BLACKDUCK

Installing, Configuring, and Using the Black Duck Suite Plugin for SonarQube

Version 2.2.0

This edition of the *Installing, Configuring, and Using the Black Duck Suite Plugin for SonarQube* refers to version 2.2.0 of the Black Duck Suite Plugin for SonarQube.

This document created or updated on Thursday, July 07, 2016.

Please send your comments and suggestions to:

Black Duck Software, Incorporated
800 District Avenue
Suite 221
Burlington, MA 01803 USA.

Contents

Chapter 1: Black Duck Suite SonarQube Plugin Overview

SonarQube® (formerly Sonar) is an open source platform for continuous code inspection.

Black Duck Code Center provides a comprehensive environment in which to search, request, discuss, approve, and manage internally- and externally-developed components used in software development. Code Center is designed to help software developers quickly and efficiently re-use software components in their applications.

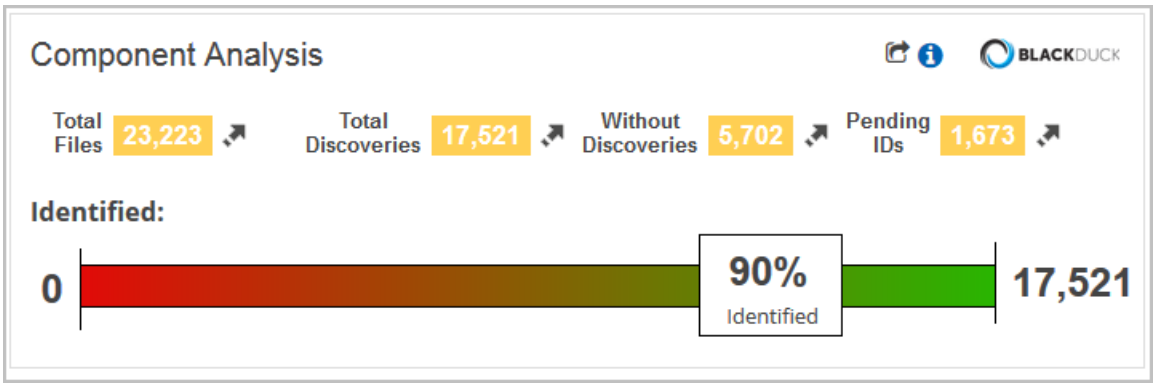
Black Duck Protex helps you reduce business risks and complete software projects on-time and on-budget by automatically analyzing software contents, providing a bill of materials (BOM), uncovering potential risks early in the development cycle, and identifying due diligence-related concerns well in advance of an audit. The goal of using Protex is to ensure that your software does not contain licensing conflicts.

The Black Duck Suite SonarQube plugin provides key open source software data that you can view on your code quality dashboard, enabling visibility and control over the open source components included in your projects. Users can access the approval status of an application's Bill of Materials (BOM), view license information, gain insight into vulnerabilities, and see the progress made in the process of identifying files discovered during code analysis.

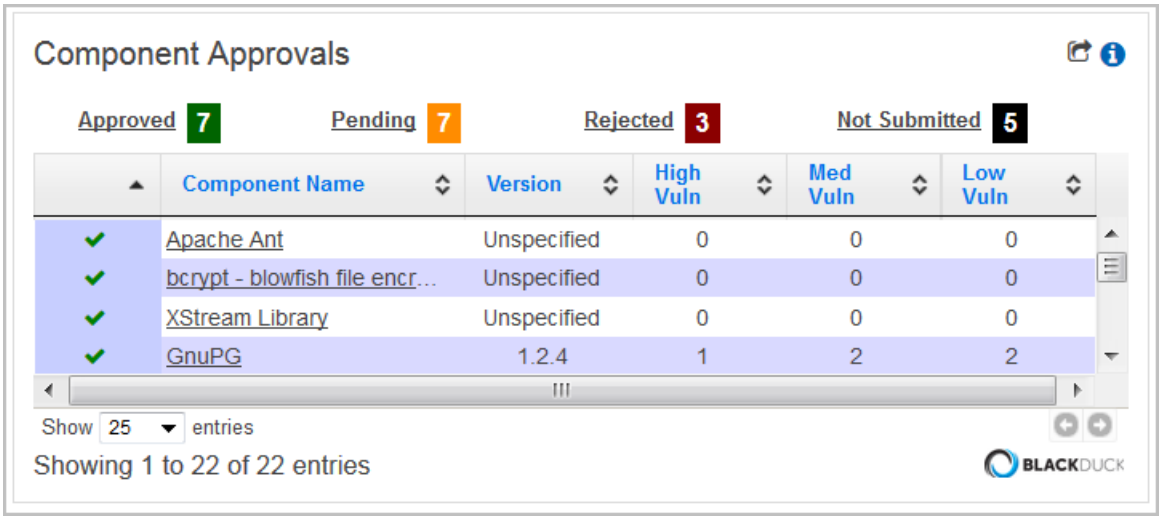
The Black Duck Suite SonarQube plugin provides you with four widgets that you can add to your SonarQube dashboard to view Black Duck data from within the SonarQube interface:

Tip: Each Black Duck widget includes a shortcut link (↗) to launch either Code Center or Protex.

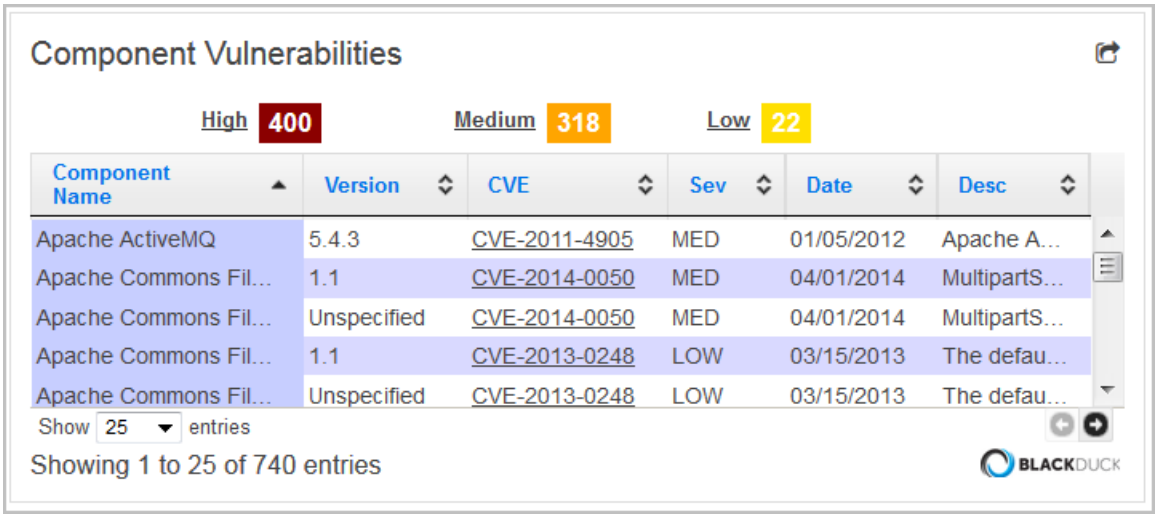
- **Protex Component Analysis** - Provides a summary of Protex analysis data for the project, including an overview of discovered files that have already been identified and resolved to either proprietary or the correct open source project.



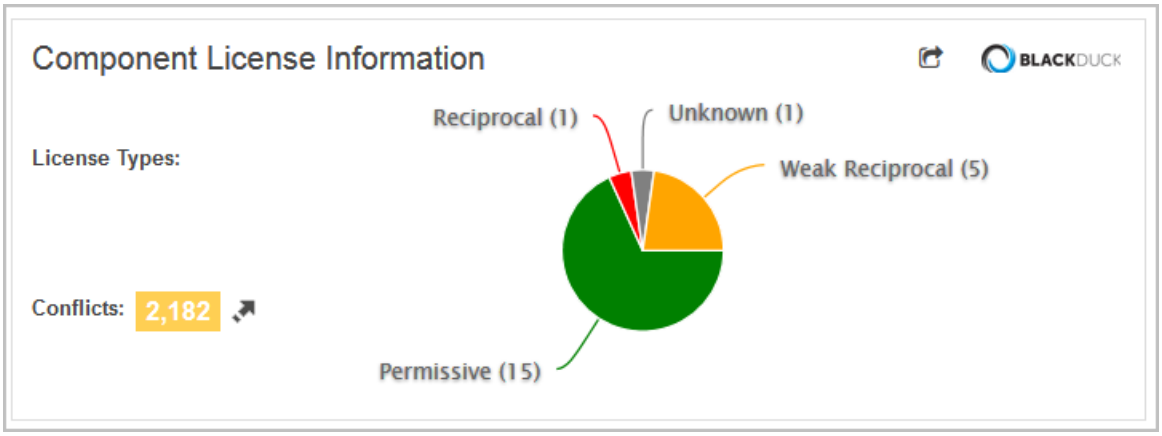
- **Code Center Component Approval** - Provides a list of known project components (the project's Bill of Materials from Code Center), and their approval status.



- **Code Center Vulnerabilities** - Provides a list of Code Center components in the project and vulnerabilities identified for each component; as well as a summary of vulnerabilities by severity: High, Medium, or Low.



- **Code Center License Info** - Provides a pie chart displaying the breakdown of current licenses for the project components. The **Code Center License Info** widget also displays the number of license conflicts with the declared license for the project. The chart displays the licenses in terms of reciprocity:
    - Permissive
    - Unknown
    - Weak Reciprocity
    - Reciprocal



1.1 Black Duck Support

If you have questions or find issues, contact Black Duck Software.

For the latest in web-based support, access the Black Duck Software Customer Support Web Site:
https://www.blackducksoftware.com/support/contact-support

To access a range of informational resources, services and support, as well as access to Black Duck experts, visit the Black Duck Customer Success portal at:
https://www2.blackducksoftware.com/support/customer-success

You can also contact Black Duck Support in the following ways:

- **Email**: support@blackducksoftware.com
- **Phone**: +1 781.891.5100, ext. 5
- **Fax**: +1 781.891.5145
- **Standard working hours**: Monday through Friday 8:00 AM to 8:00 PM EST

Note: Customers on the **Enhanced Customer Support Plan** are able to contact customer support 24 hours a day, 7 days a week to obtain Tier 1 support.

If you are reporting an issue, please include the following information to help us investigate your issue:

- Name and version of the plugin.
- Black Duck product name and version number.
- Third-party integrated product and version; for example, Artifactory, Eclipse, Jenkins, Maven, and others. For Black Duck Hub, only Jenkins, TeamCity, and Bamboo is supported.
- Java version.
- Black Duck KnowledgeBase version, where applicable.
- Operating system and version.
- Source control management system and version.
- If possible, the log files, configuration files, and Project Object Model (POM) XML files.

1.2 SonarQube Installation Overview

To install the SonarQube plugin:

1. Verify that the Java SDK is installed on your system and that other prerequisites (described in the following section) are met.
2. Download the SonarQube plugin.
3. Install the SonarQube plugin.
4. Restart SonarQube so that it recognizes the plugin.
5. Configure the SonarQube plugin.
6. Add a Black Duck widget to one or more SonarQube projects.
7. Re-scan the project using SonarQube Runner.

The following sections include detailed instructions.

1.2.1 SonarQube Installation Prerequisites

Before installing the SonarQube plugin, your server must meet the following software and network requirements and prerequisites:

Software Requirements

The installation instructions in this document assume that you already have the following installed and configured on your system:

- Code Center version 6.6 or greater
- Protex version 6.4.2 or greater
- SonarQube version 5.1 or greater

Network Requirements

Code Center, Protex, and the SonarQube plugin all require internet connectivity. The machine hosting your SonarQube Runner must be able to connect to both your Code Center and Protex servers.

In addition to the hardware and software requirements, the SonarQube plugin has the following prerequisites.

Java SDK

In order to use the Black Duck Suite SonarQube plugin, you must have the Java SDK version 7 installed, and your Java_home must point to the JDK location.

If you do not have the Java SDK installed, you can download it from this site:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Tip: If your SonarQube server is the same server that hosts either your Code Center or your Protex server, then the Java SDK is already installed.

1.2.2 Installing the SonarQube Plugin

To install the SonarQube plugin:

1. Launch your browser.
2. Navigate to the Black Duck Customer Hub:

   https://customerhubblackducksoftware.force.com/login
3. In the menu bar, click **The Exchange**.
4. On the Exchange page, click the link for the SonarQube plugin.
5. On the SonarQube plugin page, click the **Download** tab.
6. On the **Download** tab, click the **Download** URL.
7. After the download completes, locate the downloaded plugin zip file.
8. Unzip the file to a temporary location.
9. Copy or move the plugin JAR file to your SonarQube server:

   Linux - `<SonarQube>/extensions/plugins`

   Windows - `<SonarQube>\extensions\plugins`
10. Restart the SonarQube server to recognize the new plugin file.

1.3 Configuring Settings for the SonarQube Plugin

To configure the SonarQube plugin, you must have a user account for both your Code Center and Protex systems. The Protex user ID must have manager or better permissions.

The configuration for the SonarQube plugin includes:

- **Proxy settings** - This is a global setting that applies to all users of the SonarQube plugin.
- **Code Center server settings** - This is a global setting that links SonarQube to your Code Center server.
- **Code Center project settings** - This is a project-specific setting that maps your SonarQube project to the corresponding Code Center application.
- **Protex server settings** - You can configure the Protex server settings at either a global or a project-level, which allows you to connect to more than one Protex server. The SonarQube plugin checks the project-level setting first, and if that value is blank, uses the global setting.
- **Protex project settings** - This is a project-level setting that maps your SonarQube project to the corresponding Protex project.

1.3.1 Configuring a Proxy

You must have administrator privileges in SonarQube to configure the global plugin settings.

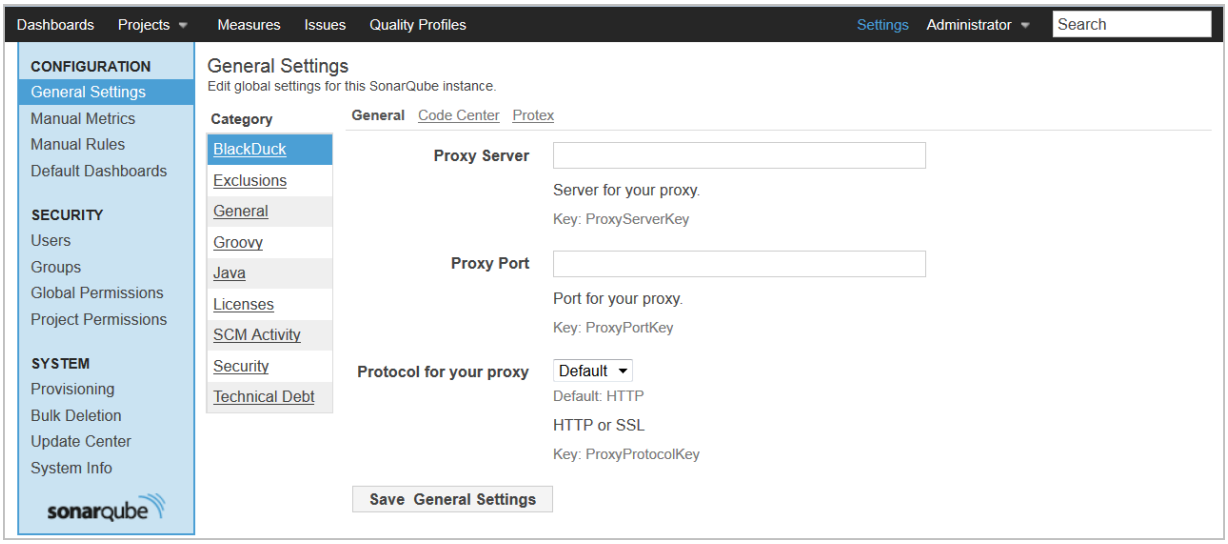
To configure the SonarQube plugin:

1. Launch your browser and navigate to your SonarQube home page. If you are on the SonarQube machine, the URL might be similar to:

```
http://localhost:9000
```

2. Log in to SonarQube as a user with administrator privileges.
3. In the toolbar, click the **Settings** link.
4. On the **Configuration** > **General Settings** page, click the **Black Duck** link in the **Category** list.

Figure 1.1: SonarQube General Settings > Black Duck General Settings



5. If necessary, on the **Black Duck General Settings** page, click the **General** link.
6. In the **Proxy server** field, enter the name of the web proxy host.
7. In the **Proxy port** field, enter the port used by the web proxy.
8. From the **Protocol for your proxy** menu, select the schema: HTTP, HTTPS, or SSL.
9. Click **Save General Settings**.

1.3.2 Configuring the Code Center Server

You must have administrator privileges in SonarQube to configure the Code Center server connection details.

To configure the Code Center server connection:

1. Launch your browser and navigate to your SonarQube home page. If you are on the SonarQube machine, the URL might be similar to:

```
http://localhost:9000
```

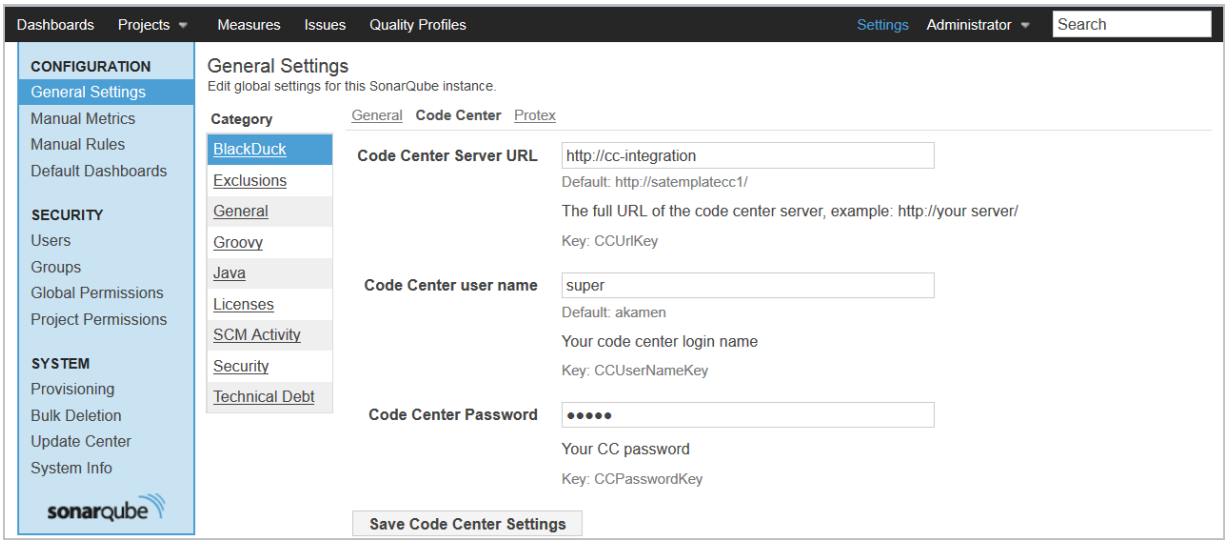
2. Log in to SonarQube as a user with administrator privileges, or use the default:

User name = admin

Password = admin

3. In the toolbar, click the **Settings** link.
4. On the **Configuration** > **General Settings** page, click the **Black Duck** link in the **Category** list.
5. On the **Black Duck General Settings** page, click the **Code Center** link.

Figure 1.2: SonarQube General Settings > Black Duck Code Center Settings



6. In the **Code Center Server URL** field, enter the URL of your Code Center server. For example, `http://<codecenterserver>`.
7. In the **Code Center user name** field, enter the user name you use to access your Protex server. For example, `admin`.
8. In the **Code Center Password** field, enter your password.
9. Click **Save Code Center Settings**.

1.3.3 Configuring the Protex Server Globally

You must have administrator privileges in SonarQube to configure the Protex server connection details. The Protex user must have manager permissions in Protex.

The SonarQube plugin checks the project level setting first, and if that value is blank, it uses the global setting to connect to the Protex server.

✱ To configure the SonarQube plugin:

1. Launch your browser and navigate to your SonarQube home page. If you are on the SonarQube machine, the URL might be similar to:

`http://localhost:9000`

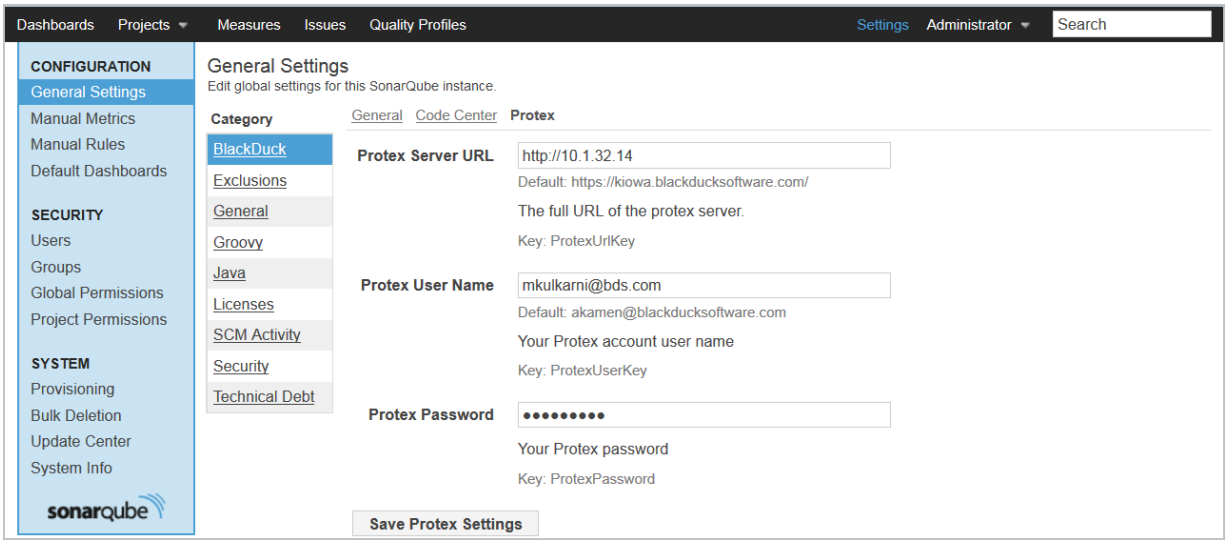
2. Log in to SonarQube as a user with administrator privileges, or use the default:

User name = admin

Password = admin

3. In the toolbar, click the **Settings** link.
4. On the **Configuration** > **General Settings** page, click the **Black Duck** link in the **Category** list.

Figure 1.3: SonarQube General Settings > Black Duck Protex Settings



5. On the **Black Duck General Settings** page, click the **Protex** link.
6. In the **Protex Server URL** field, enter the address used to access your Protex server. For example, `http://<protexserver>` .
7. In the **Protex User Name** field, enter the name you use to access your Protex server. For example, `admin@blackducksoftware.com`.
8. In the **Protex Password** field, enter your password.
9. Click **Save Protex Settings**.

1.3.4 Configuring the Code Center Application

For the SonarQube plugin to display Code Center data, you must map your projects in SonarQube to the appropriate applications within Code Center.

To configure Code Center project settings:

1. In SonarQube, select the project from the **Projects** list (**Projects** > **All Projects**).
2. From the **Configuration** menu, select **Settings**.
3. On the **Settings** page, click the **Black Duck** category.
4. On the **Black Duck project settings** page, click the **Code Center** link.
5. In the **Code Center App Name** field, enter the name of the project in Code Center. This name must match the value in the **Name** field for the application record in Code Center.

6. In the **Version of your Code Center Application** field, enter the version number of the application. This value must match the value in the **Version** field on the application record in Code Center, not the component version.
7. Click **Save Code Center Settings**.

1.3.5 Configuring the Protex Server and Project

For the Black Duck SonarQube plugin to display Protex data, you must map your projects in SonarQube to your projects in Protex.

If you configure a Protex server value at the project level, it overrides the global value. The SonarQube plugin checks the project level setting first, and if that value is blank, it uses the global setting. This lets you connect to more than one Protex server.

If you have already configured Code Center settings for a project, the plugin can use that value to look up the Protex project.

✱ **To configure Protex project settings:**

1. In SonarQube, select the project from the **Projects** list (**Projects** > **All Projects**).
2. From the Configuration menu, select **Settings**.
3. On the **Settings** page, click the **Black Duck** Category.
4. On the **Black Duck project settings** page, click the **Protex** link.
5. (Optional) In the **Protex Server URL** field, enter the full URL for the Protex server.
6. (Optional) In the **Protex User Name** field, enter your Protex user name.
7. (Optional) In the **Protex Password** field, enter your Protex password.
8. (Optional) In the **Name of Protex Project** field, enter the name of the project in Protex. If you have already configured the Code Center project name, you can leave this field blank. The plugin automatically finds the Protex project name, based on the Code Center application name.
9. Click **Save Protex Settings**.

1.4 Using the SonarQube Plugin

You can add one or more Black Duck widgets to single or multiple projects monitored by SonarQube.

1.4.1 Adding a Black Duck Widget to your Dashboard

You must be an administrator in SonarQube to add Black Duck widgets to dashboards.

✱ **To add a Black Duck widget to a project dashboard:**

1. Log in to SonarQube.
2. On the SonarQube home page, click the **Project** link in the menu and select **All Projects**.
3. Select the project to which you want to add the widget.
4. On the project page, click the **Configure widgets** link located below the **Search** field. A yellow

widget configuration area opens at the top of the page.

5. In the **Category** list, click **Black Duck**. The screen refreshes to show only the Black Duck widget options.
6. To add the widget to the project, click **Add widget** for the specific widget. The widget is added to the project.

Tip: You can drag and drop the widgets to change where they appear on the page.

7. To view the widget, click the **Back to Dashboard** link.

1.4.2 Rescan the Project with SonarQube Runner

For the Black Duck widgets to display data, you must re-run the analysis in SonarQube. The frequency with which your SonarQube system is configured to run code analysis varies, depending on your business needs. For example, if you are running continuous integration, you may only scan code once a day.

You can trigger a manual scan of a project using SonarQube Runner. Refer to the SonarQube documentation for information on how to install, configure, and use SonarQube Runner.

1.5 Blackduck Suite SonarQube Best Practices

1.5.1 Job Configuration

SonarQube performs analysis of the project and requires a Maven goal to invoke the analysis of the project. However, Protex scans are post-build operations, and Code Center is integrated into the build environment. Therefore, if the SonarQube goal of *sonar:sonar* is invoked in the build step, the Code Center and Protex projects collect metrics during that iteration, but SonarQube displays the information of the last execution of the Code Center and Protex Jenkins plugin on the project. It does not display the data of the latest build. Because of this, you do not see the latest analysis information in SonarQube for the project. To ensure that the SonarQube project displays the analysis results of the latest build, it is recommended to run the SonarQube Maven goals as a post-build operation after both the build and the Protex Scan post-build action.

To do this in Jenkins, it is recommended to execute the Maven goal as a post-build step.

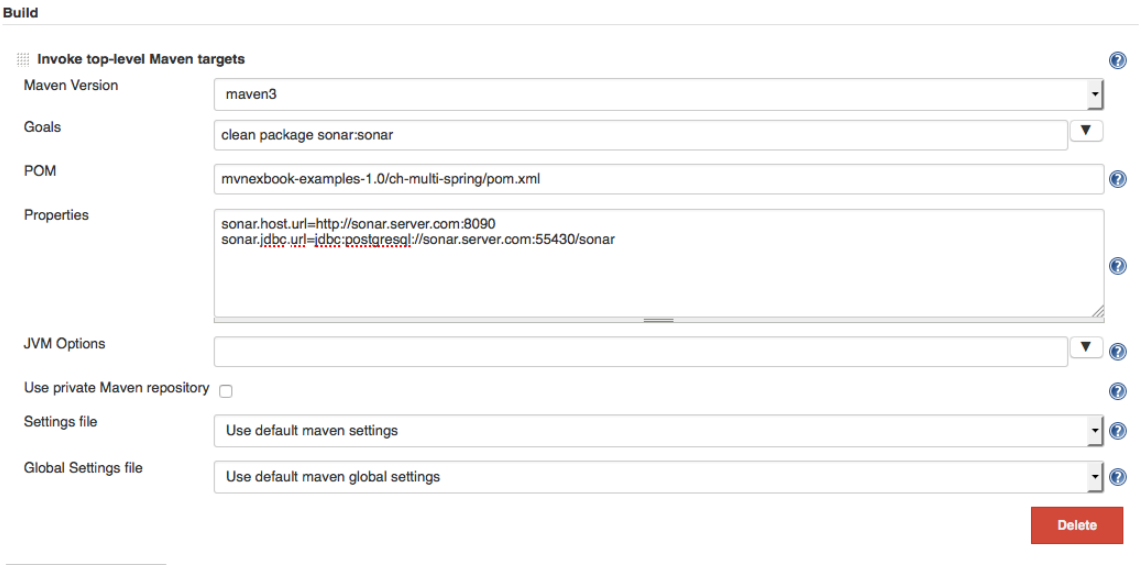
1.5.2 Invoking a Downstream Job

With the *Parameterized Trigger* plugin, you can trigger the build of another job as a post-build step. The invoked job is configured to execute a Maven build as the build step for the project and provide the parameters. This has the benefit of using the Maven tools configured in Jenkins so that you do not need to know the path to the Maven tool or be concerned about the slave on which the build is running. It is much easier to specify the properties to define the SonarQube server and database in use as well.

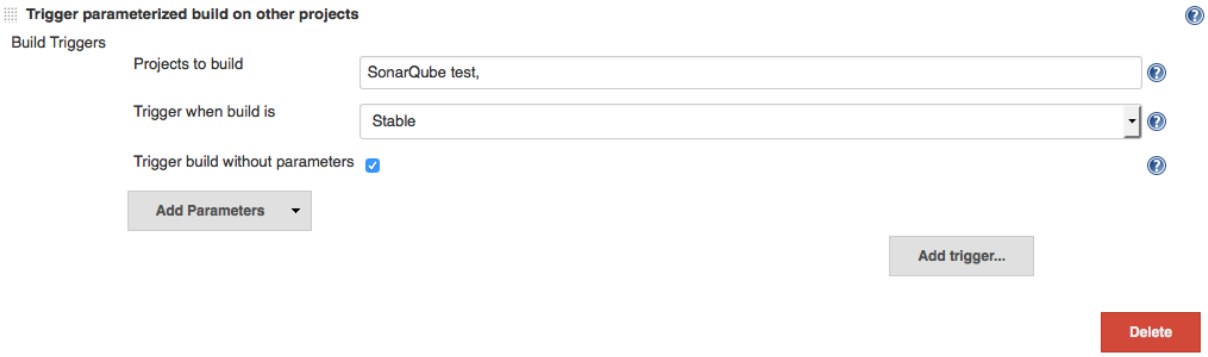
To invoke a downstream job:

1. Configure a job to execute the *sonar:sonar* Maven goal against the appropriate project.
2. Configure the build step.
3. **Goals** must have *sonar:sonar* defined as one of the goals for analysis to be run.

4. **Properties** must have:
   a. `sonar.host.url` (the SonarQube server URL).
   b. `sonar.jdbc.url` (the SonarQube JDBC URL).



5. Add a post-build step to the job that executes the Code Center and/or Protex Scan to **Trigger a parameterized build on other projects**. This step must be executed after the Protex scan post-build action to ensure the latest information is retrieved by SonarQube.
6. If you have no parameters to pass to the job, select the check box **Trigger build without parameters**.



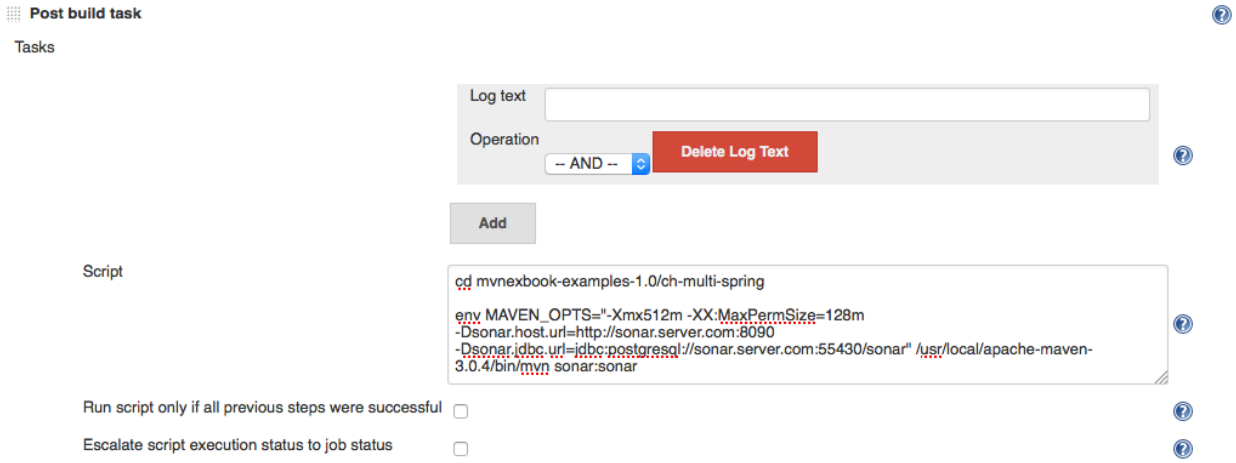
1.5.3 Post-build task

This is another mechanism that can be invoked; however, you must know the details of the Maven executable running on a particular slave to get the path correct. This is possible, but not the most widely recommended approach.

To complete the post-build task:

1. Configure a script to execute.
2. Call the `cd` command to navigate to the directory of the POM file to execute to build the project.
3. Define the single command line to invoke the Maven *sonar:sonar* goal on the project. In this command line, you must know the path of the Maven executable to analyze the project; for example, `/usr/local/apache-Maven-3.0.4/bin/mvn`.
4. Set `MAVEN_OPTS` to have the SonarQube variables for connecting to the server and the database in use:
    a. `-Dsonar.host.url` (the SonarQube server URL).
    b. `-Dsonar.jdbc.url` (the SonarQube database URL).

For example: `env MAVEN_OPTS="-Dsonar.host.url=http://sonar.server.com:8090 -Dsonar.jdbc.url=jdbc:postgresql://sonar.server.com:55430" <Path to M2_HOME>/bin/mvn sonar:sonar`.



1.5.4 SonarQube Project Settings

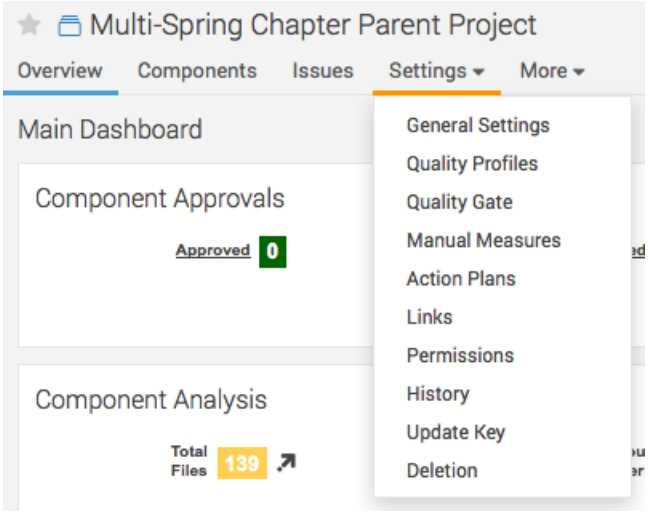
Once the initial Jenkins build has been run with the Maven goal *sonar:sonar*, the project will exist in the SonarQube server. You must log into the SonarQube dashboard and configure project settings to display accurate data with the project.

Note: This assumes that you have already installed the Black Duck SonarQube plugin, and configured SonarQube with the global Black Duck Code Center and Protex settings.

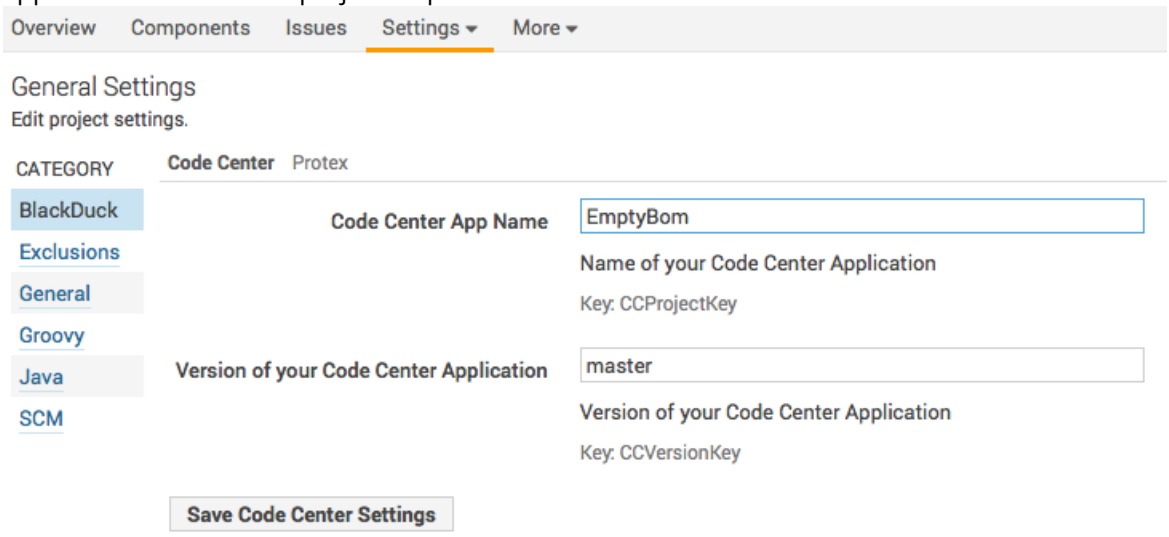
To configure the SonarQube project settings:

1. Log in to SonarQube.
2. Select the project from the dashboard.
3. Click the **Settings** drop down control.

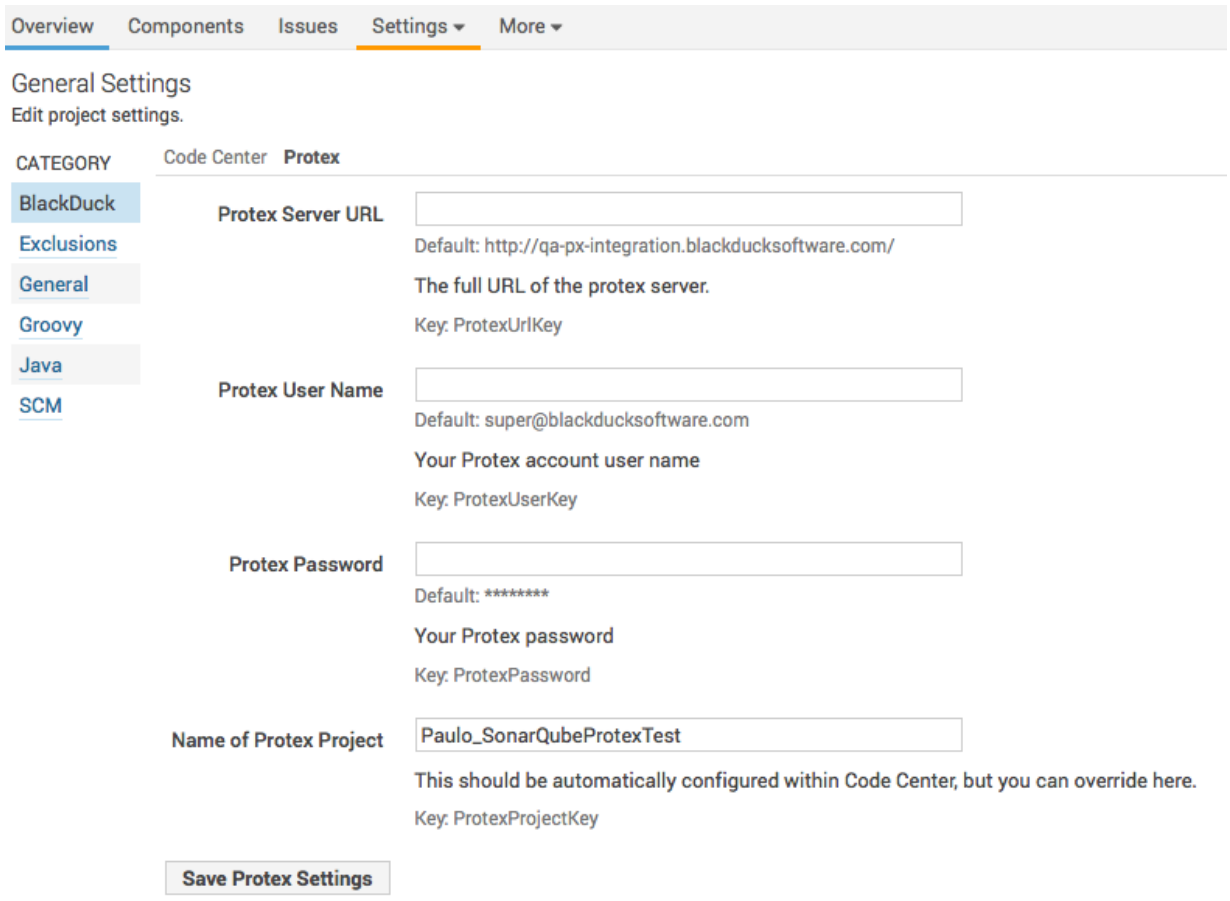
4. Select **General Settings**.



5. Configure the Code Center settings for the project.
6. In the **Code Center App Name** field, type the Code Center application to which this project maps.
7. In the **Version of your Code Center Application** field, type the version string for the Code Center application to which this project maps.



8. Configure the Protex settings.
9. Configure the server settings if they are different than the global settings.
10. In the **Name of Protex Project** field, enter the name of the Protex project to which this project maps.



1.6 Troubleshooting the SonarQube Plugin

Symptom: Widget displays an error message similar to the following: *An error occurred while trying to display the widget <widgetname>. Please contact the Administrator.*

This error message may have multiple causes:

Cause: Either the Code Center server or Protex server or both have not been configured.

Solution: Configure the Code Center server or Configure the Protex server.

Cause: The project name in SonarQube does not match the project name in Protex.

Solution: Change the project name to match the project name in Protex.

Cause: The project has not been scanned since the Black Duck plugin was installed and the widgets were added to the project.

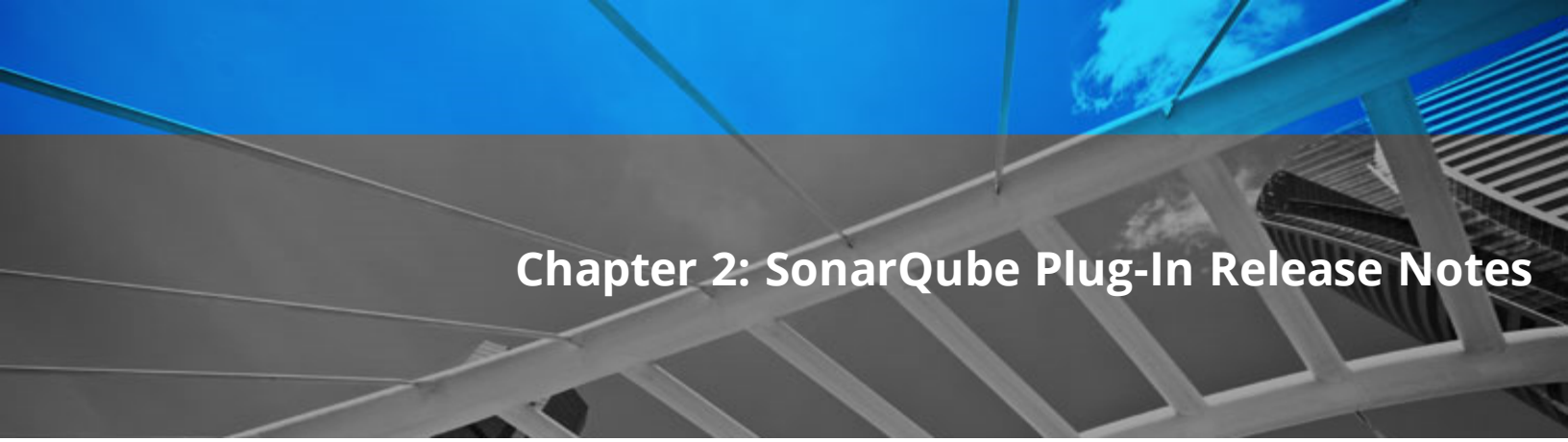
Solution: Scan the project using SonarQube Runner.

Symptom: During testing, the **Protex Component Analysis** graph could show -1% files identified when no identifications are made.

Cause: This is a known issue in the calculations of the totals, and will be addressed in a future release.

Workaround: Click the shortcut link to launch Protex to view your project's identification status.

Tip: After major releases of Code Center or Protex, check to see if there are updated versions of your Black Duck plugins. Changes to the APIs, schema, and SDK versions may require updated versions of the integration plugins.



Chapter 2: SonarQube Plug-In Release Notes

Version 2.1.0

New Features

- SonarQube is now upgraded to work with Java Software Developer Kit version 7.0.

Resolved Issues

- **Notice Report** tool now runs without exception errors.
- Blank error messages no longer display.
- The configuration is now verified prior to launching the plugin.
- The **Component Analysis** gradient slider has been improved.
- Errant project dashboard hyperlinks have been addressed.
- Addressed an issue wherein a *Could not properly authenticate Code Center* error displayed incorrectly.
- Addressed an issue wherein the **Percentage** box did not display correctly if Protex had zero identified components.
- Addressed an issue with applications containing other applications (*rolled-up* scenarios), the **Component** object displayed a null application ID.
- Invalid URLs no longer fail a build.
- Modified custom attributes now display correctly.
- Addressed an issue wherein the component analysis links were broken.
- Previously, the Code Center link was not directing the user to the Bill of Materials; this has been resolved.
- Scanned results from Code Center and Protex now display correctly.
- URLs entered in **General Settings** for Code Center and Protex servers no longer require a trailing forward slash.
- The value for **Identified %** now displays correctly.

Version 2.0.0

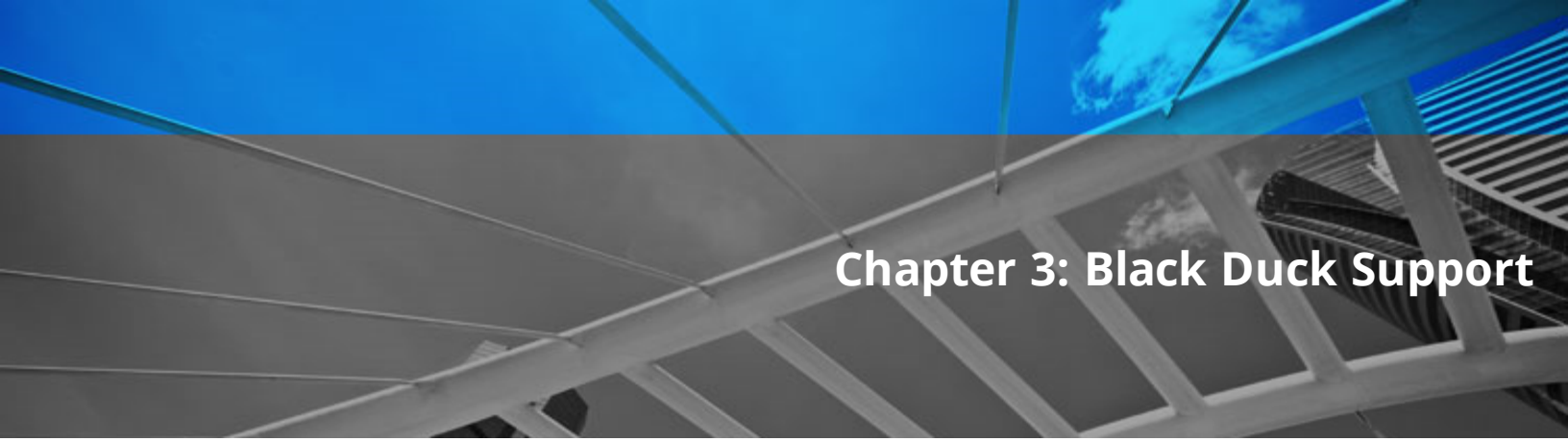
Resolved Issues

- Requires SonarQube version 5.1 or higher.
- Supports new JavaScript regimen of SonarQube 5.1.
- Improved browser compatibility.

Version 1.2.2

Resolved Issues

- Fixed an issue where correct passwords were not being properly authenticated.



Chapter 3: Black Duck Support

If you have questions or find issues, contact Black Duck Software.

For the latest in web-based support, access the Black Duck Software Customer Support Web Site: https://www.blackducksoftware.com/support/contact-support

To access a range of informational resources, services and support, as well as access to Black Duck experts, visit the Black Duck Customer Success portal at: https://www2.blackducksoftware.com/support/customer-success

You can also contact Black Duck Support in the following ways:

- **Email**: support@blackducksoftware.com
- **Phone**: +1 781.891.5100, ext. 5
- **Fax**: +1 781.891.5145
- **Standard working hours**: Monday through Friday 8:00 AM to 8:00 PM EST

Note: Customers on the **Enhanced Customer Support Plan** are able to contact customer support 24 hours a day, 7 days a week to obtain Tier 1 support.

If you are reporting an issue, please include the following information to help us investigate your issue:

- Name and version of the plugin.
- Black Duck product name and version number.
- Third-party integrated product and version; for example, Artifactory, Eclipse, Jenkins, Maven, and others. For Black Duck Hub, only Jenkins, TeamCity, and Bamboo is supported.
- Java version.
- Black Duck KnowledgeBase version, where applicable.
- Operating system and version.
- Source control management system and version.
- If possible, the log files, configuration files, and Project Object Model (POM) XML files.

3.1 Training

Black Duck training courses are available for purchase. Learn more at https://www.blackducksoftware.com/services/training.

View the full catalog of our online offerings: https://www.blackducksoftware.com/academy-catalog.

When you are ready to learn, you can log in or sign up for an account: https://www.blackducksoftware.com/academy.

3.2 Services

If you would like someone to perform Black Duck Software tasks for you, please contact the Black Duck Services group. They offer a full range of services, from planning, to implementation, to analysis. They also offer a variety of training options on all Black Duck products. Refer to https://www.blackducksoftware.com/services/ for more information.