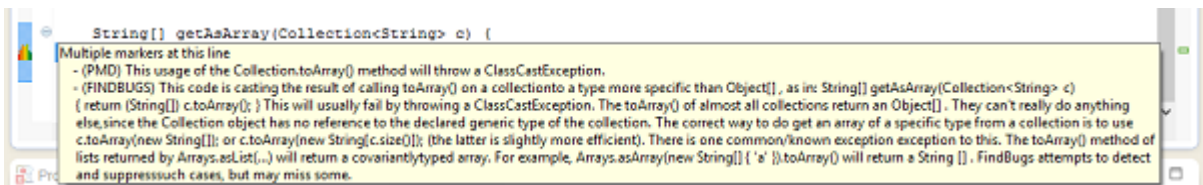
Full Rating card

→ The Findings view

lists the Findings for the current artefact as well as underlying artefacts, with their location and definition details. This list can be sorted, and allows navigation to artefacts associated to each finding by double-clicking on their name.



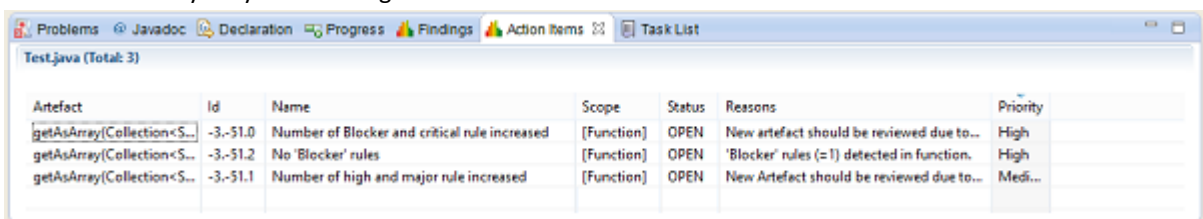
A finding and its location in the source code



A finding full description

→ The Action Items view

works much like the Findings one, except it lists the Action Items computed by the Decision Model, and the reason why they have been generated.



Action Items

→ The Highlights view

works much like the Findings one, except it provides a list of the current project's Files, Classes or Functions, sorted by decreasing rating. Selecting the type of object to sort (File, Class or Function) is performed through the top-left menu icons.

Problems @ Javadoc Declaration Console Highlights Findings

Test worst methods (Total: 171)

Artefact	Line
doXMLFinding	C 310
endElement	C 570
getLocStr	C 58
startElement	C 549
CoreArtefact	B 76
addData	B 179
addDefectReports	B 115
addDefectReports	B 219
addDrLinksTestsDependencies	B 551
addInformation	B 489
addLocation	B 56
addLogicalFatherStr	B 482
addSon	B 428

Highlights

Activating and anchoring these views to the workspace will enhance the project with global rating data, as well as useful information to help understand and ameliorate that rating.

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project structure with 'Test.java' selected. The main editor displays the code for 'Test.java', with a line of code highlighted in blue. The SquORE Card at the bottom left shows a 'B' rating and various metrics. The Problems view at the bottom right shows a table of issues.

Artefact	Line	Rule	Practice	New	DP	Nature	Remedi...	Severity
getArray(Collection<String>)	21	BC: Impossible downcast...	This code is casting the result of call...	true	FINDEUS	Risky C...	Medium	Block...
getArray(Collection<String>)	21	Class Cast Exception With ...	This usage of the Collection.toArray()...	true	PMD	Risky C...	Little	Major
f()	14	Multiple exits	A function shall have a single point of...	true	SQUARE	Non C...	Low	Warni...

Main view

Index

A

Action Items view, 11

B

Baseline build, 8

C

Configuration procedure, 7

D

Draft build, 8

F

Findings view, 10

H

Highlights view, 11

M

Memory, 5

P

Prerequisites, 5

Product activation, 6

R

Rating Configuration parameters, 7

S

Server connection parameters, 6

Server status, 6

Squore Card view, 8