

Guidelines – Jenkins and Sonar Build process

JSAG Suggests Series



Nagarro Software Pvt. Ltd.
Java Software & Architecture Group

Revision History			
Version	Date	Author/Contributor	Comments
0.1	23-Mar-2012	Amit Balwani	Compiled document.
0.1	27-Mar-2012	Archna Aggarwal	Review comments
1.0	29-Mar-2012	Amit Balwani	Baseline Release
1.1	19-Jun-2012	Amit Balwani	Updated Sonar configuration required for Ant.
1.2	28-Dec-2012	Vaibhav Shukla	Migrating from Hudson to Jenkins
1.3	28-Dec-2012	Kuldeep Singh	Reviewed and Updated
1.4	25-Jan-2013	Amit Balwani	Updated a note on SVN credentials to be updated regularly when changed in the domain

Contents

Contents.....	3
1. Introduction	4
2. Jenkins Setup.....	5
2.1. Build Job Creation	5
2.1.1. Create a new job	5
2.1.2. Add SVN location.....	7
2.1.3. Build triggers	8
2.1.4. Build Step using Maven.....	9
2.1.5. Post-build actions setup.....	10
2.2. Manual build execution	12
3. Sonar Setup for Jenkins.....	12
3.1. Sonar configuration in Jenkins project build Job	13
4. Accessing Build and Sonar reports in Jenkins	14
4.1. Accessing Build when built using Maven	14
4.2. Accessing Sonar reports.....	16

1. Introduction

The purpose of this document is to guide project teams to automate the build process for Java based projects and execute code analyzer tools (PMD, Checkstyle, Findbugs) on a pre-defined schedule. This is enabled through a Jenkins and Sonar installation on centralized JSAG server machines. Project teams can configure and run the build process through an easy to use web interface exposed by Jenkins and Sonar.

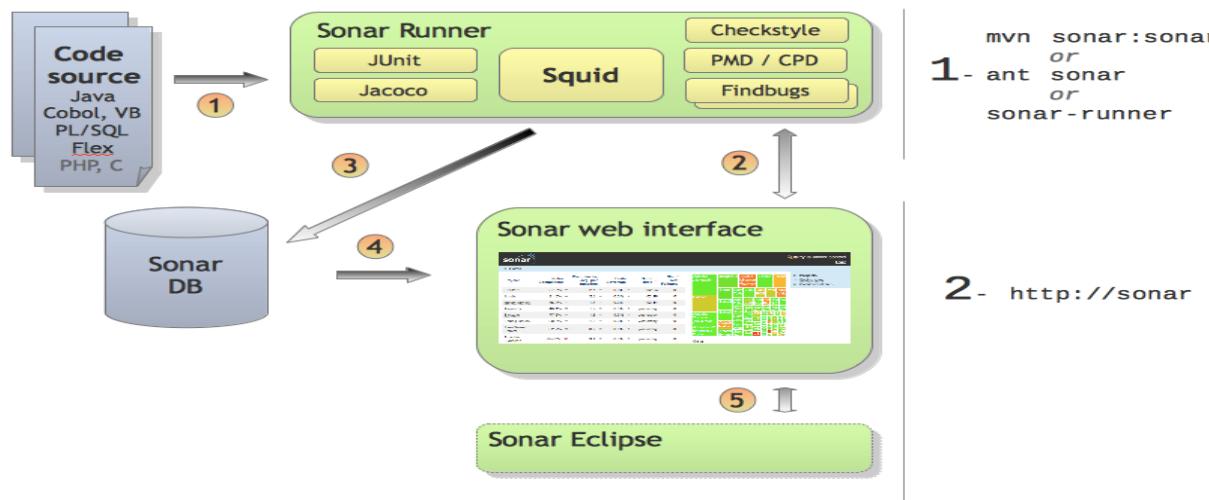
Jenkins is an award winning application that monitors executions of repeated jobs, such as building a software project or jobs run by cron. Among those things, current Jenkins focuses on the following two jobs:

1. **Building/testing software projects continuously**, just like CruiseControl or DamageControl. In a nutshell, Jenkins provides an easy-to-use so-called continuous integration system, making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. The automated, continuous build increases the productivity.
2. **Monitoring executions of externally-run jobs**, such as cron jobs and procmail jobs, even those that are run on a remote machine. For example, with cron, all you receive is regular e-mails that capture the output, and it is up to you to look at them diligently and notice when it broke. Jenkins keeps those outputs and makes it easy for you to notice when something is wrong.

Sonar is an open platform to manage code quality. As such, it covers the 7 axes of code quality:

- Architecture and design
- Comments
- Coding rules
- Potential bugs
- Complexity
- Duplications
- Unit tests

Sonar architecture is explained in the image shown below:



Project teams can benefit from this automated build process and code analysis enabling them to ensure code quality and pre-scheduled builds without any hardware or software installation.

Please note that it is a mandatory requirement for all Java projects to setup their project's build on the JSAG Jenkins and Sonar server. If because of any infrastructure issue, it is not possible to setup build on the JSAG Jenkins and Sonar server, an approval should be taken by sending email to jsag@nagarro.com at the start of the project.

2. Jenkins Setup

Jenkins server web interface can be accessed through the URL <http://jsag.ggn.nagarro.com/jenkins/>

2.1. Build Job Creation

A build job needs to be setup for a project with project specific settings. Please follow the steps in the sections below to setup a job and configure the build on it.

2.1.1. Create a new job

A new job can be created by sending an email to scm@nagarro.com with the following details:

- SVN credentials – *Make sure you change it in the job regularly when you change it on the domain else the build will fail and give “SVN update failed” error.*
- SVN location
- Job name
- Job schedule

A new job can be created by following the steps below using the Jenkins UI interface as well.

1. Go to the URL <http://jsag.ggn.nagarro.com/jenkins/> using your browser, you will see the page shown below.

```
mvn sonar:sonar
or
ant sonar
or
sonar-runner
```

2- <http://sonar>

The screenshot shows the Jenkins dashboard for the 'Nagarro Centralise Jenkins System'. On the left, there's a sidebar with links for People, Build History, Project Relationship, Check File Fingerprint, Job Import Plugin, and Dependency Graph. Below that are sections for Build Queue (empty) and Build Executor Status (4 idle executors). The main area displays a table of build jobs:

S	W	Name	Last Success	Last Failure	Last Duration	Cron Trigger
		Backlog_App	1 mo 7 days (#24)	1 hr 44 min (#26)	2 min 30 sec	
		BCR-Android-MobileBanking	1 mo 5 days (#107)	18 hr (#223)	4 min 25 sec	Build periodically: @daily
		CAP_BUILD_QA	6 mo 23 days (#47)	16 hr (#244)	3 min 22 sec	Build periodically: #Scheduled build at 2:40 AM 40 2 ***
		CMS2	5 mo 27 days (#5)	1 mo 4 days (#7)	39 sec	
		epehsoft	9 days 7 hr (#95)	9 days 7 hr (#94)	20 min	
		epehsoft_release	N/A	N/A	N/A	
		eReader	N/A	1 mo 4 days (#16)	11 min	
		JSAG Home	8 days 0 hr (#8)	8 days 1 hr (#6)	9.9 sec	
		JSAG-ExceptionHandlingFramework	N/A	N/A	N/A	
		JSAGWSPOC2	7 mo 7 days (#9)	1 mo 4 days (#11)	3 min 12 sec	
		LDrive	5 mo 28 days (#11)	1 mo 4 days (#13)	6 min 50 sec	
		LTAdmin	1 mo 4 days (#6)	2 mo 17 days (#5)	55 sec	
		LTClient	1 mo 4 days (#4)	N/A	54 sec	
		MNTHotelExtranet	1 mo 5 days (#277)	16 hr (#313)	2 min 53 sec	Build periodically: #Every Night at 2 O'clock 0 2 ***

At the bottom, there's a toolbar with icons for various applications like Internet Explorer, Firefox, and Word, along with system status indicators and the date/time (18:57, 27-12-2012).

- On top right, you'll find a "log in" button. Click and enter your domain credentials to login

The screenshot shows the Jenkins login page. The URL is <http://jsag.ggn.nagarro.com/jenkins/login?from=%2Fjenkins%2F>. The page has a sidebar with the same links as the dashboard. The main area contains a login form with fields for User (vaibhav2020) and Password (redacted), a 'Remember me on this computer' checkbox (unchecked), and a 'Log in' button.

At the bottom, there's a link 'Help us localize this page' and footer text 'Page generated: Dec 27, 2012 7:02:21 PM REST API Jenkins ver. 1.485'.

3. After login you'll see page as below

The screenshot shows the Jenkins dashboard for the 'Nagarro Centralise Jenkins System'. The left sidebar includes links for New Job, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Job Config History, Job Import Plugin, and Dependency Graph. The main area displays a table of build jobs:

S	W	Name	Last Success	Last Failure	Last Duration	Cron Trigger
		Backlog_App	1 mo 7 days (#24)	1 hr 51 min (#26)	2 min 30 sec	
		BCR-Android-MobileBanking	1 mo 5 days (#187)	19 hr (#223)	4 min 25 sec	
		CAP_BUILD_QA	6 mo 23 days (#47)	16 hr (#244)	3 min 22 sec	
		CMS2	5 mo 27 days (#5)	1 mo 4 days (#7)	39 sec	
		ephesoft	9 days 7 hr (#95)	9 days 8 hr (#94)	20 min	
		ephesoft_release	N/A	N/A	N/A	
		eReader	N/A	1 mo 4 days (#16)	11 min	
		JSAG Home	8 days 0 hr (#8)	8 days 1 hr (#6)	9.9 sec	
		JSAG-ExceptionHandlingFramework	N/A	N/A	N/A	
		JSAGWSPOC2	7 mo 7 days (#9)	1 mo 4 days (#11)	3 min 12 sec	

4. On top left, you'll see a "New Job" button. Click on New job, and you will get the page shown below

The screenshot shows the 'New Job' configuration page for a 'JSAG-ExceptionHandlingFramework'. The left sidebar is identical to the dashboard. The main area has the following configuration:

- Job name:** JSAG-ExceptionHandlingFramework
- Build a free-style software project** (radio button selected)
 - This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Build a maven2/3 project**
 - Build a maven2 project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Strawboss: External Job Monitor**
 - An extended version of Jenkins external job monitor that allows you to attach publishers to a job monitor. This adds the ability for email notifications and to trigger other job
- Build multi-configuration project**
 - Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Monitor an external job**
 - This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).
- Copy existing Job**
 - Copy from:

At the bottom right, there is an 'OK' button and a link to 'Help us localize this page'. The footer indicates the page was generated on Dec 27, 2012 at 7:10:50 PM, with links to REST API and Jenkins ver. 1.48.

5. Enter the job name as shown in the image above

6. Select "Build a free-style software project" and click on OK button

2.1.2. Add SVN location

Please follow the steps below to configure SVN on Jenkins.

- On clicking OK after entering the job name and type of project, the following page will come

The screenshot shows the Jenkins configuration interface for the 'JSAG-ExceptionHandlingFramework' project. On the left, there's a sidebar with links like Back to Dashboard, Status, Changes, Workspace, Build Now, Delete Project, Configure, Dependency Graph, Job Config History, and All Changes. The main area has sections for 'Project name' (JSAG-ExceptionHandlingFramework), 'Description' (empty), 'Preview' (empty), 'Job Priority' (set to 100), and checkboxes for Discard Old Builds, Disable Automated Maven Repository Cleanup, This build is parameterized, Disable Build (No new builds will be executed until the project is re-enabled.), and Execute concurrent builds if necessary. Below that is a 'JDK' section set to '(Default)'. Under 'Advanced Project Options', there's a 'Source Code Management' section with radio buttons for CVS, CVS Projectset, None, and Subversion, with Subversion selected. There's also a 'Repository URL' input field. At the bottom, there are 'Save' and 'Apply' buttons.

2. You can select discard builds option if you do not want to keep history on Jenkins server
3. Under section “Source code Management”, select subversion to enter the project root URL

2.1.3. Build triggers

Follow the steps below to configure Build triggers:

1. Select the “Build Periodically” option and enter the schedule in the cron syntax or short cut keys supported as shown in the screenshot below. You can click on the help icon shown besides each option to know more details about the option.

Build Triggers

Build after other projects are built ?

Poll SCM ?

Build periodically ?

Schedule ▼

This field follows the syntax of cron (with minor differences). Specifically, each line consists of 5 fields separated by TAB or whitespace:
MINUTE HOUR DOM MONTH DOW
MINUTE Minutes within the hour (0-59)
HOUR The hour of the day (0-23)
DOM The day of the month (1-31)
MONTH The month (1-12)
DOW The day of the week (0-7) where 0 and 7 are Sunday.

To specify multiple values for one field, the following operators are available. In the order of precedence,

- '*' can be used to specify all valid values.
- 'M-N' can be used to specify a range, such as "1-5"
- 'M-N/X' or '*/X' can be used to specify skips of X's value through the range, such as "*/15" in the MINUTE field for "0,15,30,45" and "1-6/2" for "1,3,5"
- 'A,B,...,Z' can be used to specify multiple values, such as "0,30" or "1,3,5"

Empty lines and lines that start with '#' will be ignored as comments.

In addition, '@yearly', '@annually', '@monthly', '@weekly', '@daily', '@midnight', and '@hourly' are supported.

Examples # every minute
* * * * *
every 5 mins past the hour
5 * * * *

2.1.4. Build Step using Maven

Follow the steps given below to configure build steps when using Maven. Your project should have a pom.xml checked in your project's SVN under the project directory for Maven to build your project as per the dependencies and other configuration given in your projects' pom.xml file.

- Under the “Build” section, click on the button “Add build step”. An Options menu will be displayed as shown below:

Build

Add build step ▾

- Execute Windows batch command
- Execute shell**
- Invoke Ant
- Invoke Maven 3
- Invoke Standalone Sonar Analysis
- Invoke Tattletale
- Invoke XShell command
- Invoke top-level Maven targets

- Click on “Invoke Maven 3” to configure Maven settings. Settings shown below are for the “Invoke Maven 3” option.

Build

Invoke Maven 3

Maven Version: Maven3

Root POM:

Goals and options: clean install

Advanced... Delete

3. Enter “clean install” in the Goals field
4. Click on “Advanced” button to add MAVEN_OPTS entries. You can click on the help icon shown besides each option to know more details about the option.

MAVEN_OPTS

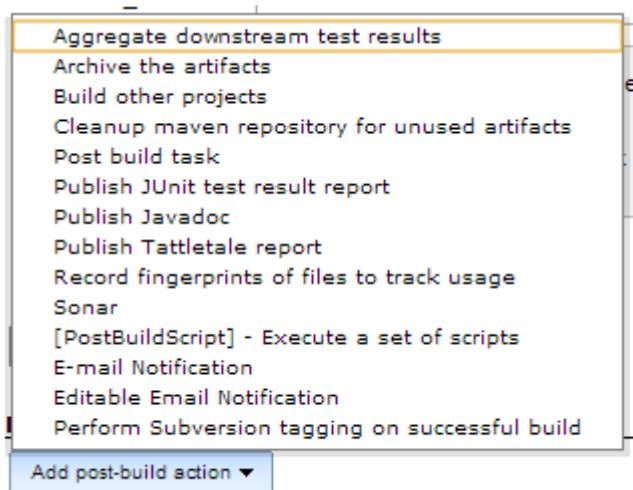
Specify JVM options needed when launching Maven as an external process. Also see [MAVEN_OPTS documentation](#) (even though this is for Maven 1.x it still applies to Maven 2.x)

Shell-like environment variable expansions work in this field, by using the \${VARIABLE} syntax.

Delete

2.1.5. Post-build actions setup

You can select the settings from “Post-build Actions” as per your project requirements, sample settings are show below:



1. Click on “Save” button once all the settings have been selected.
2. Your job setup is complete

3. You can test your build process by clicking on the “Build Now” icon as shown below

The screenshot shows the Jenkins interface for the project "JSAG-ExceptionHandlingFramework". The top navigation bar includes links for Back to Dashboard, Status, Changes, Workspace, Build Now (with a red arrow pointing to it), Delete Project, Configure, Dependency Graph, Job Config History, All Changes, and Sonar. The main content area is titled "Project JSAG-ExceptionHandlingFramework" and contains sections for Sonar, Workspace, and Recent Changes. Below this is a "Permalinks" section with three links: Last build (#1), Last stable build (#1), and Last successful build (#1). On the left, there is a "Build History" section showing one build (#1) from Dec 27, 2012 at 10:10:14 PM, with RSS links for all and failures. At the bottom, there is a link to help localize the page.

4. You can view the progress of the build by selecting the console output as shown below

This screenshot is identical to the one above, showing the Jenkins dashboard for the "JSAG-ExceptionHandlingFramework" project. The "Build History" section is open, displaying build #1 from Dec 27, 2012. A dropdown menu has been opened over the "Console Output" link, which is highlighted with a blue background. Other options in the menu include "Changes", "Edit Build Information", and "Tag this build". The rest of the interface, including the sidebar links and right-hand sidebar, remains the same.

2.2. Manual build execution

A manual build can be executed by clicking on the build now link when inside the project link or by clicking the schedule build as shown below.

S	W	Name	Last Success	Last Failure	Last Duration	Cron Trigger
		Backlog_App	1 mo 7 days (#24)	6 hr 11 min (#26)	2 min 30 sec	
		BCR-Android-MobileBanking	1 mo 5 days (#187)	23 hr (#223)	4 min 25 sec	Build periodically: @daily
		CAP BUILD_QA	6 mo 23 days (#47)	20 hr (#244)	3 min 22 sec	Build periodically: #Scheduled
		CMS2	5 mo 27 days (#5)	1 mo 4 days (#7)	39 sec	
		ephesoft	9 days 11 hr (#95)	9 days 12 hr (#94)	20 min	
		ephesoft_release	N/A	N/A	N/A	
		eReader	N/A	1 mo 4 days (#16)	11 min	
		JSAG Home	8 days 5 hr (#8)	8 days 5 hr (#6)	9.9 sec	
		JSAG-ExceptionHandlingFramework	1 hr 4 min (#2)	N/A	33 sec	
		JSAGWSPOC2	7 mo 7 days (#9)	1 mo 4 days (#11)	3 min 12 sec	
		LDrive	5 mo 28 days (#11)	1 mo 4 days (#13)	6 min 50 sec	

3. Sonar Setup for Jenkins

Sonar is installed on the JSAG server which can be accessed through the URL

<http://jsag.ggn.nagarro.com/sonar> as shown below.

Alert	Name	Version	Lines of code	Rules compliance	Build date
	BacklogApp	1.0-SNAPSHOT	5,913	95.0%	20 Nov 2012
	BCR Mobile APP	1.0	20,170	95.3%	22 Nov 2012
	CMSSalfresco	1.0	608	100.0%	03 Jul 2012
	DCMA: Root	0.0.14-SNAPSHOT	136,634 ▲	89.6% ▲	18 Dec 2012
	JSAGEexceptionFramework	1.0	1,185	100.0%	22:19
	ldrive-root	1.0-SNAPSHOT	129,712 ▲	100.0%	02 Jul 2012
	mystudyweb	0.0.1-SNAPSHOT	15,241 ▲	100.0%	05 Jul 2012
	OppeViewController	0.1-SNAPSHOT	66,542 ▲	100.0%	07 Jun 2012
	ScriptRX	1.0	29,419 ▲	12.8% ▲	15 Jun 2012
	ScriptRXAdmin	1.0	180 ▲	33.9% ▲	15 Jun 2012
	ScriptRXJavaServices	1.0	154,171 ▲	0.0%	07:02
	transport Maven Webapp	1.0-SNAPSHOT	2,337	0.0%	12 Dec 2012
	WebServicesExample Maven Webapp	1.0-SNAPSHOT	64	100.0%	24 May 2012

13 results [Alerts feed](#) com.nagarro.example:SpringWSExample

Follow the steps below to setup Sonar to be integrated with your automated project builds.

3.1. Sonar configuration in Jenkins project build Job

While setting up your project build, select the “Sonar” option in the post-build actions. If not already selected during the project build setup, you can edit the settings by selecting configure option from the Jenkins interface. The settings shown below can be selected for most of the requirements.

Post-build Actions

<input checked="" type="checkbox"/> Sonar	?	
Branch	<input type="text"/>	?
Optional sonar.branch property.		
Language	<input type="text"/>	?
Default is java		
JDK	<input type="text"/> (Inherit From Job)	?
JDK to be used for this sonar analysis		
Maven Version	<input type="text"/> (Inherit From Job)	?
Root POM	<input type="text"/>	?
Default is pom.xml		
MAVEN_OPTS	<input type="text"/>	?
MAVEN_OPTS env var to provide, if not set the plugin will use the MAVEN_OPTS defined by the maven builder config.		
Additional properties	<input type="text"/>	?
Additional properties to be passed to the mvn executable (example: -Dsome.property=some.value).		
<input type="checkbox"/> Dont use global triggers configuration		?
Delete		

Once your project is successfully built with Sonar settings, it will start showing up in the Sonar project list. You can view the dashboard by clicking on the project name as shown below:

Dashboards

Projects	Configuration	Log in	Search																																																																																																		
Treetemap Reviews Dependencies 	<table border="1"> <thead> <tr> <th>Alert</th> <th>Name</th> <th>Version</th> <th>Lines of code</th> <th>Rules compliance</th> <th>Build date</th> <th>Links</th> </tr> </thead> <tbody> <tr> <td></td> <td>BacklogApp</td> <td>1.0-SNAPSHOT</td> <td>5,913</td> <td>95.0%</td> <td>20 Nov 2012</td> <td>View</td> </tr> <tr> <td></td> <td>BCR Mobile APP</td> <td>1.0</td> <td>20,170</td> <td>95.3%</td> <td>22 Nov 2012</td> <td>View</td> </tr> <tr> <td></td> <td>CMSAlfresco</td> <td>1.0</td> <td>608</td> <td>100.0%</td> <td>03 Jul 2012</td> <td>View</td> </tr> <tr> <td></td> <td>DCMA_ Root</td> <td>0.0.14-SNAPSHOT</td> <td>136,634 ▲</td> <td>89.6% ▲</td> <td>18 Dec 2012</td> <td>View</td> </tr> <tr> <td></td> <td>JSAGEExceptionFramework</td> <td>1.0</td> <td>1,185</td> <td>100.0%</td> <td>22:19</td> <td>View</td> </tr> <tr> <td></td> <td>Idrive-root</td> <td>1.0-SNAPSHOT</td> <td>129,712 ▲</td> <td>100.0%</td> <td>02 Jul 2012</td> <td>View</td> </tr> <tr> <td></td> <td>mystudyweb</td> <td>0.0.1-SNAPSHOT</td> <td>15,241 ▲</td> <td>100.0%</td> <td>05 Jul 2012</td> <td>View</td> </tr> <tr> <td></td> <td>OppViewController</td> <td>0.1-SNAPSHOT</td> <td>66,542 ▲</td> <td>100.0%</td> <td>07 Jun 2012</td> <td>View</td> </tr> <tr> <td></td> <td>ScriptRX</td> <td>1.0</td> <td>29,419 ▲</td> <td>12.8% ▲</td> <td>15 Jun 2012</td> <td>View</td> </tr> <tr> <td></td> <td>ScriptRXAdmin</td> <td>1.0</td> <td>180 ▲</td> <td>33.9% ▲</td> <td>15 Jun 2012</td> <td>View</td> </tr> <tr> <td></td> <td>ScriptRXJavaServices</td> <td>1.0</td> <td>154,171 ▲</td> <td>0.0%</td> <td>07:02</td> <td>View</td> </tr> <tr> <td></td> <td>transport Maven Webapp</td> <td>1.0-SNAPSHOT</td> <td>2,337</td> <td>0.0%</td> <td>12 Dec 2012</td> <td>View</td> </tr> <tr> <td></td> <td>WebServicesExample Maven Webapp</td> <td>1.0-SNAPSHOT</td> <td>64</td> <td>100.0%</td> <td>24 May 2012</td> <td>View</td> </tr> </tbody> </table> <p>13 results Alerts feed</p>			Alert	Name	Version	Lines of code	Rules compliance	Build date	Links		BacklogApp	1.0-SNAPSHOT	5,913	95.0%	20 Nov 2012	View		BCR Mobile APP	1.0	20,170	95.3%	22 Nov 2012	View		CMSAlfresco	1.0	608	100.0%	03 Jul 2012	View		DCMA_ Root	0.0.14-SNAPSHOT	136,634 ▲	89.6% ▲	18 Dec 2012	View		JSAGEExceptionFramework	1.0	1,185	100.0%	22:19	View		Idrive-root	1.0-SNAPSHOT	129,712 ▲	100.0%	02 Jul 2012	View		mystudyweb	0.0.1-SNAPSHOT	15,241 ▲	100.0%	05 Jul 2012	View		OppViewController	0.1-SNAPSHOT	66,542 ▲	100.0%	07 Jun 2012	View		ScriptRX	1.0	29,419 ▲	12.8% ▲	15 Jun 2012	View		ScriptRXAdmin	1.0	180 ▲	33.9% ▲	15 Jun 2012	View		ScriptRXJavaServices	1.0	154,171 ▲	0.0%	07:02	View		transport Maven Webapp	1.0-SNAPSHOT	2,337	0.0%	12 Dec 2012	View		WebServicesExample Maven Webapp	1.0-SNAPSHOT	64	100.0%	24 May 2012	View
Alert	Name	Version	Lines of code	Rules compliance	Build date	Links																																																																																															
	BacklogApp	1.0-SNAPSHOT	5,913	95.0%	20 Nov 2012	View																																																																																															
	BCR Mobile APP	1.0	20,170	95.3%	22 Nov 2012	View																																																																																															
	CMSAlfresco	1.0	608	100.0%	03 Jul 2012	View																																																																																															
	DCMA_ Root	0.0.14-SNAPSHOT	136,634 ▲	89.6% ▲	18 Dec 2012	View																																																																																															
	JSAGEExceptionFramework	1.0	1,185	100.0%	22:19	View																																																																																															
	Idrive-root	1.0-SNAPSHOT	129,712 ▲	100.0%	02 Jul 2012	View																																																																																															
	mystudyweb	0.0.1-SNAPSHOT	15,241 ▲	100.0%	05 Jul 2012	View																																																																																															
	OppViewController	0.1-SNAPSHOT	66,542 ▲	100.0%	07 Jun 2012	View																																																																																															
	ScriptRX	1.0	29,419 ▲	12.8% ▲	15 Jun 2012	View																																																																																															
	ScriptRXAdmin	1.0	180 ▲	33.9% ▲	15 Jun 2012	View																																																																																															
	ScriptRXJavaServices	1.0	154,171 ▲	0.0%	07:02	View																																																																																															
	transport Maven Webapp	1.0-SNAPSHOT	2,337	0.0%	12 Dec 2012	View																																																																																															
	WebServicesExample Maven Webapp	1.0-SNAPSHOT	64	100.0%	24 May 2012	View																																																																																															

Powered by [SonarSource](#) - Open Source [GPL](#) - v.3.3.2 - [Plugins](#) - [Documentation](#) - [Ask a question](#)

The dashboard will be available as shown below.

4. Accessing Build and Sonar reports in Jenkins

4.1. Accessing Build when built using Maven

Please follow the steps below to access the build files for a project built using Maven.

- After a successful build, the build files generated can be accessed by going to Jenkins URL <http://jsag.ggn.nagarro.com/jenkins/>, clicking on your project from the home page. The project's home page will look like the one shown below.

- Click on the workspace link to enter the Jenkins workspace for the selected project as shown below.

<http://jsag.ggn.nagarro.com/jenkins/job/JSAG-ExceptionHandlingFramework/>

Jenkins

Jenkins > JSAG-ExceptionHandlingFramework

[Back to Dashboard](#) [Status](#) [Changes](#) [Workspace](#) [Build Now](#) [Delete Project](#) [Configure](#) [Dependency Graph](#) [Job Config History](#) [All Changes](#) [Sonar](#)

Project JSAG-ExceptionHandlingFramework

Sonar
Workspace
Recent Changes

Permalinks

- [Last build \(#3\), 6 min 1 sec ago](#)
- [Last stable build \(#3\), 6 min 1 sec ago](#)
- [Last successful build \(#3\), 6 min 1 sec ago](#)

Build History (trend)

#	Date
#3	Dec 27, 2012 11:23:35 PM
#2	Dec 27, 2012 10:19:00 PM
#1	Dec 27, 2012 10:10:14 PM

[RSS for all](#) [RSS for failures](#)

<http://jsag.ggn.nagarro.com/jenkins/job/JSAG-ExceptionHandlingFramework/ws/>

Jenkins

Jenkins > JSAG-ExceptionHandlingFramework

[Back to Dashboard](#) [Status](#) [Changes](#) [Workspace](#) [Wipe Out Workspace](#) [Build Now](#) [Delete Project](#) [Configure](#) [Dependency Graph](#) [Job Config History](#) [All Changes](#) [Sonar](#)

[.settings](#) [.svn](#) [src/main/java/com/nagarro/jsag/frameworks/exception](#) [target](#) [.classpath](#) [.project](#) [pom.xml](#)

431.00 B [view](#)
576.00 B [view](#)
583.00 B [view](#)

[\(all files in zip\)](#)

Build History (trend)

#	Date
#3	Dec 27, 2012 11:23:35 PM
#2	Dec 27, 2012 10:19:00 PM
#1	Dec 27, 2012 10:10:14 PM

[RSS for all](#) [RSS for failures](#)

3. If built using Maven, the build files will be created inside the target folder as shown below. The files can be downloaded by doing right click → save target as or by clicking (all files in zip) which will download a zip file containing all the files being shown in the current view.

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Dependency Graph

Job Config History

All Changes

Sonar

Build History (trend)

- #3 Dec 27, 2012 11:23:35 PM
- #2 Dec 27, 2012 10:19:00 PM
- #1 Dec 27, 2012 10:10:14 PM

RSS for all RSS for failures

4.2. Accessing Sonar reports

Sonar reports can be accessed by clicking on workspace from the Project home page on Jenkins as shown below.

The screenshot shows the Jenkins workspace interface for the 'JSAG-ExceptionHandlingFramework' job. On the left, there's a sidebar with various project management links like 'Back to Dashboard', 'Status', 'Changes', 'Workspace', etc. A cartoon character icon is also present. Below the sidebar is a 'Build History' section showing three successful builds (#1, #2, #3) from December 27, 2012. The main workspace area shows a folder structure with files and folders: '.settings', '.svn', 'src/main/java/com/nagarro/jsag/frameworks/exception', 'target', '.classpath', '.project', and 'pom.xml'. File sizes are listed next to them: 431.00 B, 576.00 B, and 583.00 B. There's also a link to download all files in a zip archive.

Click on the target folder to see the files and folders inside it as shown below. Sonar creates a pdf report which is shown highlighted below (bcrmobil.pdf).

The report gives details of the code analysis performed by Sonar as shown below:



1. JSAGExceptionFramework

This chapter presents an overview of the project measures. This dashboard shows the most important measures related to project quality, and it provides a good starting point for identifying problems in source code.

1.1. Report Overview

Static Analysis

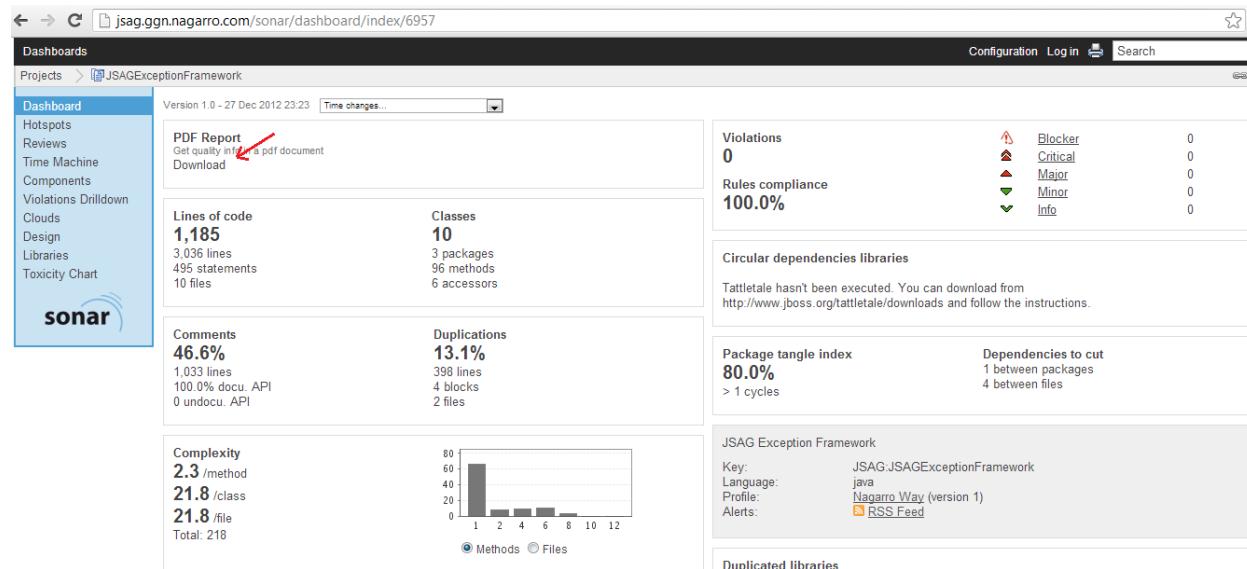
Lines of code	Comments	Complexity
1,185	46.6%	2.3
3 packages	1,033 comment lines	21.8 /class
10 classes		218 decision points
96 methods		
13.1% duplicated lines		

	Sonar PDF Report	JSAGExceptionFramework
---	------------------	------------------------

Most complex files	
ExceptionHelper	92
ExceptionUtil	53
BaseException	27
TechnicalRuntimeException	26
BusinessException	6

Most duplicated files	
TechnicalRuntimeException	210
BaseException	188

The same details can be accessed from the Sonar Web interface (<http://jsag.ggn.nagarro.com/sonar>).



The screenshot shows the Sonar dashboard for the 'JSAGExceptionFramework' project. The top navigation bar includes links for Configuration, Log in, and Search. The dashboard features several cards:

- PDF Report**: A button to download a PDF document.
- Violations**: 0 violations across all severity levels (Blocker, Critical, Major, Minor, Info).
- Rules compliance**: 100.0%.
- Circular dependencies libraries**: Tattletale hasn't been executed; instructions available at <http://www.jboss.org/tattletale/downloads>.
- Package tangle index**: 80.0% (with > 1 cycles).
- Dependencies to cut**: 1 between packages, 4 between files.
- JSAG Exception Framework**: Key: JSAG.JSAGExceptionFramework, Language: java, Profile: Nagarro Way (version 1), Alerts: RSS Feed.
- Duplicated libraries**: A section for identifying duplicate libraries.

On the left, a sidebar lists various Sonar modules: Dashboards, Hotspots, Reviews, Time Machine, Components, Violations Drilldown, Clouds, Design, Libraries, and Toxicity Chart. The 'sonar' logo is also present.

Independent violation details can be accessed by clicking on the violation or the count reported.

Independent results generated by PMD and findbugs can be accessed by clicking on the sonar directory inside the target directory as shown below.

The screenshot shows two levels of Jenkins file browser views. The top view shows the main Jenkins dashboard with a sidebar of links like Back to Dashboard, Status, Changes, Workspace, Build Now, Delete Project, Configure, Dependency Graph, Job Config History, All Changes, and Sonar. The URL is jsag.ggn.nagarro.com/jenkins/job/JSAG-ExceptionHandlingFramework/ws/target/. The bottom view is a detailed file listing under 'target /' with a sub-directory 'sonar' highlighted with a red box. The 'sonar' directory contains files: classes/com/nagarro/jsag/frameworks/exception, maven-archiver, sonar, and surefire. Under 'sonar' are JSAGExceptionFramework.pdf (107.32 KB) and JSAGExceptionFramework-1.0.jar (25.70 KB). A link '(all files in zip)' is also present. The bottom section shows the contents of the 'sonar' directory: checkstyle.xml, findbugs-exclude.xml, findbugs-include.xml, findbugs-result.xml, pmd.xml, and sonar-pom.xml. Each file has its size and a 'view' link. A link '(all files in zip)' is at the bottom of this list.