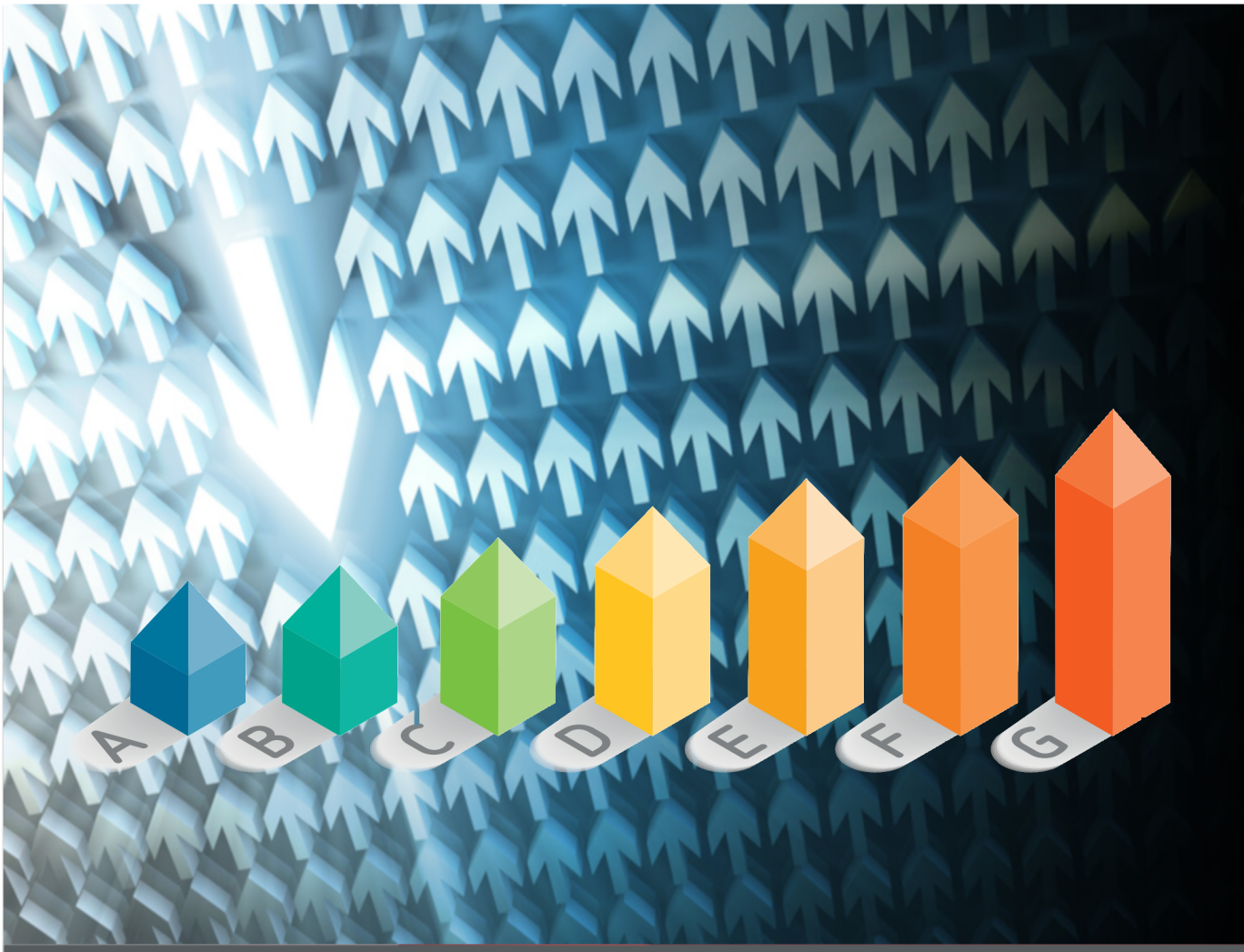
VECTOR



Eclipse Plugin Guide

Squore 19.1.18

Last updated 2021-07-08

Table of Contents

Preface . . . 1
- Foreword . . . 1
- Licence . . . 1
- Warranty . . . 1
- Responsabilities . . . 2
- Contacting Vector Informatik GmbH Product Support . . . 2
- Getting the Latest Version of this Manual . . . 2

1. Introduction . . . 3
2. Introducing the Squore Eclipse Plugin . . . 4
- Squore Eclipse Plugin in the Squore ecosystem . . . 4
- Who is the Squore Eclipse Plugin destined to? . . . 5

3. Installing and setting up the Squore Eclipse Plugin . . . 6
- Installation Prerequisites . . . 6
  - Supported Operating Systems . . . 6
  - For All Systems . . . 6
  - Packages for Windows . . . 7
  - Packages for CentOS and Red Hat Enterprise Linux . . . 7
  - Packages for Ubuntu . . . 7
  - Packages for other Linux distributions . . . 8
- Squore Eclipse Plugin installation . . . 8
- Squore Eclipse Plugin setup . . . 9

4. Getting Started With the Squore Eclipse Plugin . . . 10
- Launching Your First Analysis . . . 10
- Visualising Analysis Results . . . 12
- Exporting Analysis Results . . . 15

Index . . . 18

Preface

© 2021 Vector Informatik GmbH - All rights reserved - https://www.vector.com/ - This material may not be reproduced, displayed, modified or distributed without the express prior written permission of the copyright holder. Squore is protected by an Interdeposit Certification registered with Agence pour la Protection des Programmes under the Inter Deposit Digital Number IDDN.FR.001.390035.001.S.P.2013.000.10600.

Foreword

This edition of the Eclipse Plugin Guide was released by Vector Informatik GmbH.

It is part of the user documentation of the Squore software product edited and distributed by Vector Informatik GmbH.

For information on how to use and configure Squore, the full suite of manuals includes:

User Manual	Target Audience
Squore Installation Checklist	New users before their first installation
Squore Installation and Administration Guide	IT personnel and Squore administrators
Squore Getting Started Guide	End users, new users wanting to discover Squore features
Squore Command Line Interface	Continuous Integration Managers
Squore Configuration Guide	Squore configuration maintainers, Quality Assurance personnel
Squore Eclipse Plugin Guide	Eclipse IDE users
Squore Reference Manual	End Users, Squore configuration maintainers
Squore API Guide	End Users, Continuous Integration Managers
Squore Software Analytics Handbook	End Users, Quality Assurance personnel

You can also use the online help from any page when using the Squore web interface by clicking **? > Help**.

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, Vector Informatik GmbH. Vector Informatik GmbH 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 Vector Informatik GmbH to determine whether any such changes have been made. The terms and conditions governing the licensing of Vector Informatik GmbH software consist solely of those set forth in the written contracts between Vector Informatik GmbH and its customers. All third-party products are trademarks or registered trademarks of their respective companies.

Warranty

Vector Informatik GmbH 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. Vector Informatik GmbH 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.

This edition of the Eclipse Plugin Guide applies to Squore 19.1.18 and to all subsequent releases and modifications until otherwise indicated in new editions.

Responsabilities

Approval of this version of the document and any further updates are the responsibility of Vector Informatik GmbH.

Contacting Vector Informatik GmbH Product Support

If the information provided in this manual is erroneous or inaccurate, or if you encounter problems during your installation, contact Vector Informatik GmbH Product Support: https://support.squoring.com/

You will need a valid 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**
- **Vector Informatik GmbH Product Support**

  Squoring Technologies - 9/11 rue Matabiau / 31000 Toulouse - FRANCE

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 Vector Informatik GmbH documentation site to consult or download the latest Squore manuals at https://support.squoring.com/documentation/latest. Manuals are constantly updated and published as soon as they are available.

Chapter 1. Introduction

This document is the Eclipse Plugin Guide for Squore.

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

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

Chapter 2. Introducing the Squore Eclipse Plugin

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 runs the Squan Sources static code analyser on the sources in your Eclipse project.

  Note that no other data provider is run by the Squore Eclipse Plugin, so the overall rating of your project may vary from the one obtained when analysing the same for the same code base on Squore Server. Metrics and violations returned by Squan Sources will nevertheless be the same.

- *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 and highlights 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.

  Trends are based on the evolution of indicators between a baseline analysis and the last-rated analysis.

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 users with enough information to help them:

- Process the generated list of Action Items
- 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 before the code is referenced in a SCM system

As the Squore Eclipse Plugin seamlessly integrates Squan Sources results in the IDE, and uses the same model as the one on Squore Server, developers will gain visibility over the evaluation process, and improve their 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.

Chapter 3. Installing and setting up the Squore Eclipse Plugin

Installation Prerequisites

Supported Operating Systems

The following is a list of the officially supported and tested operating systems:

A 64-bit version of the OS is required

- CentOS 6.7
- CentOS 7.3
- CentOS 8
- Ubuntu 18.04 LTS
- Ubuntu 20.04 LTS
- Windows 10
- Windows Server 2016

The following is a list of the operating systems that are not regularly tested but are known to be working:

- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 7
- Fedora 29
- SuSe Linux 11.1
- Ubuntu Server 16.04
- Windows 7
- Windows 8
- Windows Server 2008 R2
- Windows Server 2012 R2

For All Systems

For a successful installation of Squore, you will need:

- A Java Runtime Environment version 8 (64-bits) (other versions are not supported)
- A supported Eclipse version (Luna SR2 and up)
- Valid credentials to log into a Squore Server
- At least 4 GB of space available on the disk for a full installation with demo projects
- At least 8 GB of RAM on the server machine
- At least 4 GB of RAM on the client machine

Keep in mind that the requirements above are the strict minimum. In production, Squore Server generally runs on a dedicated machine. A performant configuration is usually:

- 16 threads CPU.
- 64GB of RAM.
- SSD hard drives.

Squore reserves 25% of the available RAM of the machine to the database and another 25% to the server. External processes (like Checkstyle or FindBugs) running on the same machine as Squore may add to the amount of RAM required for analysing source code. Linux is known to offer better performances than Windows when running Squore. For a production database, you should plan a minimum of 20 GB of disk space.

Packages for Windows

A compatible version of Eclipse must be downloaded from http://www.eclipse.org/downloads/ .

Packages for CentOS and Red Hat Enterprise Linux

On Red Hat Enterprise Linux and CentOS (6.5, 7.1 and 8), the dependencies are satisfied by the following packages:

Mandatory packages:

- **java-1.8.0-openjdk**
- **perl**
- **perl-Date-Calc**
- **perl-Digest-SHA**
- **perl-JSON**
- **perl-libwww-perl**
- **perl-Time-HiRes**
- **perl-XML-Parser**
- **fontconfig**
- **tcl**
- A compatible installation of Eclipse downloaded from http://eclipse.org/downloads .

For more information about how to install the Extra Packages for Enterprise Linux (EPEL) repository, consult https://fedoraproject.org/wiki/EPEL.

Packages for Ubuntu

On Ubuntu 16.04 LTS, 18.04 LTS and 20.04 LTS, the dependencies are satisfied by the following packages:

Mandatory packages:

- **libdate-calc-perl**
- **libhttp-message-perl**
- **libjson-perl**
- **libwww-perl**
- **libxml-parser-perl**

- **openjdk-8-jre**
- **perl**
- **tcl**
- A compatible installation of Eclipse downloaded from http://eclipse.org/downloads

Packages for other Linux distributions

On Linux platforms, the following must be installed before installing Squore:

- **Perl** version 5.10.1 or greater including the following extra-modules:
  - Mandatory packages:
    - **Date::Calc** [module details]
    - **Digest::SHA** [module details]
    - **HTTP::Request** [module details]
    - **JSON** [module details]
    - **LWP** [module details]
    - **LWP::UserAgent** [module details]
    - **Time::HiRes** [module details]
    - **XML::Parser** [module details]

If some of these modules are not available as packages on your operating system, use your perl installation's cpan to install the modules. Using the OS packages is recommended, as it avoids having to reinstall via cpan after upgrading your version of perl.

- **Tcl** version 8.5 or greater,
- A supported Eclipse version (Luna SR2 and up)

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.

In order to launch analyses, the plugin needs to use an installation of perl and tclsh. If you are using a Windows operating system, the easiest way to install both dependencies is to install the Squore CLI package on your machine following the procedure outlined in the Command Line Interface manual.

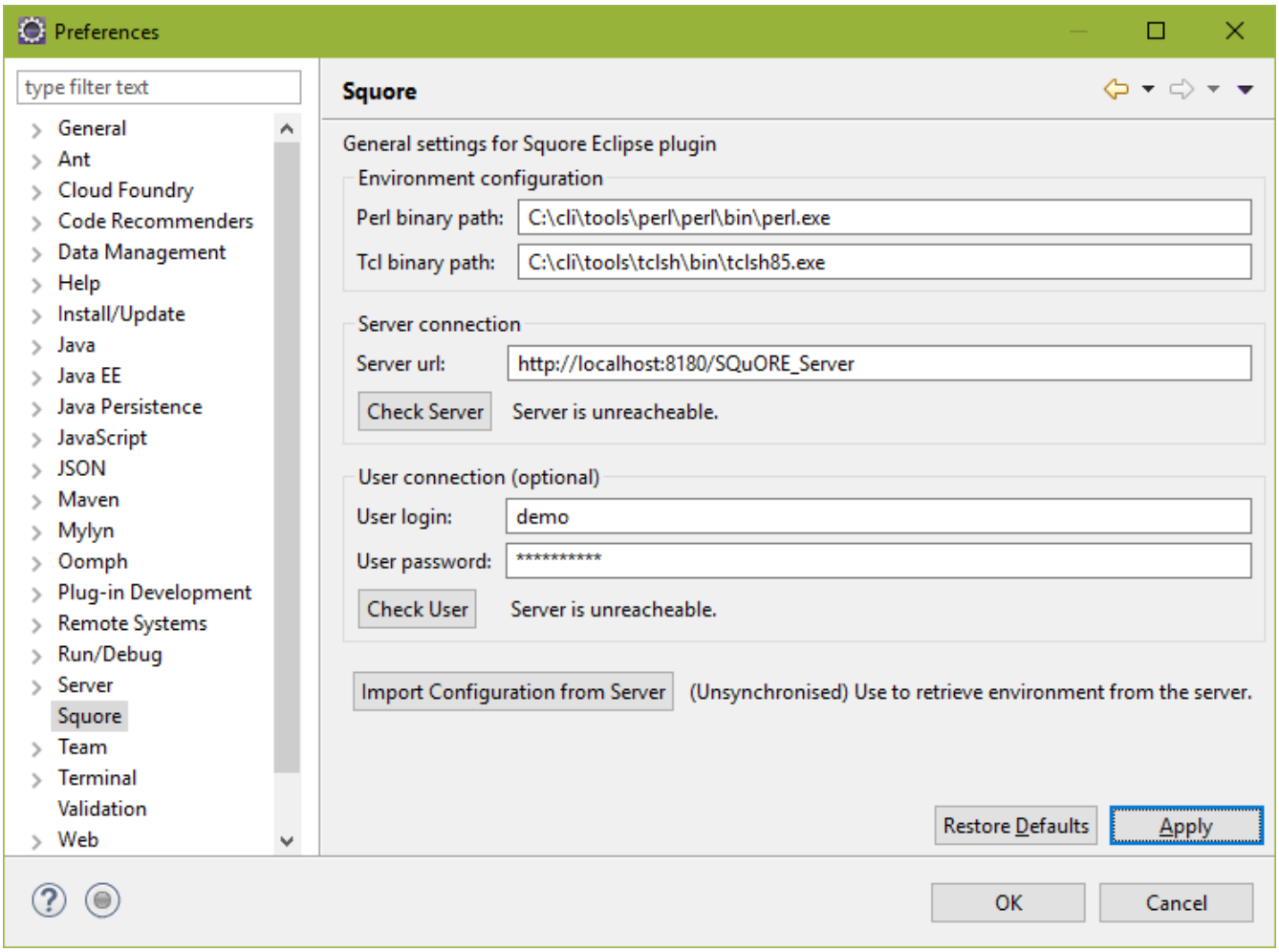
- Create a new update site within Eclipse
  - Select **Help > Install new software;**.
  - Click on **Available Software Sites;**.
  - Add a new site;
  - Give it a name (for example: "Squore Eclipse Plugin");
  - Set the location to **http://localhost:8180/SQuORE_Server/eclipse**
- Go back to the **"Install"** window with the **"OK"** buttons;
- Select the newly created update site;
- Proceed with the plugin installation as with any Eclipse plugin.

- 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.

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 **Window > Preferences > Squore** page. Note that all provided information uses the Eclipse secure storage.



The Squore Eclipse Plugin preferences page

Fill in the paths to the perl and tclsh executables and the server details and click **Check Server** to ensure that your configuration is correct.

If you can successfully connect to the server, click **Import Configuration from Server** to synchronise your environment with the one from Squore Server.

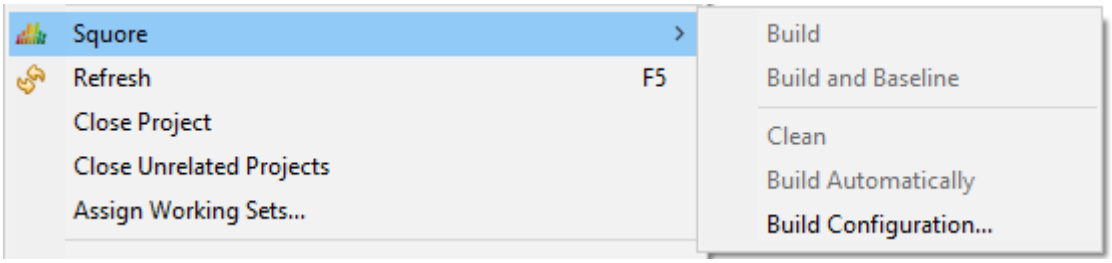
When the synchronisation finishes, click **OK** to close the preferences window.

Chapter 4. Getting Started With the Squore Eclipse Plugin

Launching Your First Analysis

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

First locate the Squore contextual menu entry for the project you want to rate.



Contextual menu on an Eclipse project

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

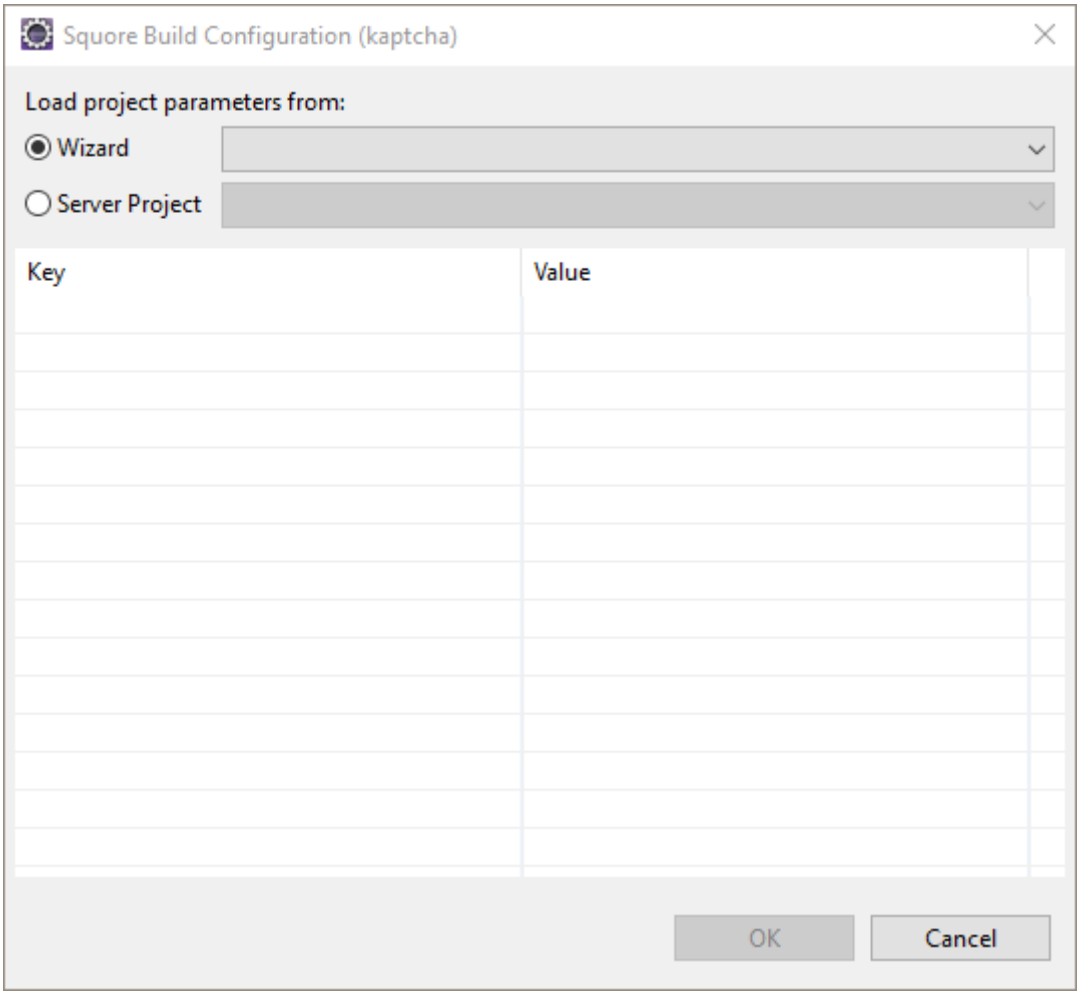
- When choosing baseline build, the project rating will produce a new reference version, with which all subsequent ratings will be compared to. This will affect rating computations based on metrics previous values, as well as all trends displayed in the Squore Eclipse Plugin views.

  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. Each new draft build overwrites the previous analysis and displays trends and deltas relative to the last defined baseline.

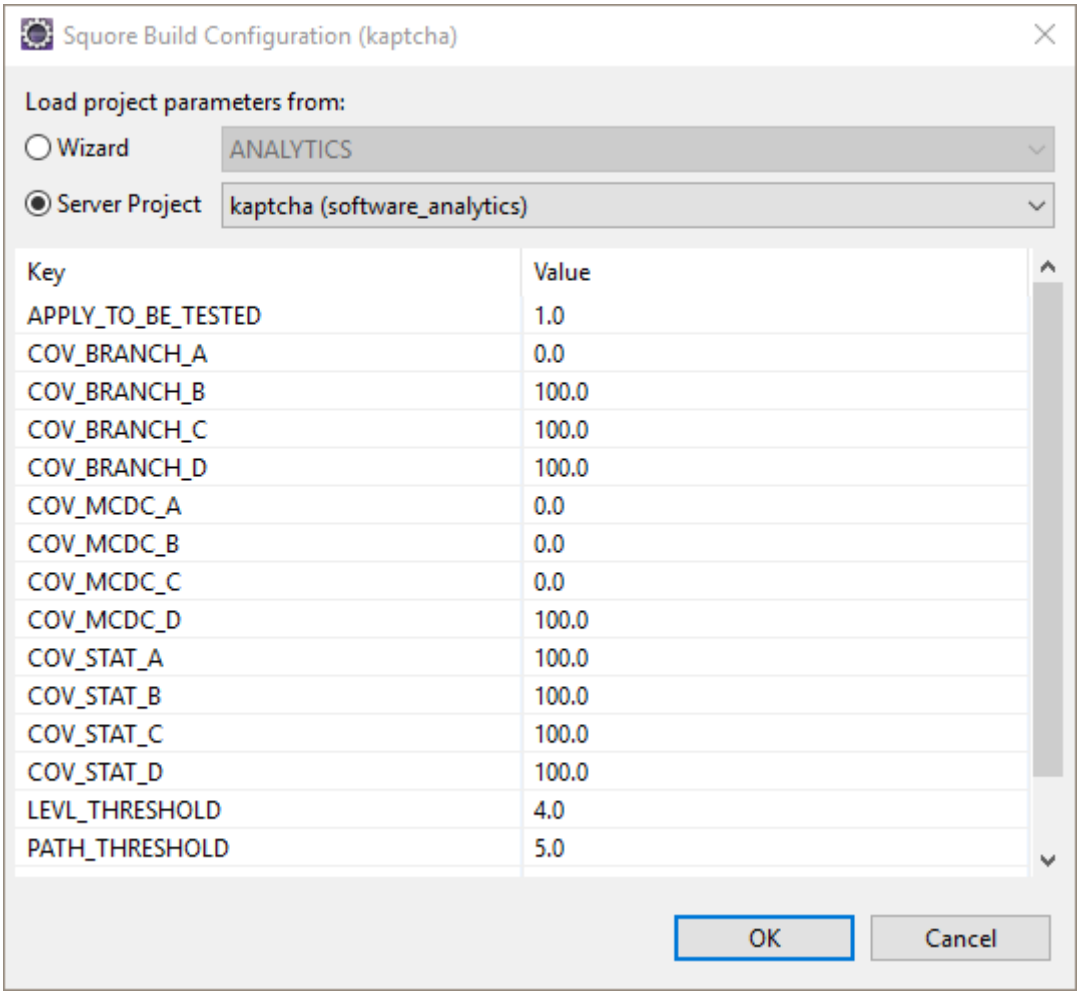
  The draft mode is useful during a development phase, when the project can undergo several back and forth changes, and the user wants to keep comparing the current version to the last stable one.

Before you can build a project, you first need to define its build configuration:



The Build Configuration screen

On the Build configuration screen, you can choose an analysis model for your project, or you can select the project on the server that corresponds to the code you are working on locally. By selecting an existing project, you can retrieve the project attributes from the server, and therefore run your local analysis in the same conditions as it is run on the server.



Project options for the kaptcha project on the server

Click **OK** to close the configuration dialogue. You are now ready to start building your project using the draft or baseline submenu in the Squore Eclipse Plugin context menu.

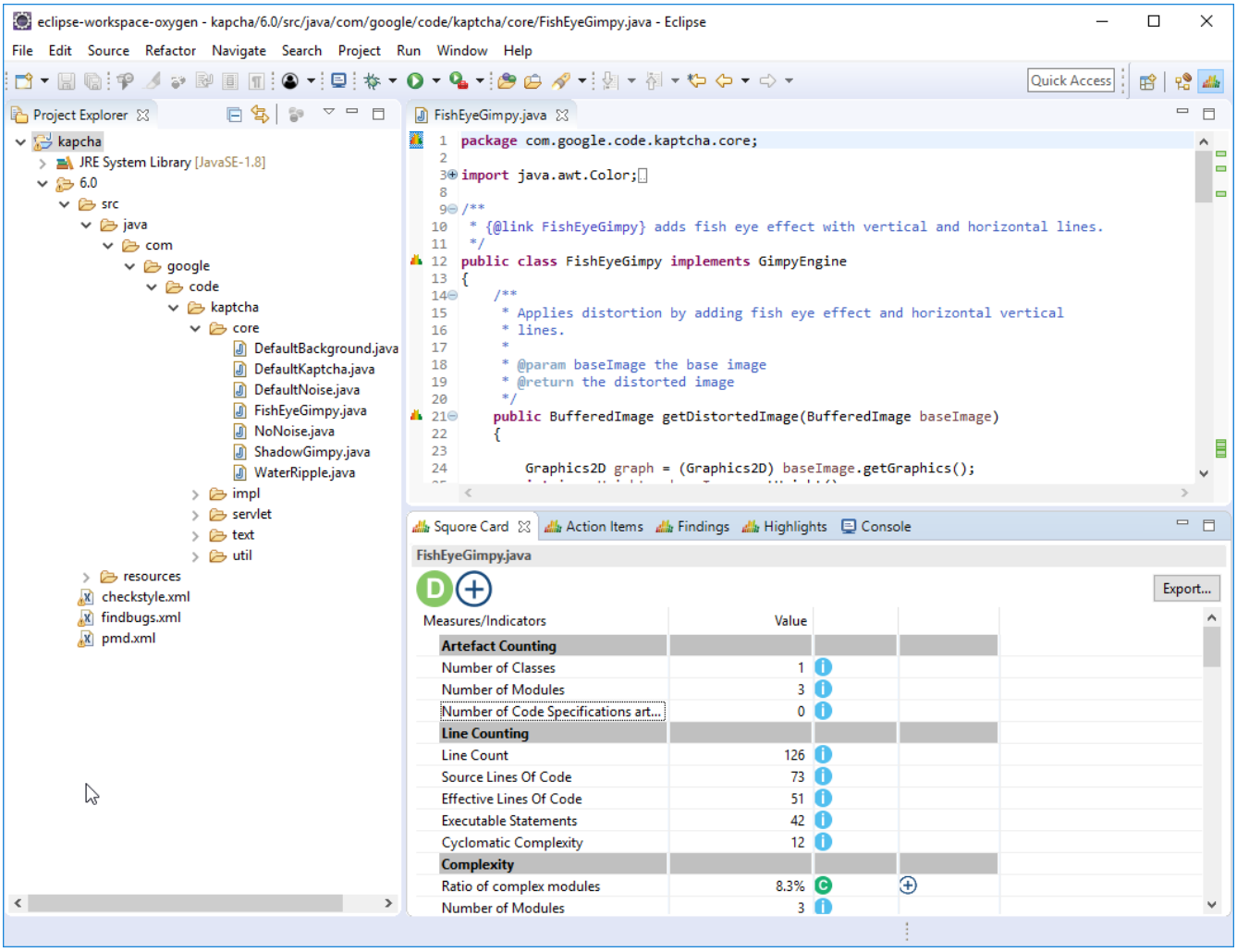
With the Squore Eclipse Plugin, in addition to projects, you can build folders, sub-folders and individual files as well. Go to the Squore contextual menu of the desired item (right click) and click build, just like a project.

Visualising Analysis Results

After your analysis completes, switch to the Squore perspective to view the results.

Click **Window > Perspective > Open Perspective > Other** and search for **Squore** in the list of available perspectives if you are using it for the first time.

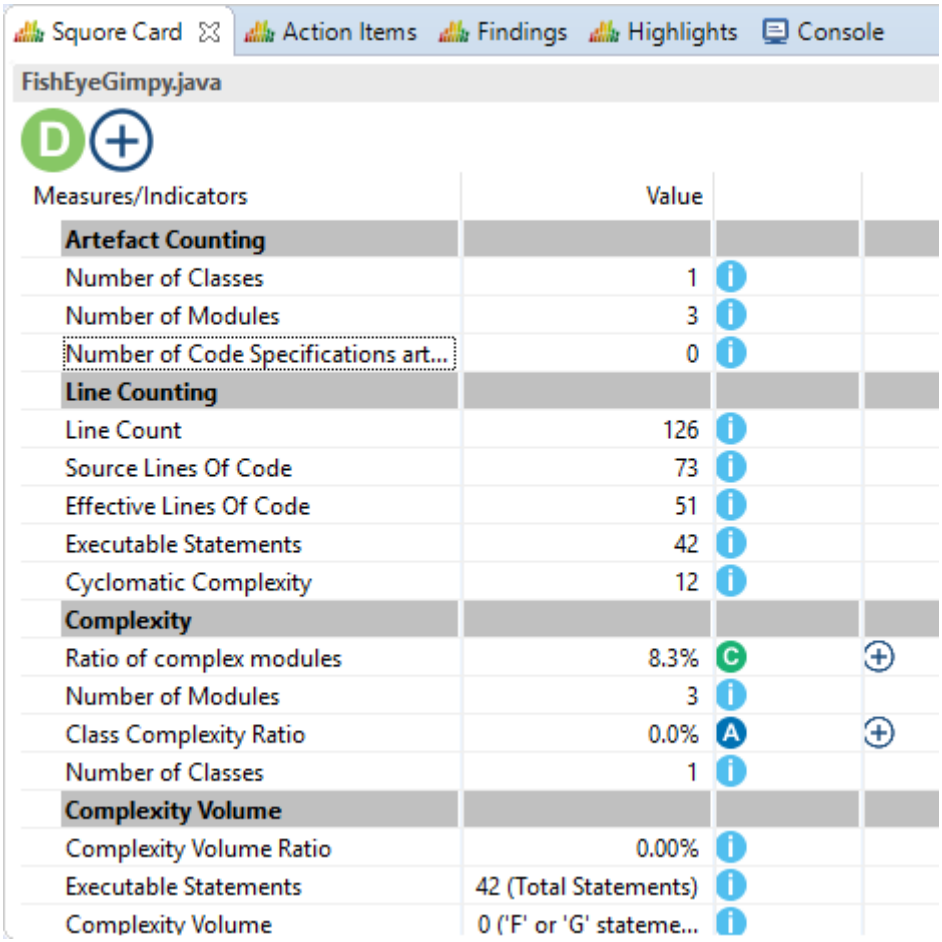
The Squore perspective displays various views with information relative to the file or folder you click in the Project Explorer.



The Squore Perspective with the Squore Card, Action items, Findings and Highlights views in the bottom panel

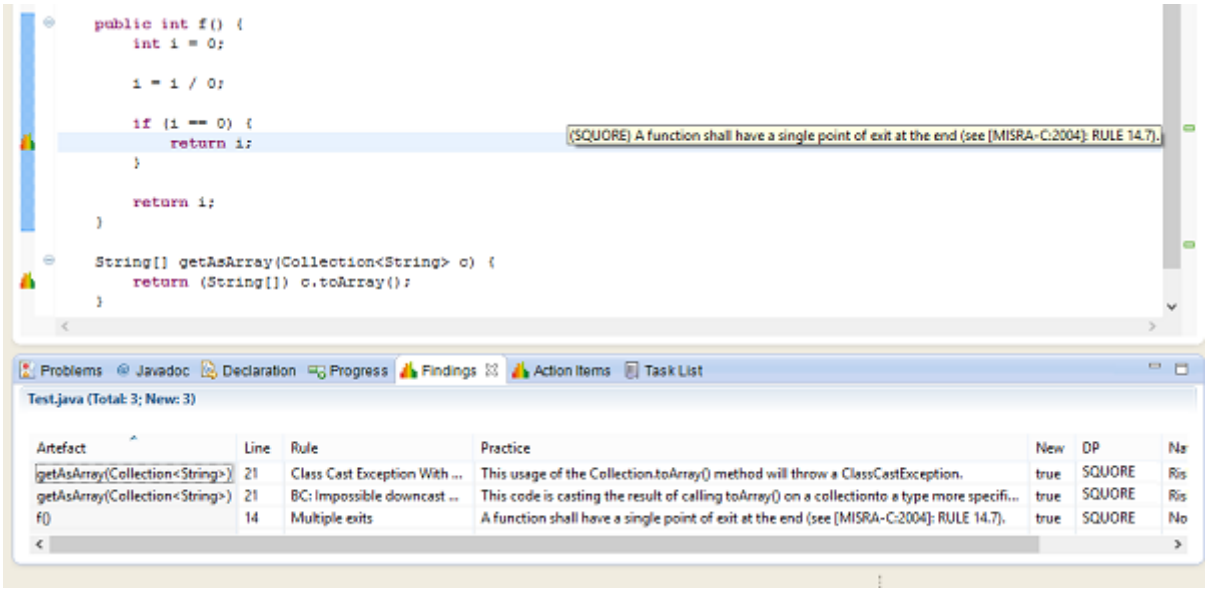
Click a file or folder in the Project Explorer and use one of the following views in the bottom panel to understand your rating:

- 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.

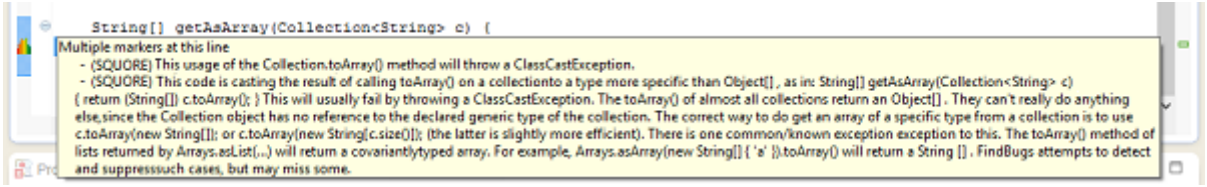


The Squore card

- 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.

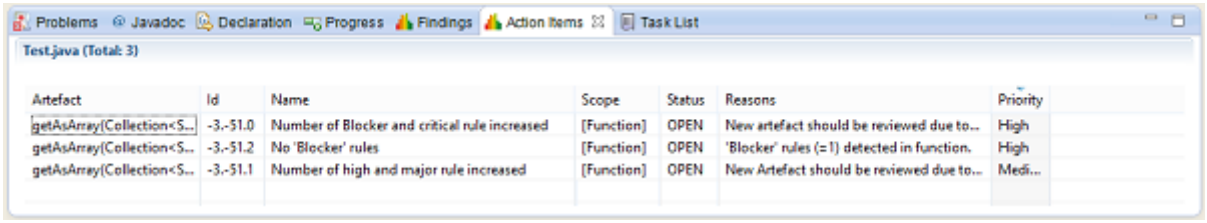


A finding and its location in the source code



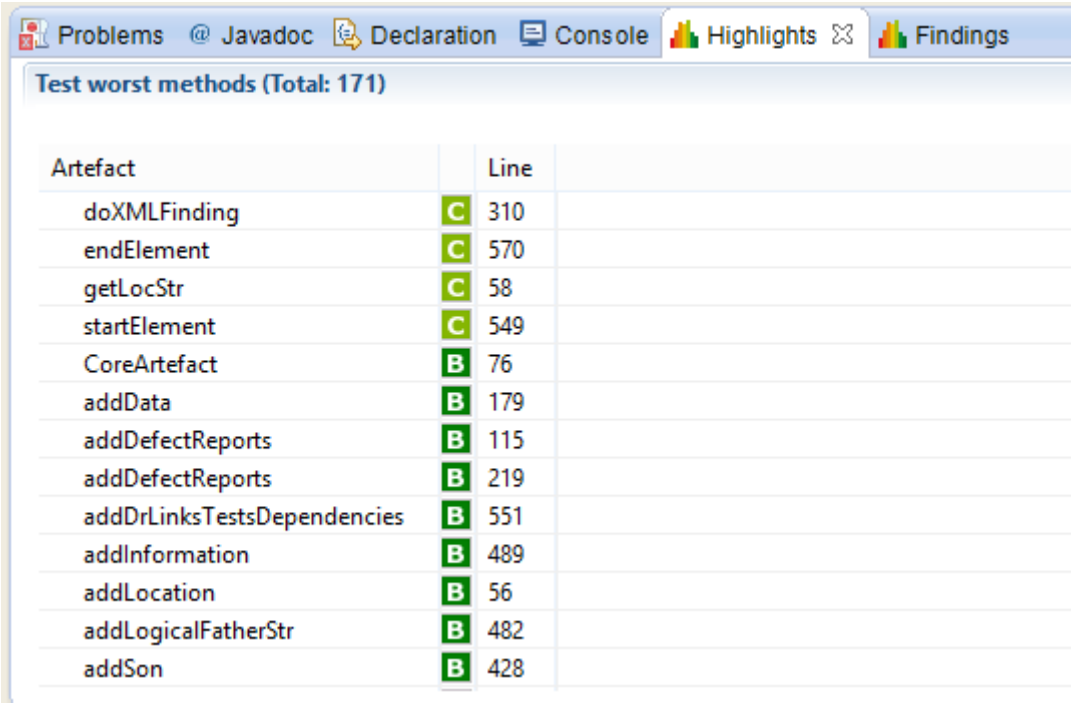
A finding full description

- 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.



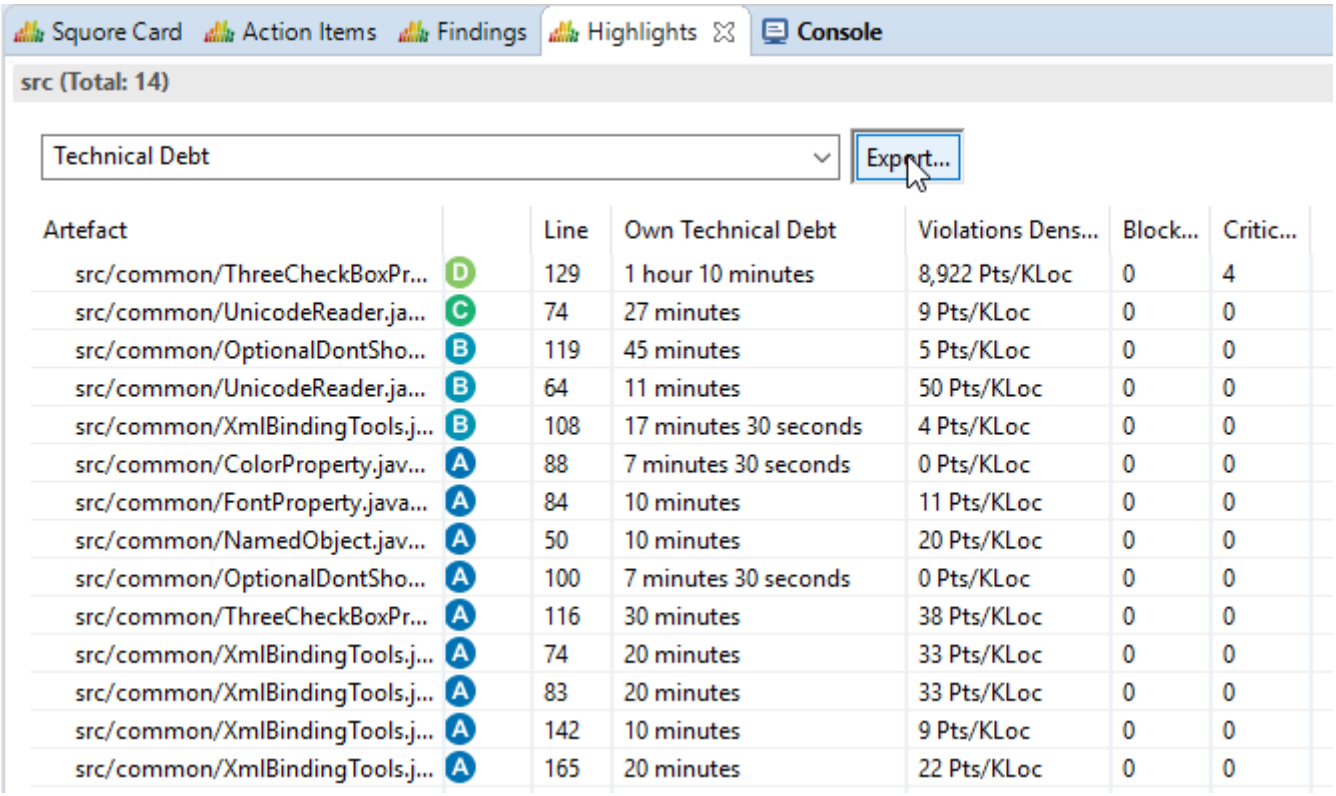
Highlights

Only the highlights created for the role **PLUGIN** are displayed in Eclipse. For more information about configuring highlights in your model, refer to the Configuration Guide.

Exporting Analysis Results

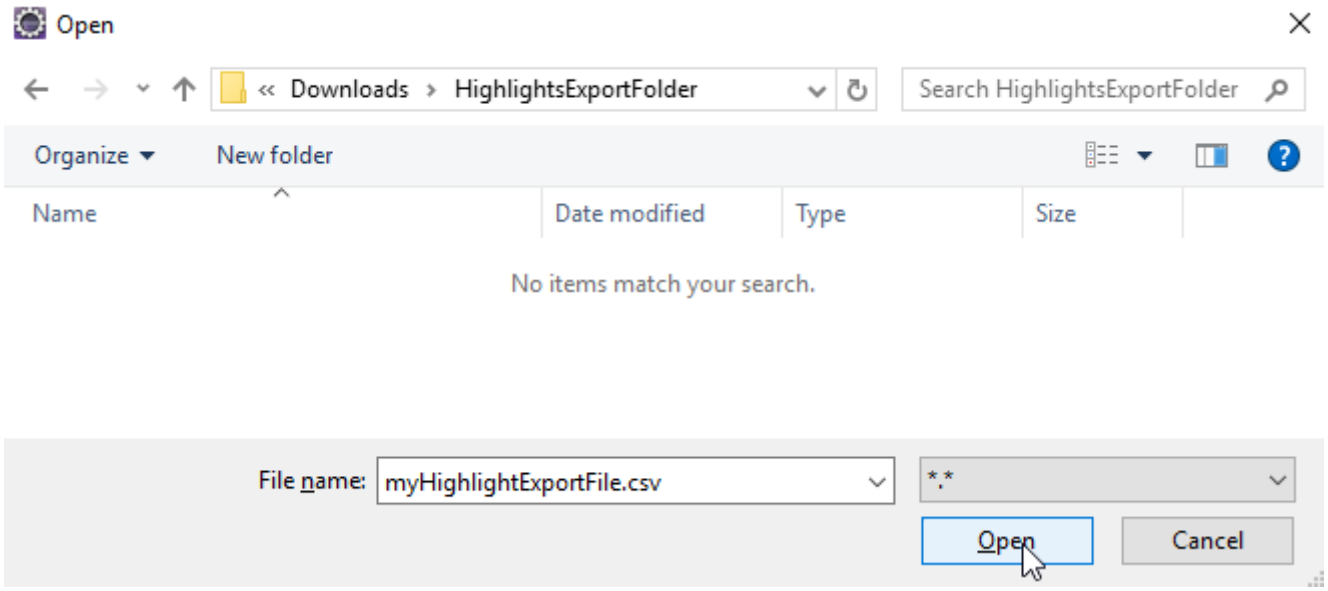
If you need to share your analysis results, it is possible to export the highlights as well as the squore card into a CSV file.

For example, go in the Highlights view, choose your highlight in the drop-down list and click export :



Highlights Export Button

Choose the name and folder path of your CSV export file :



Highlights Export File Dialog

You can then open your CSV file and work on your data :

	A	B	C	D	E	F	G
1	Artefact	Rating	Line	Own Technical Debt	Violations Density	Blocker Issues	Critical Issues
2	src/common/ThreeCheckB	D	129	1 hour 10 minutes	8,922 Pts/KLoc	0	4
3	src/common/UnicodeReac	C	74	27 minutes	9 Pts/KLoc	0	0
4	src/common/OptionalDon	B	119	45 minutes	5 Pts/KLoc	0	0
5	src/common/UnicodeReac	B	64	11 minutes	50 Pts/KLoc	0	0
6	src/common/XmlBindingT	B	108	17 minutes 30 seconds	4 Pts/KLoc	0	0
7	src/common/ColorPropert	A	88	7 minutes 30 seconds	0 Pts/KLoc	0	0
8	src/common/FontProperty	A	84	10 minutes	11 Pts/KLoc	0	0
9	src/common/NamedObjec	A	50	10 minutes	20 Pts/KLoc	0	0
10	src/common/OptionalDon	A	100	7 minutes 30 seconds	0 Pts/KLoc	0	0
11	src/common/ThreeCheckB	A	116	30 minutes	38 Pts/KLoc	0	0
12	src/common/XmlBindingT	A	74	20 minutes	33 Pts/KLoc	0	0
13	src/common/XmlBindingT	A	83	20 minutes	33 Pts/KLoc	0	0
14	src/common/XmlBindingT	A	142	10 minutes	9 Pts/KLoc	0	0
15	src/common/XmlBindingT	A	165	20 minutes	22 Pts/KLoc	0	0

Highlights CSV Export File Example

Index

A
Action Items view, 15

B
Baseline build, 10

D
Disk Space, 6
Draft build, 10

F
Findings view, 14

H
Highlights view, 15

J
Java, 6

M
Memory, 6, 6

P
Prerequisites, 6

S
Squore Card view, 13