

# CODE ANALYSIS USING SONARQUBE

---

*Author -* **ANGELIN**

## TABLE OF CONTENTS

1.	INTRODUCTION .....	4
1.1	SonarQube .....	4
1.2	SonarQube Features .....	4
2.	Environment setup to use SonarQube .....	5
2.1	SonarQube Server and SonarQube Runner – Setup .....	5
2.2	SonarQube Server Setup Validation .....	6
2.3	Environment Variables Setup .....	8
3.	Analyzing a Non - Maven project .....	9
3.1	Setup Of Non - Maven Project For SonarQube Analysis .....	9
3.2	Generating and Viewing the SonarQube report .....	11
4.	Analyzing a Maven project .....	12
4.1	Maven and Environment Setup .....	12
	Installing Maven .....	12
	Configuring Maven .....	12
4.2	Setup Of Maven Project For SonarQube Analysis .....	14
4.3	Generating and Viewing the SonarQube report .....	16
	Run SonarQube analysis .....	16
	Viewing the SonarQube report .....	16
5.	Analyzing with SonarQube Ant Task .....	17
5.1	Installing and Configuring SonarQube Ant Task .....	17
	Prerequisites .....	17
	SonarQube Ant Task Installation .....	17
5.2	Ant Build Script configuration .....	17
5.3	Generating and Viewing the SonarQube report .....	18
	Run SonarQube analysis .....	18
	Viewing the SonarQube report .....	19

6.	Analysis using SonarQube Eclipse Plugin .....	19
6.1	Installing SonarQube Eclipse plugin .....	19
6.2	Configuring SonarQube Eclipse plugin .....	23
6.3	Linking a Project to One Analyzed on a SonarQube Server .....	24
6.4	Working with SonarQube in Eclipse .....	26
	Run SonarQube analysis.....	26
	Viewing Issues using SonarQube Views in Eclipse .....	26
7.	References .....	30

## 1. INTRODUCTION

This document contains all essential information required to install and configure SonarQube and execute SonarQube analysis in Maven and non-Maven projects.

### 1.1 SONARQUBE

SonarQube™ software (previously known as “Sonar”) is an open source project for Continuous Inspection of code quality hosted at Codehaus. It is distributed under license LGPL v3.

### 1.2 SONARQUBE FEATURES

SonarQube covers the 7 axes of code quality:

- Architecture & Design
- Comments
- Coding rules
- Potential Bugs
- Duplications
- Unit Tests
- Complexity

SonarQube can perform analysis on [25+ different languages](#). The outcome of this analysis will be quality measures and issues (instances where coding rules were broken). However, what gets analyzed will vary depending on the language:

- On all languages, a static analysis of source code is performed (Java files, COBOL programs, etc.)
- A static analysis of compiled code can be performed for certain languages (.CLASS files or jars in Java, .DLL files in C#, etc.)
- A dynamic analysis of code can be performed on certain languages (execution of unit tests in Java, C#, etc.)

- SonarQube is also used for Android Development.
- SonarQube internally uses PMD, Findbugs, CheckStyle etc.
- Integration of SonarQube with standard ALM components such as Maven, Ant, SVN, Git, Mercurial, JIRA, Mantis, Google Analytics, Piwik, Fortify etc. comes out of the box.

## 2. ENVIRONMENT SETUP TO USE SONARQUBE

### 2.1 SONARQUBE SERVER AND SONARQUBE RUNNER – SETUP

- **Java** - Download and install latest version of Java if it is not already available.
- **SonarQube** - Download latest version of SonarQube from <http://www.SonarQube.org/downloads> and unzip to desired location.
- **SonarQube Runner** - Download latest version of SonarQube runner from <http://docs.codehaus.org/display/SONAR/Installing+and+Configuring+Sonar+Runner> and unzip to desired location.

The screenshot shows a Windows File Explorer window with the address bar set to 'My Computer > OS (C:) > Program Files (x86)'. The main area displays a list of folders with columns for Name, Date modified, and Type. The folders listed are: Microsoft Synchronization Services, Microsoft Visual Studio 8, Microsoft.NET, Mozilla Firefox, Mozilla Maintenance Service, MSBuild, Oracle, Reference Assemblies, sonarqube-4.3.1, and sonar-runner-2.4. The 'sonarqube-4.3.1' and 'sonar-runner-2.4' folders are highlighted in blue, indicating they are selected.

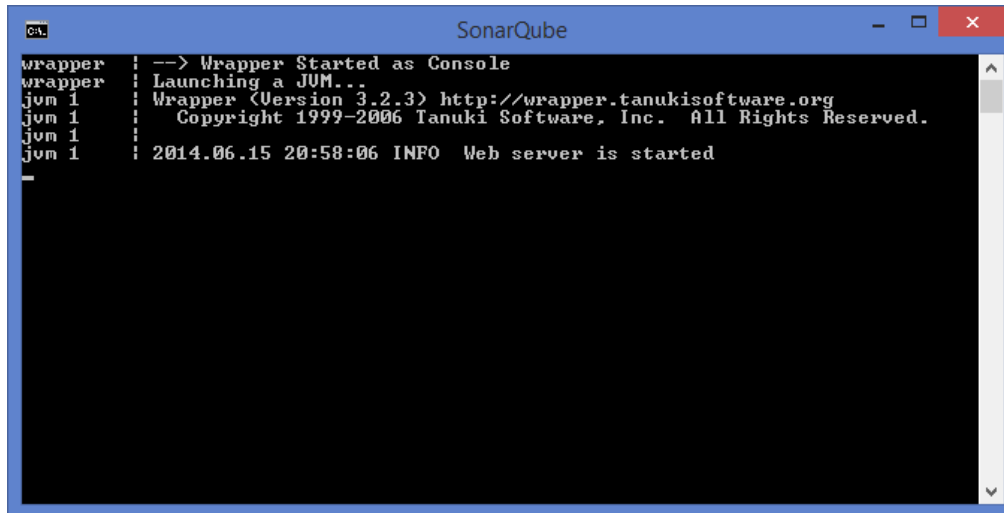
Name	Date modified	Type
Microsoft Synchronization Services	4/12/2014 11:59 AM	File folder
Microsoft Visual Studio 8	4/12/2014 11:56 AM	File folder
Microsoft.NET	4/12/2014 11:58 AM	File folder
Mozilla Firefox	4/12/2014 11:54 AM	File folder
Mozilla Maintenance Service	4/12/2014 11:54 AM	File folder
MSBuild	4/12/2014 11:59 AM	File folder
Oracle	4/12/2014 5:22 PM	File folder
Reference Assemblies	2/12/2014 8:21 AM	File folder
sonarqube-4.3.1	6/8/2014 10:38 PM	File folder
sonar-runner-2.4	6/8/2014 5:37 PM	File folder

#### **NOTE:**

SonarQube Runner is recommended as the default launcher to analyze a project with SonarQube.

## 2.2 SONARQUBE SERVER SETUP VALIDATION

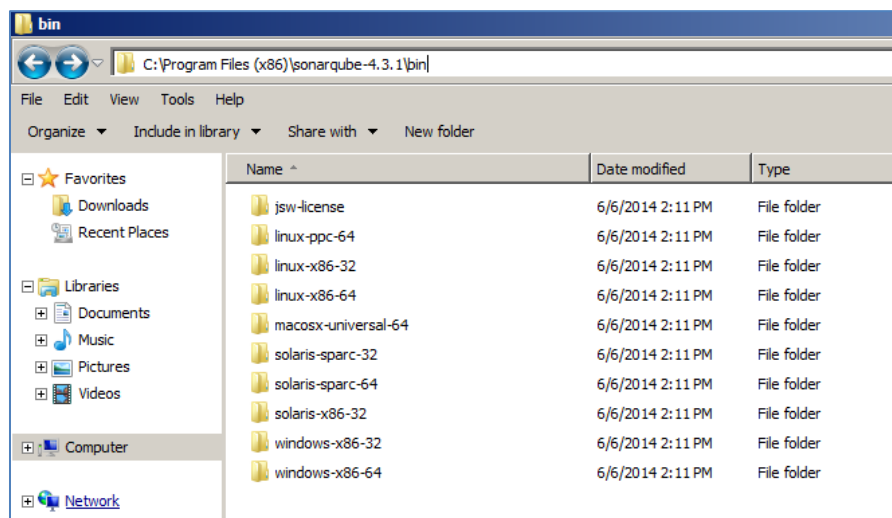
1. Start the SonarQube server using the startup script available in **SONAR\_HOME\bin** folder.



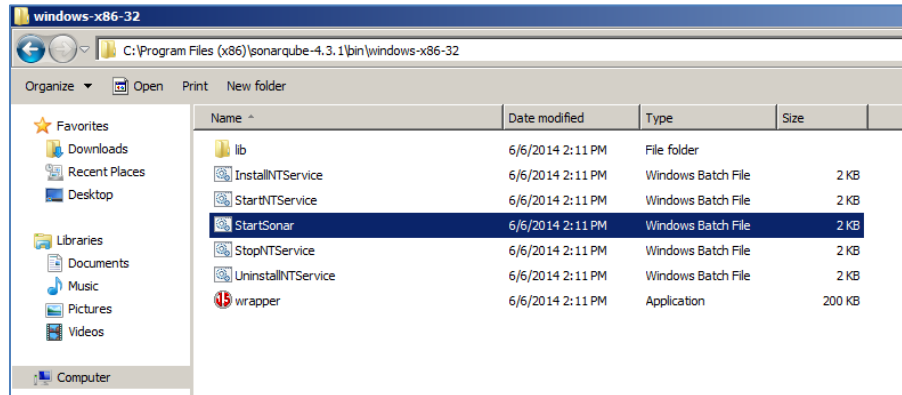
```
CA. SonarQube
wrapper | --> Wrapper Started as Console
wrapper | Launching a JVM...
jvm 1 | Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
jvm 1 | Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.
jvm 1 |
jvm 1 | 2014.06.15 20:58:06 INFO Web server is started
```

### NOTE:

In **SONAR\_HOME\bin** folder, subfolders appropriate to the Operating System will be available and they will in turn contain the SonarQube server startup scripts.



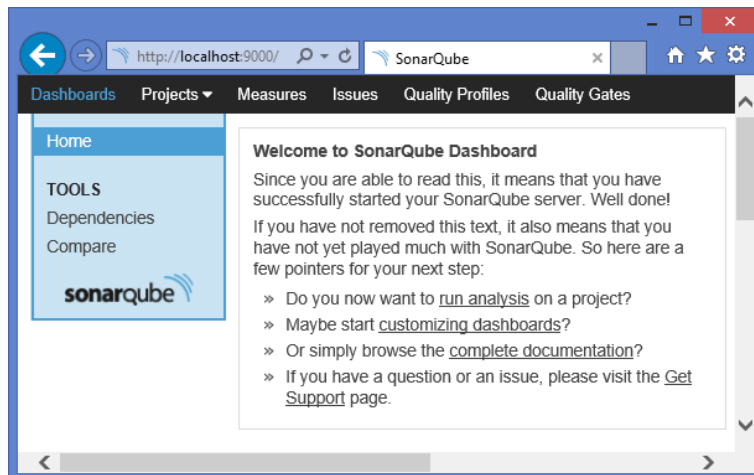
**EXAMPLE: `SONAR_HOME\bin\windows-x86-32\StartSonar.bat`**



2. After the server starts, access the web interface of the SonarQube at <http://localhost:9000/>

**Note:**

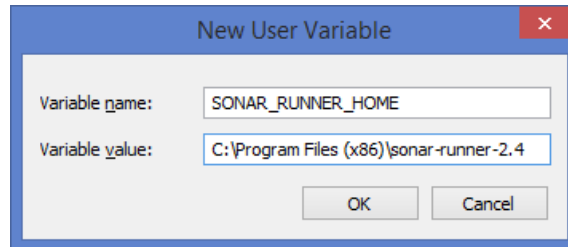
- For remote access, replace *localhost* with the address of the SonarQube server.
- 9000 is the default port and can be changed in `SONAR_HOME\conf\sonar.properties`
- Default credentials is *admin/admin*



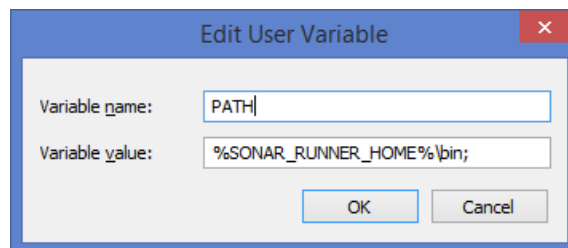
## 2.3 ENVIRONMENT VARIABLES SETUP

1. Set a new environment variable as **SONAR\_RUNNER\_HOME**. And its value should be the unzipped path of *sonar-runner* zip file.

Example, "C:\Program Files (x86)\sonar-runner-2.4"



2. Append Sonar Runner's bin path (**%SONAR\_RUNNER\_HOME%\bin**) to the environment variable "PATH".



3. Update the global settings (database connection, server URL) by editing *sonar-runner.properties* file under Sonar Runner's **conf** folder.

*<Sonar\_runner\_installed\_directory>/conf/sonar-runner.properties*



```
sonar-runner - Notepad
File Edit Format View Help
#Configure here general information about the environment, such as
SonarQube DB details for example
#No information about specific project should appear here

#----- Default SonarQube server
sonar.host.url=http://localhost:9000

#----- PostgreSQL
#sonar.jdbc.url=jdbc:postgresql://localhost/sonar

#----- MySQL
#sonar.jdbc.url=jdbc:mysql://localhost:3306/sonar?
useUnicode=true&characterEncoding=utf8

#----- Oracle
#sonar.jdbc.url=jdbc:oracle:thin:@localhost/XE

#----- Microsoft SQLServer
#sonar.jdbc.url=jdbc:jtds:sqlserver://localhost/sonar;SelectMethod=Cursor

#----- Global database settings
#sonar.jdbc.username=sonar
#sonar.jdbc.password=sonar

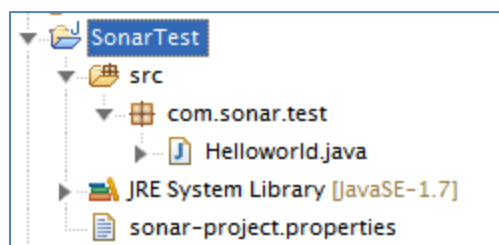
#----- Default source code encoding
#sonar.sourceEncoding=UTF-8

#----- Security (when 'sonar.forceAuthentication' is set to 'true')
#sonar.login=admin
#sonar.password=admin
```

### 3. ANALYZING A NON - MAVEN PROJECT

#### 3.1 SETUP OF NON - MAVEN PROJECT FOR SONARQUBE ANALYSIS

A ***sonar-project.properties*** file has to be placed into the root of the project folder.



Minimum content of the **sonar-project.properties** file will be as follows.

**Sample *sonar-project.properties* file**

```
# Required metadata

sonar.projectKey=java-sonar-runner-simple

sonar.projectName=Simple Java Project analyzed with the SonarQube
Runner

sonar.projectVersion=1.0

# Comma-separated paths to directories of source codes to be analyzed.

# Path is relative to the sonar-project.properties file.

# Replace "\" by "/" on Windows.

# Since SonarQube 4.2, this property is optional.

# If not set, SonarQube starts looking for source code

# from the directory containing the sonar-project.properties file.

sonar.sources=src

# Language

sonar.language=java

# Encoding of the source code
sonar.sourceEncoding=UTF-8
```

1. Prepare a similar file with content relevant to the Project to be analyzed.
2. Place the ***sonar-project.properties*** files into the root of the project folder.

## 3.2 GENERATING AND VIEWING THE SONARQUBE REPORT

1. Ensure that the steps in Section 2 above are executed and the SonarQube server is started.
2. Go to command line and change directory (cd) to the *project's root directory* and run the command

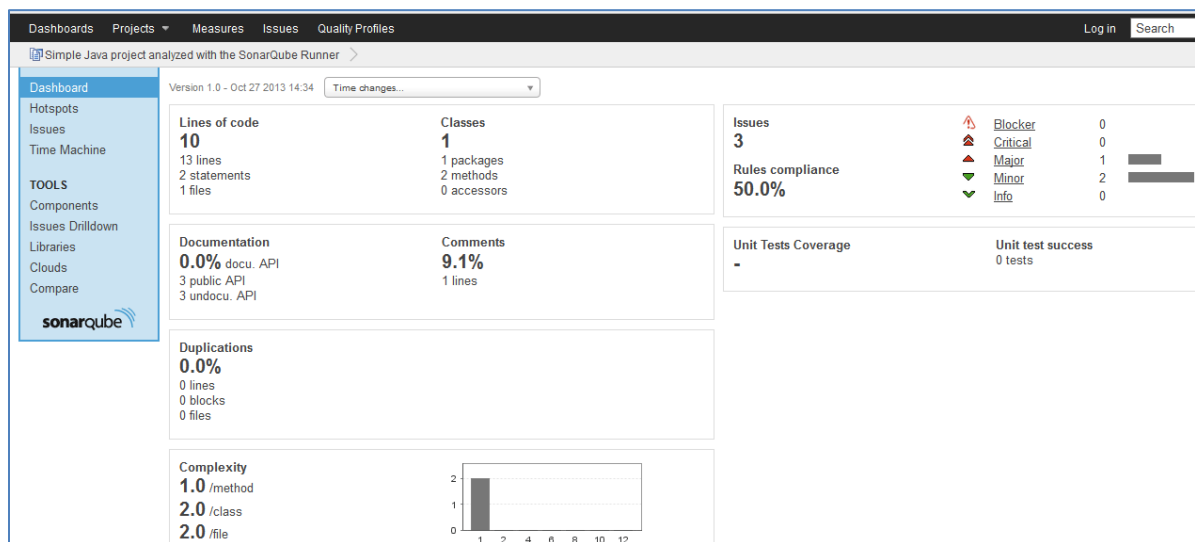
`[path of sonar-runner]\bin\sonar-runner.bat`

3. Upon successful execution of the above command, check the SonarQube dashboard by opening SonarQube Server's admin page at <http://localhost:9000> (default URL).

Default credentials – *admin/admin*.

4. The project will be listed on the dashboard with the name given in *sonar-project.properties* file.
5. Click on the project name and explore further to view the code analysis report.

A sample report is shown below:



### Note:

The above result is only from PMD and Checkstyle execution by SonarQube. To run code coverage as well as Findbug plugins, the binary path is needed in the *sonar-project.properties* file.

## 4. ANALYZING A MAVEN PROJECT

### 4.1 MAVEN AND ENVIRONMENT SETUP

#### INSTALLING MAVEN

Download and install [Maven](#) (with a version compatible with the SonarQube version installed) if not already installed.

#### CONFIGURING MAVEN

1. Edit the [settings.xml](#) file located in `$MAVEN_HOME/conf` or `~/.m2`, to set the database parameters to be used as well as the SonarQube server URL.

**Example:**

```
<settings>

  <profiles>

    <profile>

      <id>sonar</id>

      <activation>

        <activeByDefault>true</activeByDefault>

      </activation>

      <properties>

        <!-- Example for MySQL-->

        <sonar.jdbc.url>
```

```
    jdbc:mysql://localhost:3306/sonar?useUnicode=true&characterEncoding=utf8

</sonar.jdbc.url>

<sonar.jdbc.username>sonar</sonar.jdbc.username>

<sonar.jdbc.password>sonar</sonar.jdbc.password>

<!-- Optional URL to server. Default value is http://localhost:9000 -->

<sonar.host.url>

http://localhost:9000

</sonar.host.url>

</properties>

</profile>

</profiles>

</settings>
```

**Note:**

The Maven plugin requires some additional configuration to establish a connection to the web platform, where it reports the results of the analysis – *sonar.host.url* and *sonar.jdbc.url* will help with that. Other properties, like *sonar.analysis.mode*, *sonar.profile* and *sonar.dynamicAnalysis* configure the plugin to gather certain statistics only.

2. Restart the SonarQube server.

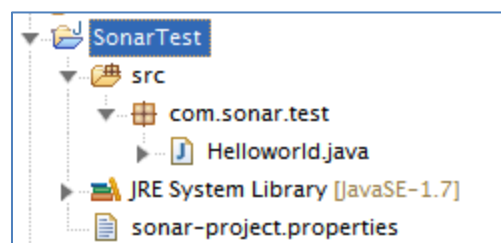
## 4.2 SETUP OF MAVEN PROJECT FOR SONARQUBE ANALYSIS

1. Go to the Maven project and edit *pom.xml* file adding the SonarQube plugin for the Maven version (<http://docs.codehaus.org/display/SONAR/Analyzing+with+Maven>)

### Maven3 configuration:

```
<build>
  <pluginManagement>
    <plugins>
      <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>sonar-maven-plugin</artifactId>
        <version>2.0</version>
      </plugin>
    </plugins>
  </pluginManagement>
</build>
```

2. A **sonar-project.properties** file has to be placed into the root of the project folder.



Minimum content of the **sonar-project.properties** file will be as follows.

**Sample *sonar-project.properties* file**

```
# Required metadata

sonar.projectKey=java-sonar-runner-simple

sonar.projectName=Simple Java Project analyzed with the SonarQube
Runner

sonar.projectVersion=1.0

# Comma-separated paths to directories of source codes to be analyzed.

# Path is relative to the sonar-project.properties file.

# Replace "\" by "/" on Windows.

# Since SonarQube 4.2, this property is optional.

# If not set, SonarQube starts looking for source code

# from the directory containing the sonar-project.properties file.

sonar.sources=src

# Language

sonar.language=java

# Encoding of the source code

sonar.sourceEncoding=UTF-8
```

- a) Prepare a similar file with content relevant to the Project to be analyzed.
- b) Place the ***sonar-project.properties*** files into the root of the project folder.

## 4.3 GENERATING AND VIEWING THE SONARQUBE REPORT

### RUN SONARQUBE ANALYSIS

From the project root directory where the pom.xml file is also placed, execute the following command giving *user/password* credentials:

```
mvn sonar:sonar
```

#### Note:

# The sonar:sonar goal must be executed in a dedicated mvn command

```
mvn clean install
```

```
mvn sonar:sonar
```

# The following command may lead to unexpected issues

```
mvn clean install sonar:sonar
```

### VIEWING THE SONARQUBE REPORT

1. Upon successful execution of the **mvn sonar:sonar** command, check the SonarQube dashboard by opening SonarQube Server admin page <http://localhost:9000> (default URL). Default credentials – *admin/admin*.
2. The project will be listed on the dashboard with the name given in **sonar-project.properties**.
3. Click on project name and explore further to view the code analysis report.



## 5. ANALYZING WITH SONARQUBE ANT TASK

### 5.1 INSTALLING AND CONFIGURING SONARQUBE ANT TASK

The SonarQube Ant Task allows integration of [SonarQube analysis](#) into an Apache Ant build script.

#### PREREQUISITES

- Ant 1.7.1 or higher
- Java 1.6 or higher

#### SONARQUBE ANT TASK INSTALLATION

Download the SonarQube Ant Task on your machine.

##### Download Path:

<http://repository.codehaus.org/org/codehaus/sonar-plugins/sonar-ant-task/2.1/sonar-ant-task-2.1.jar>

### 5.2 ANT BUILD SCRIPT CONFIGURATION

Define a new sonar Ant target in your Ant build script:

#### build.xml

```
<project name="My Project" default="all" basedir="." xmlns:sonar="antlib:org.sonar.ant">
...
<!-- Define the SonarQube global properties (the most usual way is to pass these properties via the
command line) -->
<property name="sonar.jdbc.url"
value="jdbc:mysql://localhost:3306/sonar?useUnicode=true&characterEncoding=utf8" />
<property name="sonar.jdbc.username" value="sonar" />
<property name="sonar.jdbc.password" value="sonar" />
...
<!-- Define the SonarQube project properties -->
```

```
<property name="sonar.projectKey" value="org.codehaus.sonar:example-java-ant" />

<property name="sonar.projectName" value="Simple Java Project analyzed with the SonarQube Ant
Task" />

<property name="sonar.projectVersion" value="1.0" />

<property name="sonar.language" value="java" />

<property name="sonar.sources" value="src" />

<property name="sonar.binaries" value="build/*.jar" />

...

<!-- Define the SonarQube target -->

<target name="sonar">

<taskdef uri="antlib:org.sonar.ant" resource="org/sonar/ant/antlib.xml">

<!-- Update the following line, or put the "sonar-ant-task-*.jar" file in your "$HOME/.ant/lib" folder -->

<classpath path="path/to/sonar/ant/task/lib/sonar-ant-task-*.jar" />

</taskdef>

<!-- Execute the SonarQube analysis -->

<sonar:sonar />

</target>

...
```

### 5.3 GENERATING AND VIEWING THE SONARQUBE REPORT

#### RUN SONARQUBE ANALYSIS

Run the following command from the project base directory to launch the analysis:

```
ant sonar
```

---

## VIEWING THE SONARQUBE REPORT

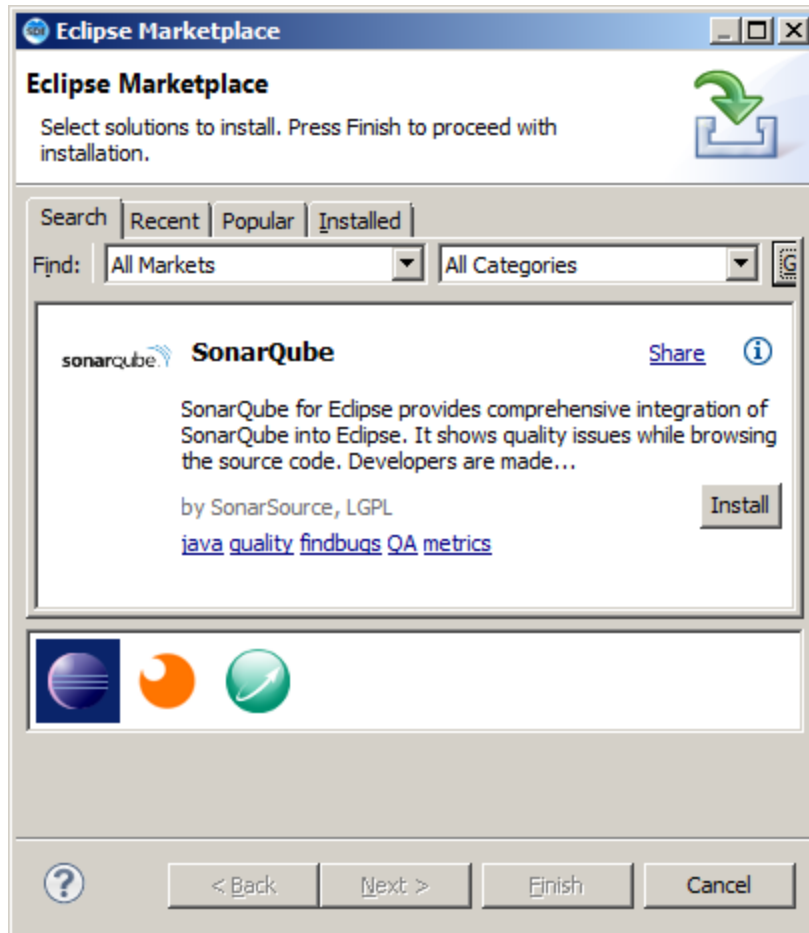
1. Upon successful execution of the ***mvn sonar:sonar*** command, check the SonarQube dashboard by opening SonarQube Server admin page <http://localhost:9000> (default URL). Default credentials – *admin/admin*.
2. The project will be listed on the dashboard with the name given in ***sonar-project.properties***.
3. Click on project name and explore further to view the code analysis report.

## 6. ANALYSIS USING SONARQUBE ECLIPSE PLUGIN

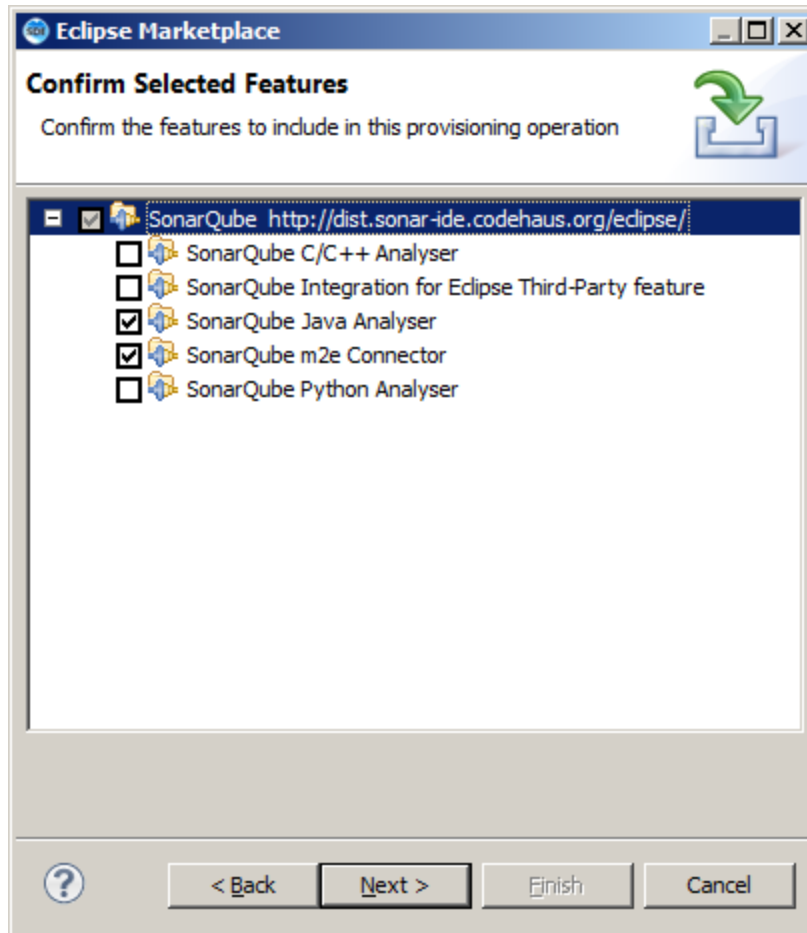
The SonarQube Eclipse Plugin provides a comprehensive integration of SonarQube in Eclipse for Java, C/C++ and Python projects.

### 6.1 INSTALLING SONARQUBE ECLIPSE PLUGIN

- If a previous version of SonarQube Eclipse plugin is already installed, you can update it. Go to **Help > Check for Updates**.
- To install SonarQube Eclipse plugin in Eclipse IDE:
  1. Go to **Help > Eclipse Marketplace...** and search for "SonarQube"



This should display the list of available plugins and components:



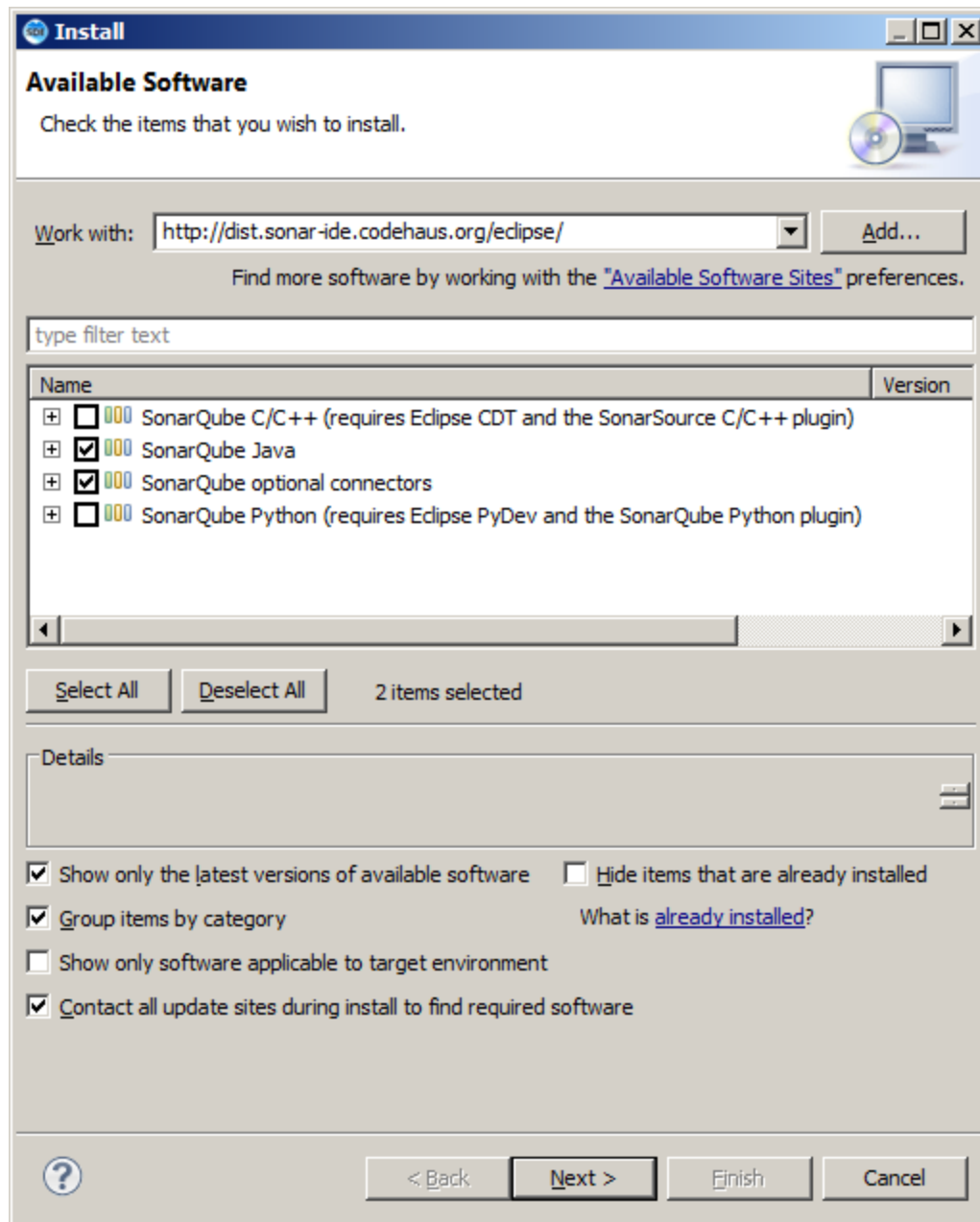
Or

Go to **Help > Install New Software...**

This should display the **Install** dialog box.

Paste the Update Site URL (<http://dist.sonar-ide.codehaus.org/eclipse/> the latest version or <http://dist.sonar-ide.codehaus.org/eclipse-archives/> for previous ones) into the field **Work with** and press **Enter**.

This should display the list of available plugins and components:

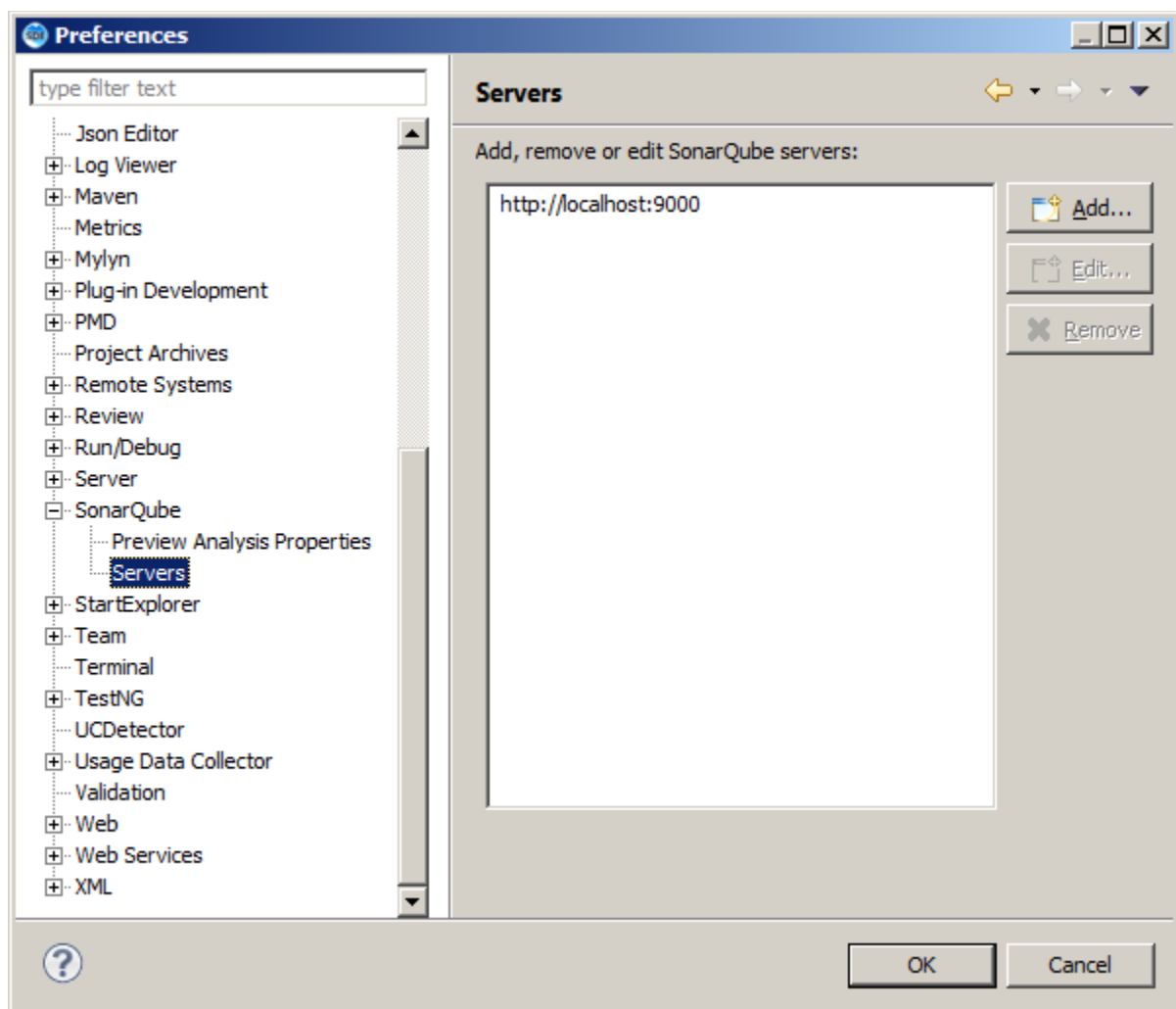


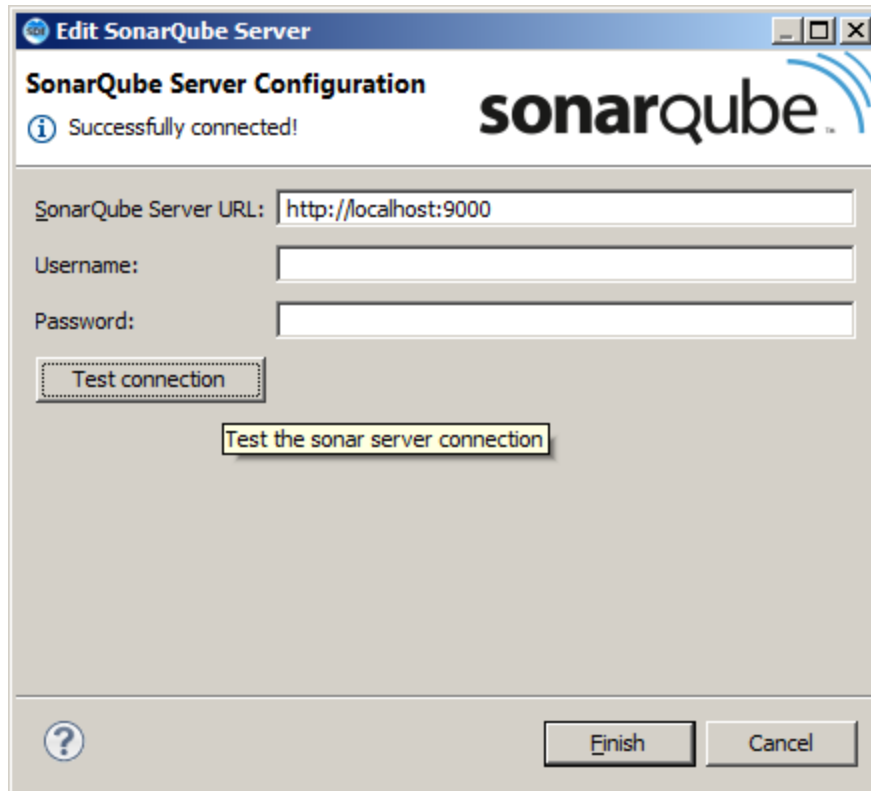
1. Check the SonarQube components to install.
2. Click **Next**. Eclipse will then check to see if there is any issue which would prevent a successful installation.
3. Click **Finish** to begin the installation process. Eclipse will then download and install the necessary components.
4. Once the installation process is finished, Eclipse will ask if you want to restart the IDE. It is strongly recommended that you restart the IDE.

## 6.2 CONFIGURING SONARQUBE ECLIPSE PLUGIN

- A project should have been already created and configured with SonarQube server in order to start analyzing it with SonarQube in Eclipse.
- The SonarQube server connection should be established before attempting to run the analysis. This can be ensured by testing the Server connection in Eclipse through the menu option: **Windows > Preferences > SonarQube > Servers**.

Default server is <http://localhost:9000/>





If the test does not return the message “**Successfully connected!**”, then start the SonarQube server and retest the connection using the same steps.

## 6.3 LINKING A PROJECT TO ONE ANALYZED ON A SONARQUBE SERVER

### LINKING FOR THE FIRST TIME

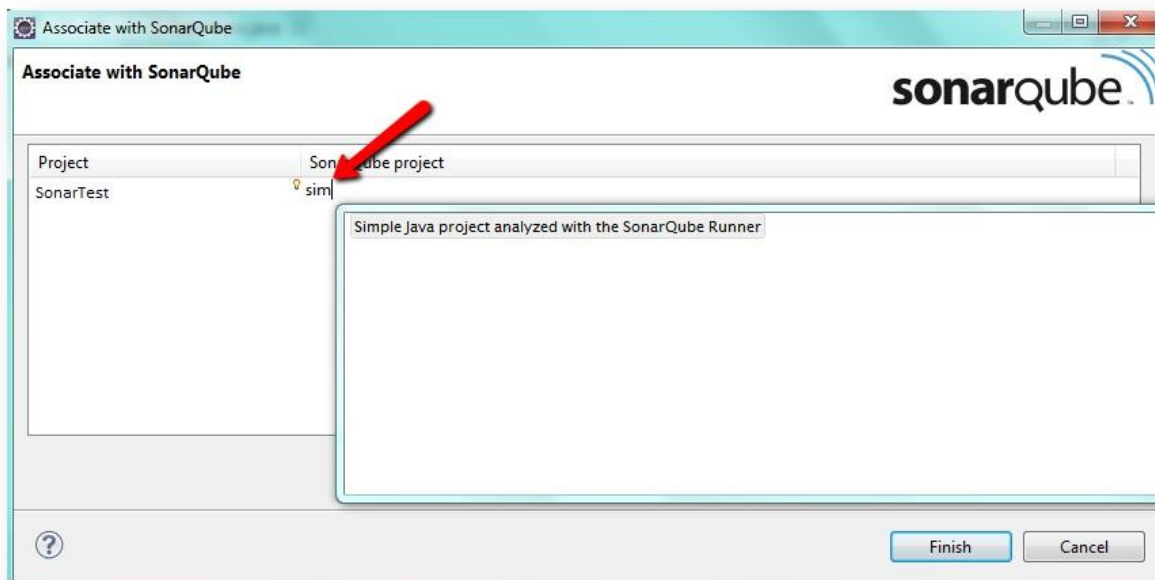
Once the SonarQube server is defined, the next step is to link your Eclipse project with a project defined on this SonarQube server.

To do so, right-click on the project in the **Project Explorer**, and then choose the option **Configure > Associate with SonarQube**





A dialog appears with the **SonarQube project** text field. Start typing the name of the project in the text field against SonarQube project column and select it in the list box:



Click on **Finish**.

Now the project is associated to the one analyzed on the SonarQube server and the SonarQube reports can be viewed within eclipse.

**Note:**

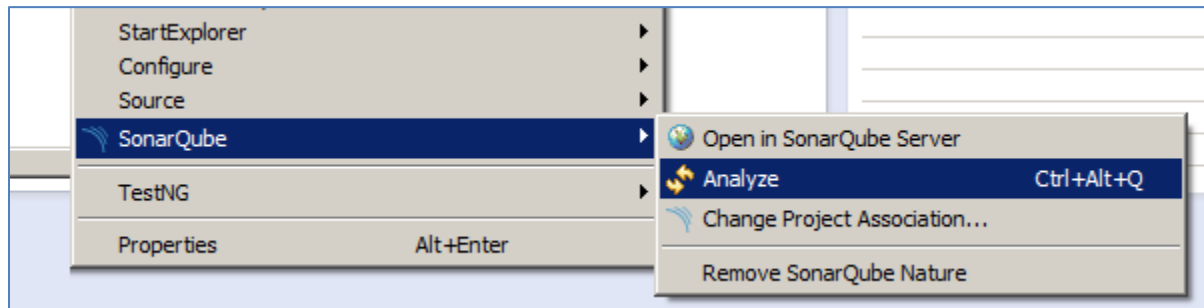
**Automatic association for Maven projects**

For Maven projects, the association is performed automatically.

## 6.4 WORKING WITH SONARQUBE IN ECLIPSE

### RUN SONARQUBE ANALYSIS

To run a new analysis, right click on the project and go to **SonarQube > Analyze**.

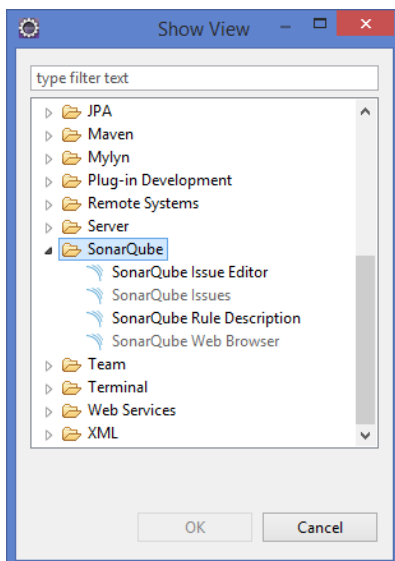


#### Note:

You can also hit **Ctrl+Alt+Q** wherever you are in your project to trigger a new analysis.

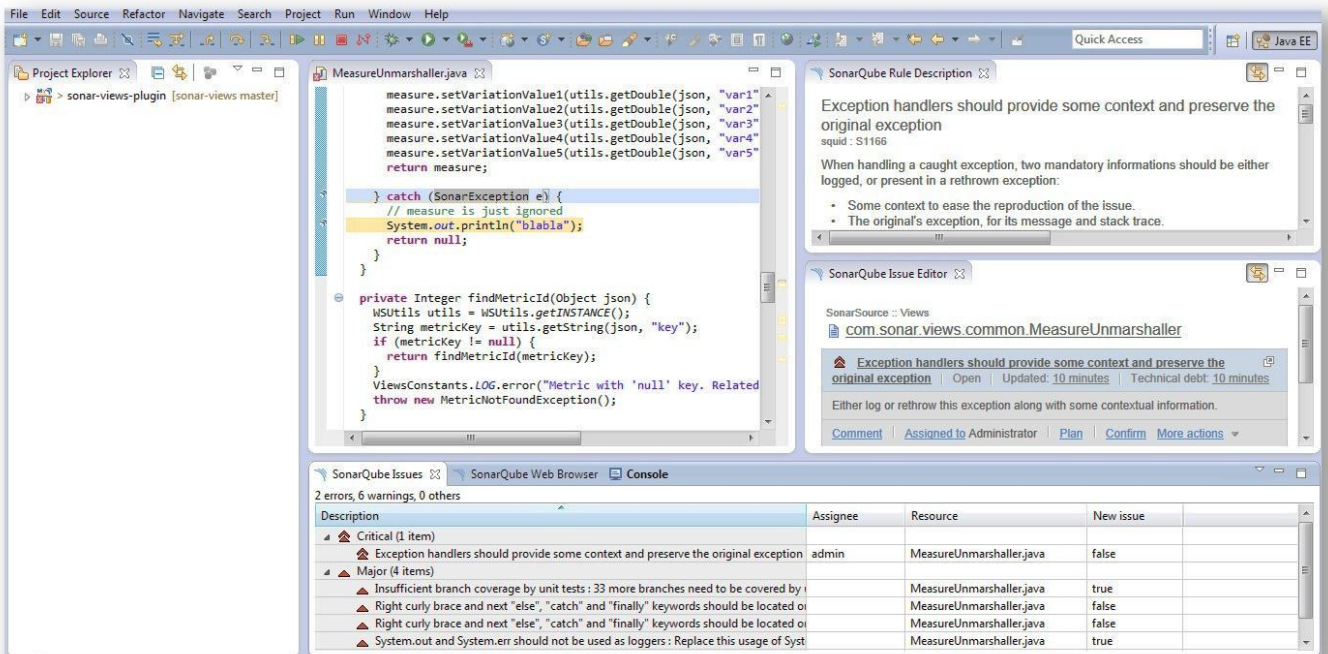
### VIEWING ISSUES USING SONARQUBE VIEWS IN ECLIPSE

Four different views are available in Eclipse to browse the quality of the projects:



- **SonarQube Issues** that list the issues on the selected component
- **SonarQube Issue Editor** to review the issue

- **SonarQube Rule Description** to get the detailed description of the rule that is violated
- **SonarQube Web Browser**



## SONARQUBE ISSUES

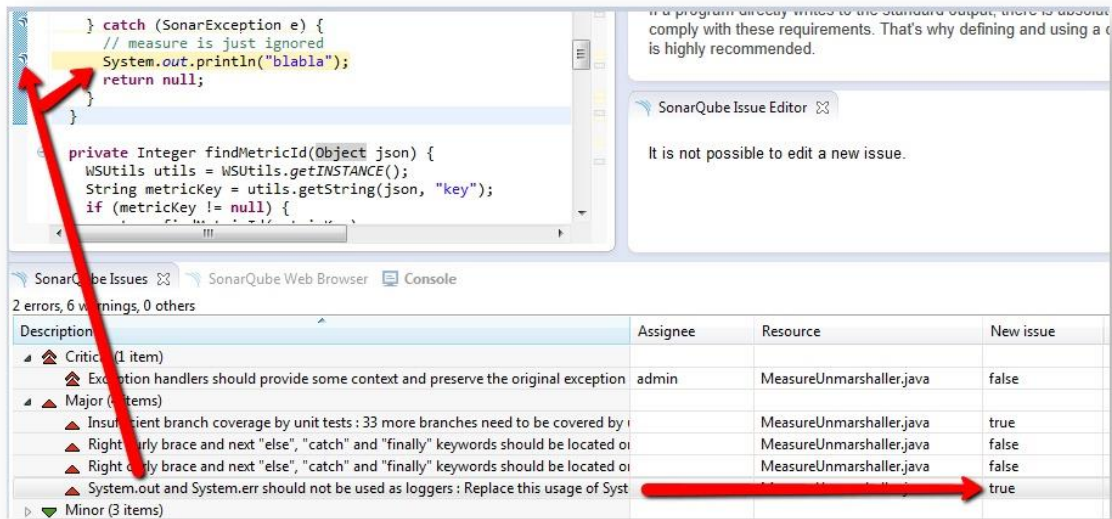
The **SonarQube Issues** view displays the list of issues of the selected component (project, module, file, etc.).

To display this view, go to **Window > Show View > Other... > SonarQube > SonarQube Issues**.

**Problems** and **Markers** views also display issues.

Issues are flagged in the source code viewer.

New issues (compared to the latest version on the SonarQube server) are highlighted in yellow. This allows you to focus on the new issues that you have introduced.



You can add/remove/order columns, group issues (by severity, new issues only, ...), filter issues (new issues only, issues assigned to me, ...), sort (by assignee, ...).

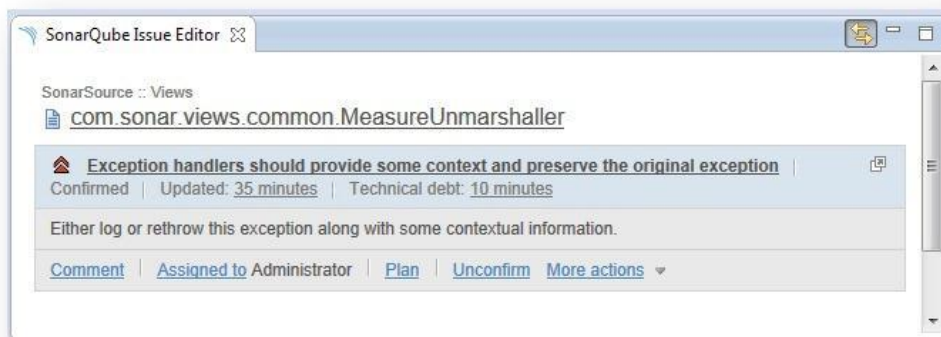
**Note:**

"Resolved" issues are not displayed.

**SONARQUBE ISSUE EDITOR**

This view allows you to review the selected issue (add comments, confirm it, plan it, etc.) the same way you would do it through the web interface.

To display this view, go to **Window > Show View > Other... > SonarQube > SonarQube Issue Editor**.



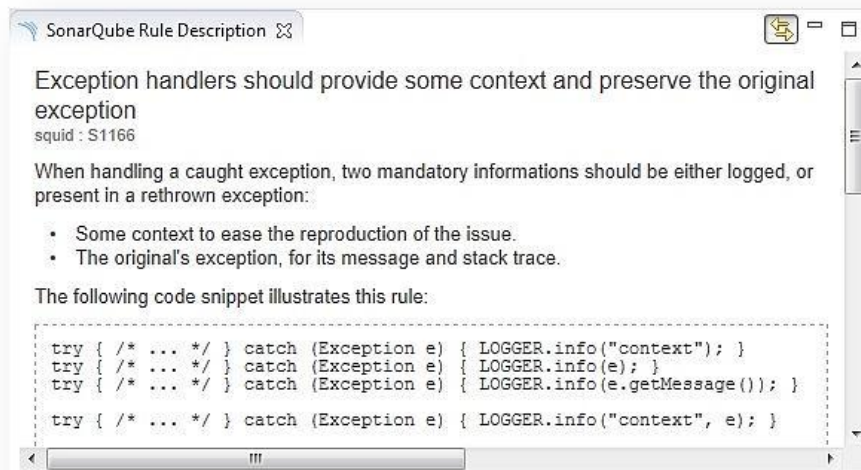
## Note:

It is not possible to review new issue that you have introduced.

## SONARQUBE RULE DESCRIPTION

This view allows you to access the detailed description of the coding rule of the selected issue.

To display this view, go to **Window > Show View > Other... > SonarQube > SonarQube Rule Description**.



## SONARQUBE WEB BROWSER

This view allows you to access the SonarQube web interface.

This view is automatically displayed when clicking on **SonarQube > Open in SonarQube server**.

It can also be displayed by going to **Window > Show View > Other... > SonarQube > SonarQube Web Browser**.



## 7. REFERENCES

1. Compatibility matrix showing the versions compatible between SonarQube, SonarQube Ant Task, SonarQube Runner and Maven at :  
<http://docs.codehaus.org/display/SONAR/Analyzing+Source+Code#AnalyzingSourceCode-CompatibilityMatrix>
2. <http://docs.codehaus.org/display/SONAR/Working+with+SonarQube+in+Eclipse>