BLACKDUCK

Hub Plugin for JIRA

Version 3.2.0

This edition of the *Hub Plugin for JIRA* refers to version 3.2.0 of the Black Duck Hub Plugin for JIRA.

This document created or updated on Monday, April 03, 2017.

Please send your comments and suggestions to:

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

Contents

Chapter 1: Hub JIRA Plugin Overview

Introduction

The Hub JIRA plugin is a JIRA add-on that enables organizations to use JIRA to manage and track issues detected by Black Duck Hub that are related to your use of open source software. Black Duck Hub (referred to as *the Hub*) is a risk management tool, designed to help you manage the logistics of using open source software in your organization. JIRA is an issue tracking application that enables software development organizations to track and manage issues related to the software applications they are developing.

Purpose

The Hub JIRA plugin is designed for organizations that use JIRA and want to manage open source-related issues within JIRA, the same way you track other software development related issues. It enables you to use the Hub to detect open source security risks, compliance issues, and policy violations, and to use JIRA to track those issues through the various steps required to investigate and resolve each one. The Hub generates notifications as important events occur, and the Hub JIRA plugin reads those notifications from the Hub. Some examples of notification events are:

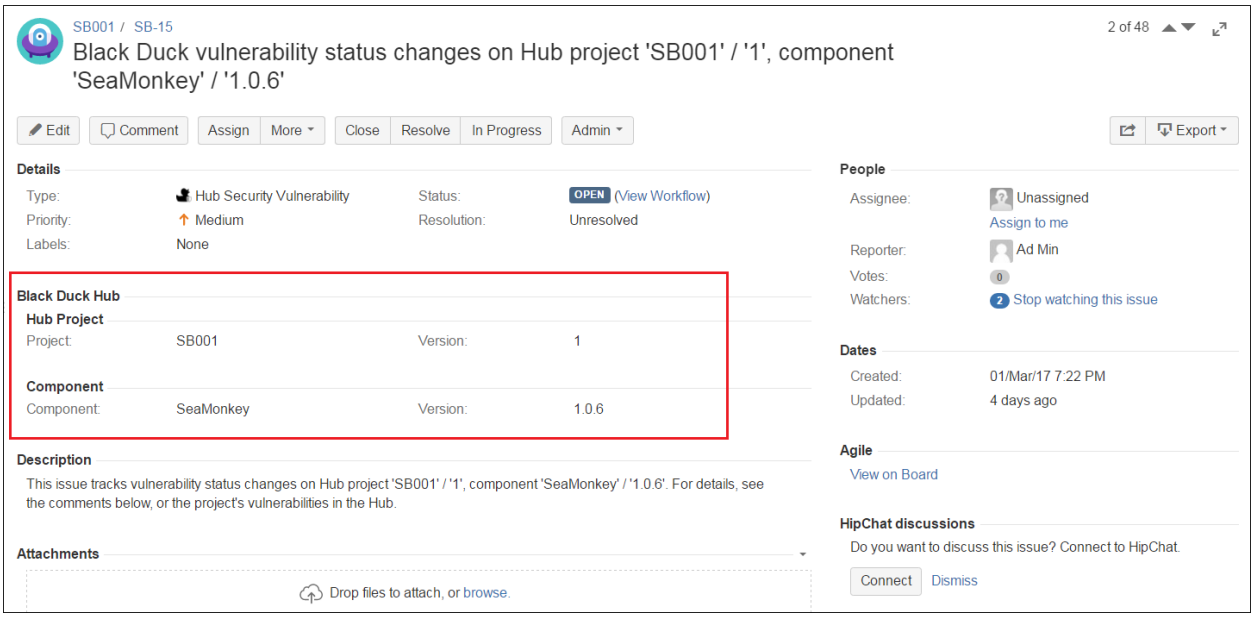
- The introduction of a component containing a known security vulnerability into a project's Bill of Materials.
- The introduction of a component containing a policy violation into a project's Bill of Materials.
- The manual override of a policy violation on a component.

In response to these notification events, the Hub JIRA plugin:

- Creates a JIRA ticket to the track the component issue; whether security vulnerability, policy violation, or other.
- Assists in resolving the corresponding JIRA issues.

After the Hub JIRA plugin has created an issue in JIRA, you can take advantage of JIRA's capabilities to track and manage those issues. Issue-specific metadata is pulled from the Hub into each plugin-created JIRA ticket, providing access to that data through JIRA's search and reporting capabilities.

The following is an example of a JIRA issue with the added Black Duck Hub information.



1.1 Feature Support by Hub Version

The following Hub JIRA features are supported per the specified Black Duck Hub version.

1.1.1 Links to Hub pages

Vulnerable Components

These links appear in the descriptions of vulnerability issues created with version 3.0.1 (or later) of the plugin.

- When the user is already logged into the Hub in the same browser: since Hub 3.4.

Project version

These links appear in the description of all issues created with version 3.1.0 (or later) of the plugin.

- When the user is already logged into the Hub in the same browser: since Hub 3.4.2.

1.1.2 Hub Policy Violation Notifications

- Create a JIRA issue on a new Hub policy violation: since Hub 3.0.
- Resolve a JIRA issue when a violation is overridden: since Hub 3.2.0.
- Re-open a JIRA issue when a violation returns: since Hub 3.2.0.
- Resolve a JIRA issue when a violation is cleared: since Hub 3.3.1.

1.1.3 Hub Vulnerability Notifications

- Create a JIRA issue on the first vulnerability status change: since Hub 3.3.1.
- Add a JIRA issue comment on each vulnerability status change: since Hub 3.3.1.

- Loading (clicking on) a *vulnerable-bom-components* link from a browser loads the project version's **Vulnerability** view in the Hub: since Hub 3.4.0.

Chapter 2: Installation Overview

The following topics discuss downloading and installation of the Hub JIRA plugin.

Note: You can find JIRA documentation at:
https://confluence.atlassian.com/jira/jira-documentation-1556.html

2.1 Installation Prerequisites

Before you install the Hub JIRA plugin, ensure that:

- Your JIRA instance is up-to-date and fully patched.
- You know the host name and port for the Hub server.
- You have a user account on the Hub system that you can use for the integration.
- You have connectivity to the internet. The machine that hosts your JIRA server must be able to connect to the Hub server.
- As of Hub JIRA plugin versions 3.2.0 and higher, the Hub Admin plugin is no longer required, and should be uninstalled where applicable.
- To view the steps for installing a JIRA plugin, refer to Downloading and Installing the Hub JIRA Plugin on page 8.

2.2 Software Version Prerequisites

Prior to installing the Hub JIRA plugin, you must have the following software with the specified versions.

- Atlassian JIRA versions 6.4 - 7.0.11 for Hub JIRA 2.0.x. Support for these versions will soon be end of life.
- Atlassian JIRA versions 7.1.0 - 7.3.x for Hub JIRA 3.0.x and higher.
- Black Duck Hub 3.4 or higher.

For more information, refer to Feature Support by Hub Version on page 6.

2.3 Downloading and Installing the Hub JIRA Plugin

You can access the Hub JIRA plugin in two ways; both are described as follows.

✱ To manually upload the Hub JIRA plugin:

1. Log into JIRA as a system administrator.
2. Navigate to the **Add-ons** section.
3. Then, navigate to **Manage add-ons**.

4. Click **Upload add-on**, and select the *hub-JIRA* jar file.
5. Click **Upload**.

✱ **To download the Hub JIRA plugin from the Atlassian Marketplace:**

1. Log into JIRA as a system administrator.
2. Navigate to the **Add-ons** section. Or, click the drop-down menu appearing below your user profile icon on the far right of the JIRA taskbar, and select **Atlassian Marketplace**.
3. Then, navigate to **Find new add-ons**.
4. Alternatively, you can search for *Black Duck Hub* using the **Atlassian Marketplace** *Search* field.
5. Locate the **Black Duck Hub JIRA plugin**.
6. Click **Install**. The **Black Duck Hub JIRA plugin** automatically downloads and installs.
7. When the installation is complete, the **Installed and ready to go!** status message displays.

2.4 Updating the Hub JIRA plugin

You can update the Hub JIRA plugin as new versions are released.

✱ **To update the Hub JIRA plugin:**

1. Navigate to **Administration** > **Add-ons** > **Manage add-ons**.
2. Select **Black Duck Hub JIRA plugin**.
    a. If there are updates for **Black Duck Hub JIRA plugin**, the updates display in the list.
    b. Alternatively, you can force JIRA to check for plugin updates by clicking **Check now**.
3. If there are updates, select the one you want, and click **Update**.

Chapter 3: Configuring the Hub JIRA Plugin

After downloading and installing the Hub JIRA plugin, you can configure it for your Hub environment. Configuration options are discussed in the following sections.

3.1 User Access and Configuration

If non-system administrator users are not allowed to configure the Hub JIRA plugin; in other words, not allowed to access the plugin configuration page, then the administrator user must perform all of the configuration procedures outlined in the following sections.

- Hub JIRA Initial Setup
- Configuring Logging for the Hub JIRA Plugin
- Configuring the Hub Server
- Issue Creation Configuration
- Issue Field Configuration for the Hub JIRA Plugin
- Viewing and Resetting Error Messages
- Disabling Automatic Resolving and Re-Opening of Issues

However, if non-system administrator users are allowed to configure the Hub JIRA plugin, then the administrator user does not have to execute the procedures described in:

- Configuring the Hub Server
- Issue Creation Configuration
- Issue Field Configuration

3.1.1 Initial Setup for Hub JIRA

After you have installed the Hub JIRA plugin, use the following process to set up your Hub JIRA plugin.

To set up your Hub JIRA plugin:

1. Log into JIRA as a system administrator.
2. If you plan to allow non-system administrator users to configure the Hub JIRA plugin, then determine which JIRA user group(s) will be allowed to configure the Hub JIRA plugin, and make sure all users that should be allowed to configure the Hub JIRA plugin are assigned to those groups. These groups are referred to as *configuration-enabled groups*, and users in these groups are known as *configuration-enabled users*.
3. At the far right of the toolbar at the top of the page, click the gear icon to display the **JIRA ADMINISTRATION** drop-down menu.
4. In the **JIRA ADMINISTRATION** drop-down menu, select **Add-ons**. The **Administration** > **Add-ons**

page displays. Note that you may be required to re-enter your administrator password.

5. In the left navigation panel of the **Administration** > **Add-ons** page, select **Hub JIRA**.
6. In the **Add-ons** page for Hub JIRA, select the **Configuration Access** tab.
7. If you plan to allow non-system administrator users to configure the Hub JIRA plugin, then navigate to the **User Access to Hub JIRA** configuration section section, and in the **Groups that can access the Hub JIRA configuration** field, select the configuration-enabled groups. Click in the field to see all available groups; you can select multiple groups.

8. Click **Save**.
9. If you do not plan to allow non-system administrator users to configure the Hub JIRA plugin, then configure the plugin as described in Configuring Your Hub JIRA Plugin.

Important: Ensure that you click **Save** after completing each tab. Otherwise, your configuration and settings are not applied, and may cause errors when generating tickets and issues.

3.2 Configuration by a Configuration-Enabled User

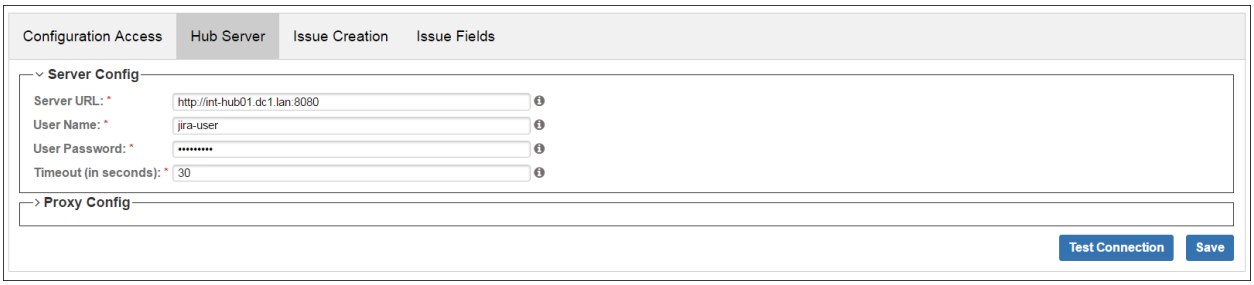
Configuration-enabled users can configure the Hub JIRA plugin by accessing the URL: `(JIRA url)/plugins/servlet/hub-jira-integration`.
Configuration-enabled users are defined as users that have been granted access to the configuration page by the administrator user.

3.3 Configuring the Hub Server

You can add your Hub configuration to the Hub JIRA plugin as follows.

To add a Hub server configuration:

1. At the far right of the toolbar at the top of the page, click the gear icon to display the **JIRA ADMINISTRATION** drop-down menu.
2. In the **JIRA ADMINISTRATION** drop-down menu, select **Add-ons**. The **Administration** > **Add-ons** page displays. Note that you may be required to re-enter your administrator password.
3. In the left navigation panel of the **Administration** > **Add-ons** page, select **Hub JIRA**.
4. In the **Add-ons** page for Hub JIRA, select the **Hub Server** tab.



5. **Server URL**: Type the URL for your Hub server instance.
6. **User Name**: Type your Hub user name.
7. **User Password**: Type your Hub password.
8. **Timeout (in seconds)**: Type the number of seconds for timing out; the default value is 30.
9. Click **Test Connection**. If the connection is successful, click **Save**.

3.4 Configuring a Proxy

Note: Black Duck does not currently support proxies with authentication when connecting to an HTTP Hub server. To use an authenticated proxy, you must configure your Hub server as HTTPS. Pass-through (non-authenticated) proxies work with an HTTP Hub server. Pass-through (non-authenticated), basic authenticated, and digest authenticated proxies work if the Hub server is HTTPS.

To configure the Hub JIRA plugin to use a proxy:

1. At the far right of the toolbar at the top of the page, click the gear icon to display the **JIRA ADMINISTRATION** drop-down menu.
2. In the **JIRA ADMINISTRATION** drop-down menu, select **Add-ons**. The **Administration** > **Add-ons** page displays. Note that you may be required to re-enter your administrator password.
3. In the left navigation panel of the **Administration** > **Add-ons** page, select **Hub JIRA**.
4. In the **Add-ons** page for Hub JIRA, select the **Hub Server** tab. The **Proxy Config** panel is located directly below the **Server Config** panel on the **Hub Server** page.
5. In the **Proxy Config** panel, type the information for your proxy:
   - Server
   - Port
   - User name
   - Password
   - No Proxy Host
6. Click **Test Connection**. If the connection is successful, click **Save**.

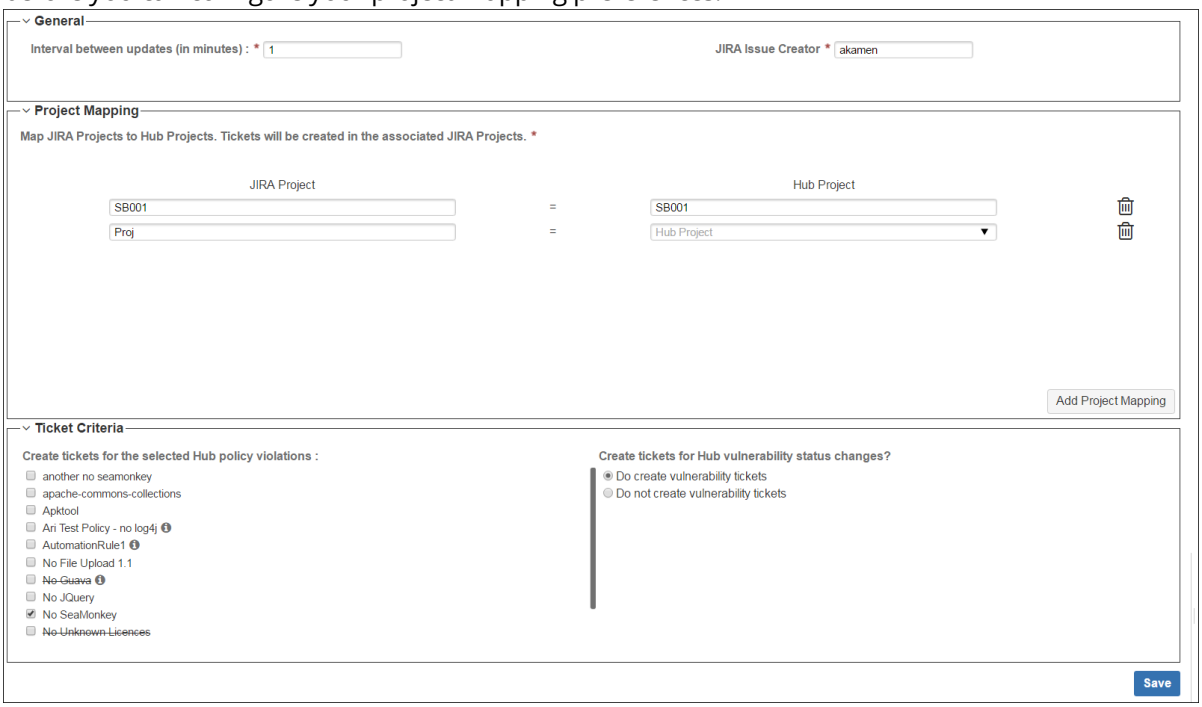
3.5 Issue Creation Configuration for the Hub JIRA Plugin

Issue creation configuration for the Hub JIRA plugin is described as follows. Issue creation configuration

is required.

To perform issue configuration for the Hub JIRA plugin:

1. Navigate to the **Add-ons** > **Hub JIRA** configuration page, and select the **Issue Creation** tab.
2. In the **Interval between updates (in minutes)** section, type the amount of time in minutes to wait before the plugin wakes up to check for Hub notifications and performs the necessary actions.
3. In the **JIRA Issue Creator** field, select the user name from the drop down, or type the user name of the JIRA user to be the creator of the tickets that are created by the plugin. There can only be one issue creator user. The drop down list consists of all users in the groups configured on the **Configuration Access** tab. If the issue creator is not assigned to one of those groups, you must type in the user name.
4. In the **Project Mapping** section, select the JIRA project issues on the left to which you want Black Duck Hub information added. For each additional project mapping, click **Add Project Mapping** to display an additional project mapping field. Note that you must create one or more JIRA projects before you can configure your project mapping preferences.



5. To control ticket volume creation, in the **Ticket Criteria** section, complete the following:
    a. Under **Create tickets for the selected policy violations**: Select one or more policy violations for which new issues are created. Note that disabled policies display in strikethrough font.
    b. Under **Create tickets for vulnerabilities status chanages?**, choose **Do** or **Do not**, depending on whether or not you want the plugin to generate vulnerability issues in response to Hub notifications:

1. **Do create vulnerability tickets:** Turns on the creation of vulnerability tickets.
2. **Do not create vulnerability tickets**: Turns off the creation of vulnerability tickets.

6. Click **Save**.

Note: When clicking **Save**, an error message displays if the selected **JIRA Issue Creator** user does not have rights to JIRA : *The user specified as the issue creator is not a valid JIRA user or has not been granted access to the Hub JIRA plugin.*

Important: Ensure that you click **Save** after completing each tab. Otherwise, your configuration and settings are not applied, and may cause errors when generating tickets and issues.

3.6 Issue Field Configuration for the Hub JIRA Plugin

Issue field configuration for the Hub JIRA plugin is optional. Issue field setup is only required if you want Hub field data such as **Project**, **Project Version**, and others copied to other JIRA fields. Issue field configuration for the Hub JIRA plugin is described as follows.

✱ **To perform Issue field configuration for the Hub JIRA plugin:**

1. Navigate to the **Add-ons** > **Hub JIRA** configuration page, and select the **Issue Fields** tab.

1. In the **Field Mapping** section, configure **Map Hub Fields to JIRA Fields**.
   a. Using the drop-down selector on the left, select the **Hub Field** value to map to the selected JIRA field.
   b. Using the drop-down selector on the right, select the **JIRA Field** to which the Hub field you selected in *step a* maps.
   c. For additional field mappings, click **Add Field Copy Mapping**.
2. Click **Save**.

Important: Ensure that you click **Save** after completing each tab. Otherwise, your configuration and settings are not applied, and may cause errors when generating tickets and issues.

3.7 Ticket Creation Errors

The **Ticket Creation Errors** section displays at the bottom of the **Configuration** page in the **Add-ons**

page. The monitoring of successful or failed ticket creation continually and automatically runs as a service in the background. This section displays ticket creation error information the same as in the log files. As a best practice, Black Duck recommends that you periodically check the **Ticket Creation Errors** section. Click the trashcan icon to remove individual error messages.

Resetting

Clicking **Reset** resets the date/time stamp for the last successful run to the current date/time. Consult with your Black Duck support personnel as to when and how the reset option should be used.

3.8 Configuring Logging for the Hub JIRA plugin

Because recent error messages are displayed on the configuration screen, logging may not be required. However, you can enable logging so that you can refer to logs from the Hub JIRA plugin. These logs contain logged actions, and monitor plugin activity. To enable logging, you can select one of the following options.

- *Temporary* logging; persists until the server is restarted.
- *Permanent* logging; persists through server restarts. Permanent logs are contained in the `catalina.out` file on the server.

Procedures to configure both scenarios are as follows.

To configure temporary logging:

1. Log into JIRA as an administrator, and navigate to the **Administration** section.
2. Select the **System** tab.
3. Under the **Troubleshooting and Support** section, click **Logging and profiling**.
4. Under **Default Loggers**, click the **Configure** link.
5. For the package name, type `com.blackducksoftware` to get all Black Duck logs.
6. Select the desired logging level. Most log messages are logged at the *debug* level; for fewer logging messages, set the logging level to *info*.

To configure permanent logging:

1. You must edit the file `log4j.properties`, located in the `WEB-INF/classes` directory.
2. Edit the `log4j.properties` file by adding the following lines:

```
log4j.logger.com.blackducksoftware = DEBUG, console,
filelog log4j.additivity.com.blackducksoftware = false
```

Chapter 4: Using the Hub JIRA Plugin

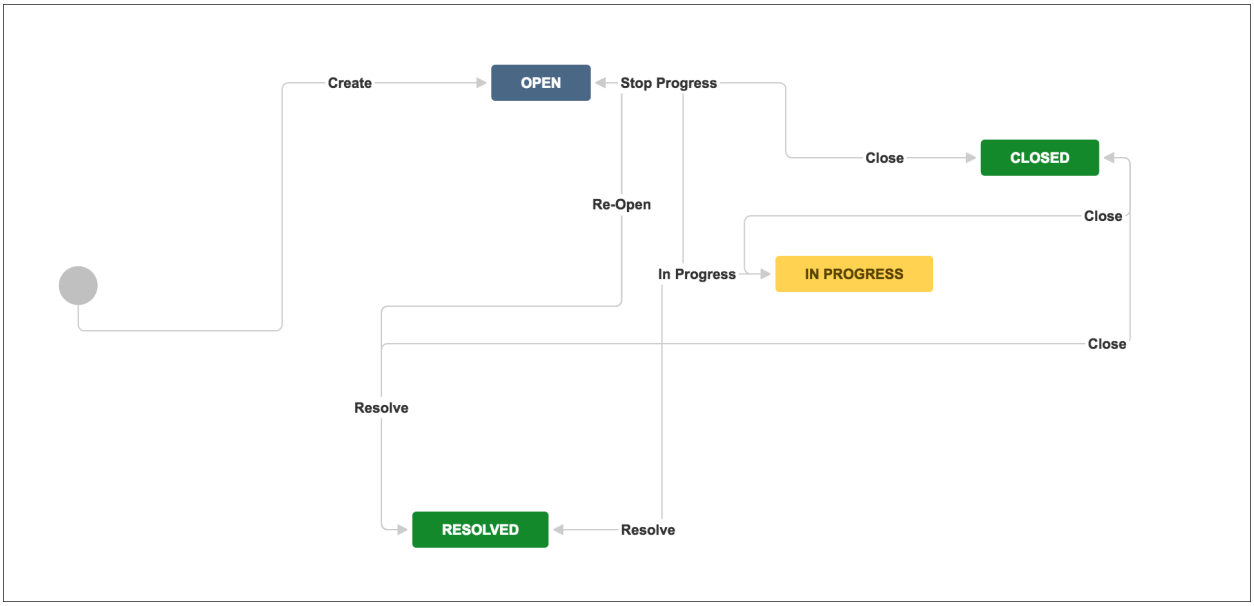
The following topics describe the features and usage of the Black Duck Hub JIRA plugin.

4.1 Automated Configuration of Issue Types, Screens, Workflow, and Projects

The first time the scheduled job runs, it creates the following Hub-specific JIRA objects:

1. Two issue types:
    a. **Hub Policy Violation**
    b. **Hub Security Vulnerability**
2. Two screens:
    a. **Hub Policy**
    b. **Hub Security**
3. Five custom fields:
    a. **Hub Project**
    b. **Hub Project Version**
    c. **Hub Component**
    d. **Hub Component Version**
    e. **Hub Policy Rule**
        - The **Hub Project**, **Hub Project Version**, **Hub Component**, and **Hub Component Version** fields are mapped to both screens: the **Hub Policy** screen and the **Hub Security** screen.
        - The **Hub Policy Rule** field is only mapped to the **Hub Policy** screen.
4. One field configuration: the **Hub Field Configuration**. This field includes all default fields, including the Hub-specific custom fields, and makes all fields optional except **Summary** and **Issue Type**.
5. One workflow: the Hub workflow is automatically imported. Refer to the following workflow image.
6. The screens are mapped to the Hub issue types.
7. The Hub issue types are added to the JIRA projects that are configured in project mapping.
8. The Hub workflow is assigned to the Hub issue types in the JIRA projects that are configured in project mapping.

The Hub workflow is illustrated as follows.



In all future runs, the scheduled job attempts to setup and validate all Hub-specific JIRA objects to maintain the environment expected by the plugin.

4.2 Ticket Handling in Hub JIRA

Refer to Feature Support by Hub Version on page 6 for details on which Hub JIRA features are supported in which Black Duck Hub version.

Default Hub JIRA ticket handling is described as follows, which includes automatic resolving and re-opening of JIRA issues based on Hub events. For information on disabling automatic resolving and re-opening of JIRA issues, refer to Replacing the Workflow, and Disabling Automatic Resolving and Re-Opening of Issues on page 18.

- Tickets are created when:
  - For policy violations:
    - A component in the Hub violates a policy rule, creating a Hub policy violation issue.
    - The policy rule is selected in the Hub JIRA configuration.
    - The Hub project is mapped to a JIRA project.
  - For vulnerabilities:
    - A component containing vulnerabilities in the Hub is added to the Bill of Materials. Or, due to a change to the KnowledgeBase, vulnerabilities change, resulting in the component having at least one vulnerability.
    - The Hub project is mapped to a JIRA project.
- Policy violation tickets are resolved if:
  - A component in the Hub containing a violation has that violation overridden, or that

component is removed from the Bill of Materials.
    - The policy rule is selected in the Hub JIRA configuration.
    - The Hub project is mapped to a JIRA project.
    - The policy violation issue type is configured to use the Black Duck Hub plugin workflow.
- Policy violation tickets are re-opened if:
    - They have a *Resolved* status. Policy violation tickets are not re-opened by the plugin if the status is *Closed*.
    - The component containing a violation in the Hub has the violation override removed, or that component is re-added to the Bill of Materials.
    - The policy violation issue type is configured to use the Black Duck Hub plugin workflow; this is the default configuration.
- Vulnerability tickets are resolved if:
    - Creation of tickets for vulnerabilities is enabled on the plugin configuration screen.
    - The corresponding component in the Hub project is removed from the Bill of Materials. Or, due to a change in the Knowledge Base, there are no more vulnerabilities on that component.
    - The Hub project is mapped to a JIRA project.
    - The vulnerability issue type is configured to use the Black Duck Hub plugin workflow; this is the default configuration.
- Vulnerability tickets are re-opened if:
    - Creation of tickets for vulnerabilities is enabled on the plugin configuration screen.
    - The corresponding component in the Hub project is added to the Bill of Materials. Or, due to a change in the KnowledgeBase, there are new, updated, or deleted vulnerabilities on that component, resulting in at least one vulnerability on that component.
    - The Hub project is mapped to a JIRA project.
    - The vulnerability issue type is configured to use the Black Duck Hub plugin workflow; this is the default configuration.

4.3 Replacing the Workflow, and Disabling Automatic Resolving and Re-Opening of Issues

This feature is available in the Hub JIRA plugin versions 3.0.1 and higher. Automatic resolving and re-opening of tickets results in the following behavior:

- When the policy is violated, a JIRA ticket is created and is in the *Open* state.
- When the policy is cleared, the JIRA ticket remains *Open*. A comment is added to notify the JIRA owner that the policy is cleared.
- If the same policy is violated again, a new comment is added to the existing ticket.
- When the same policy is cleared again, another comment is added to the existing ticket.

You can disable automatic resolving and re-opening of tickets as described in the following procedure.

To disable automatic resolving and re-opening of issues for a JIRA project:

1. In the Hub JIRA configuration screen, the JIRA project must be mapped to a Hub project, and that configuration must be saved.
2. Disable the periodic task by navigating to **Admin** > **Add-ons** > **Manage add-ons**, expanding the **Hub JIRA Plugin** panel, and clicking **Disable**.
3. Access the JIRA project setup by navigating to **Admin** > **Projects**, and clicking the project name.
4. In the **Workflows** section, click the **Workflow Scheme**. Note that the *Hub Policy Violation* and *Hub Security Vulnerability* issue types are assigned to the *BDS Hub PlugIn Workflow*.
5. Move the *Hub Policy Violation* and *Hub Security Vulnerability* issue types to another workflow listed on that screen (the target workflow) by clicking the **Assign** link in the target workflow row, and selecting the **Hub Policy Violation** and **Hub Security Vulnerability** issue types.
6. Click **Finish**.
7. Click **Publish**.
8. On the **Publish Workflows** screen, map the *BDS Hub PlugIn Workflow* states to target workflow states for each issue type.
9. Click **Associate**.
10. Enable the plugin.

4.4 Viewing and Resetting Error Messages

Errors that occur during runs of periodic tasks; for example, connection to the Hub, internal failure, cannot creat tickets, and so forth, appear at the bottom of the Hub JIRA configuration, regardless of which configuration tab is selected. The date format in each message is MM/DD/YYYY. Click the trash can icon next to each message to clear that message.

The **Reset** button that appears below error messages should only be used if the same error recurs each time the periodic task runs due to a notification that the plugin cannot handle. Resetting skips over all notifications generated between the last successful run and the time the reset button is pressed; therefore, it should only be used if the plugin is stuck on a notification it cannot handle.

4.5 Periodic Task Timeout

This feature is available in the Hub JIRA plugin versions 3.0.1 and higher.

Creation and updating of JIRA issues is performed by a periodic task that is scheduled to run at an interval specified in the Hub JIRA plugin configuration. This value is found in the **Interval between updates (in minutes)** field; the default value is *1*. If the task times out, it re-attempts to process the notifications it did not previously complete at the next interval.

If the task is frequently timing out, you may be able to reduce the number of task timeouts by increasing the value for **Interval between updates (in minutes)** on the Hub JIRA configuration screen.

4.6 Considerations When Using the Hub JIRA Plugin

When using the Black Duck Hub JIRA plugin, understanding the following points can be beneficial.

- Removing components from the Bill of Materials (BOM) that caused tickets to be created resolve the ticket, as long as the component version removed is the component version that caused the ticket on the same project and project version.
- Issues are specific to *Project/Project version/Component/Component version* policy violation rules.
- If a vulnerability issue arises: these are specific to*Project/Project version/Component/Component version*. Adding a component that is in error back to the BOM reopens a *Resolved* ticket. Adding it back to a *Closed* ticket does not reopen that ticket. Once a ticket is closed, it cannot be reopened.

Chapter 5: Troubleshooting your Black Duck Hub Plugin

Refer to the following sections should issues arise during use of your Hub plugin instance.

Tip: After major releases of Black Duck Hub, check for updated versions of your Black Duck plugins and their installation prerequisites. Changes to the APIs, schema, and SDK versions may require updated versions of the integration plugins.

5.1 Troubleshooting the Hub JIRA Plugin

For general issues with the Hub JIRA plugin, you can check the **Configuration** page for errors. If there are no error messages and you feel that you may be experiencing problems, or if the error messages are not helpful, contact Black Duck for support.

Other Troubleshooting Scenarios

If an error message is generated that states *During development and testing the following errors were encountered*, use the following solutions:

- If you try to use Java 6 instead of Java 7, instead of getting an *Unsupported major:minor version error* message, the plugin sometimes throws a false *java.lang.OutOfMemoryError: Java heap space* message instead.
- If you get a message that reads *Service Unavailable*, either the Hub server can't be reached, or the request to the server is invalid. Contact your Hub server administrator.
- If you get a *Precondition failed* error message, then the request to the server is invalid. Verify that your global configuration is correct, and verify that the job configuration is correct. If you are still getting this message after you have checked your configuration, contact your Black Duck technical account manager.
- If you get a *Not Found (404) - Not Found* error message, then the request to the server is invalid. Contact your Black Duck technical account manager.
- If you try to use the Hub JIRA integration, and you configure a job with a project and version, and that project already exists but the current Hub user is not assigned to it, then the following errors display:
    - In the job configuration **Project Name** field, a notification displays *This project does not exist on the Hub Server.* Clicking **Create project/version** displays a message reading *This version may already exist.com.blackducksoftware.integration.hub.exception.BDRestException: There was a problem creating this Hub project. Error Code: 412*.
    - If you run the build, the following displays:

        Status : 412

        Response : {"errorMessage":"project name already exists","arguments":

{"fieldName":"name"},"errors":[{"errorMessage":"project name already exists","arguments":{"fieldName":"name"},"errorCode":"{central.constraint_violation.project_name_duplicate_not_allowed}"}],"errorCode":"{central.constraint_violation.project_name_duplicate_not_allowed}"}

Problem creating the project.

Assigning the current user to the existing project with this name resolves the issue.

5.2 Ticket Creation Errors

If you are a Hub JIRA configuration-enabled user, you can perform the following troubleshooting steps for ticket creation.

To address ticket creation errors:

1. Navigate to the Hub JIRA configuration screen located at **Admin** > **Add-ons** > **Hub JIRA**.
2. Scroll to the bottom of the page.
3. If the section **Ticket Creation Errors** exists, use the error message timestamp to look for error messages that may be related to the problem you are experiencing. Use the **+** icon to view more details about each error message when applicable.

5.3 Investigating Issues Using Debug-level Logging

You can investigate issues for troubleshooting using debug-level logging. For information on enabling debug-level logging, refer to Enabling Debug-level Logging. After debug-level logging is enabled, perform the steps required in the Hub and/or JIRA to reproduce the problem, which creates debug messages in the log file. For example, if the problem occurs during JIRA issue creation, then you should perform actions in the Hub that cause a new JIRA issue to be created.

Note: The log file can only be accessed by a JIRA administrator.

To investigate issues using debug-level logging:

1. In JIRA, navigate to **Admin** > **System**.
2. Click **Atlassian support tools**.
3. Click the **Support Zip** tab.
4. Deselect all checkboxes except **Tomcat Logs** and **Limit File Sizes**.
5. Click **Create**.
6. The file path to the support zip file, located on the JIRA server, displays on the screen in the status message *Your support zip file is ready and can be found at: ...*. Retrieve the support zip file from the JIRA server.

The file `tomcat-logs/catalina.out`, which is contained in the support zip file, contains log messages from the *com.blackducksoftware.integration* classes that are helpful in troubleshooting these issues.

5.4 Steps That Can Be Performed by a Configuration-Enabled User

The following steps can be performed by a configuration-enabled user.

1. Navigate to the Hub JIRA configuration **Admin** > **Add ons** > **Hub Jira**.
2. Scroll to the bottom of the page.
3. If the section **Ticket Creation Errors** displays, use the error message time stamp to look for error messages that may be related to the problem you are experiencing. Use the **+** icon to see more details about each error message when available.

5.5 Steps Requiring a JIRA Administrator and a User

A JIRA administrator and a user are required for the following steps.

First, a JIRA administrator must enable debug-level logging as described in the following procedure.

1. In JIRA, navigate to **Admin** > **System**.
2. Click **Logging and profiling**.
3. If `com.blackducksoftware.integration` does not appear in the **Package name** list, click **Configure logging level for another package** to add it.
4. Set the logging level for `com.blackducksoftware.integration` to *DEBUG*.

After debug-level logging is enabled, a JIRA user must perform the steps required in the Hub and/or JIRA to reproduce the problem, which causes debug messages to be written to the log. For example, if the problem occurs during JIRA issue creation, the user must perform actions in the HUB that cause a new JIRA issue to be created.

Then, a JIRA administrator must retrieve the log file, as described in the following procedure.

1. In JIRA, navigate to **Admin** > **System**.
2. Click **Atlassian support tools**.
3. Click the **Support Zip** tab.
4. Deselect all check boxes except **Tomcat Logs** and **Limit File Sizes**.
5. Click **Create**.
6. In older JIRA versions, the file path to the support zip file on the JIRA server displays on the screen with the message *Your support zip file is ready and can be found at: ...*. Retrieve the support zip from the JIRA server.
7. In newer JIRA versions, you can download the support zip file through the browser.

The file `tomcat-logs/catalina.out`, which is located within the support zip file, contains log messages from `com.blackducksoftware.integration` classes to help troubleshoot the issue.

Chapter 6: Hub JIRA Plugin Release Notes

6.1 New and Changed Features

Changes in Release 3.2.0

- The **Configuration** page tabs are re-organized. The new tabs are:
    - Configuration Access
    - Hub Server
    - Issue Creation
    - Issue Fields
- You can now configure a JIRA issue creator user.
- When upgrading from an earlier version of the Hub JIRA plugin to version 3.2.0 and higher, the issue creator remains unchanged; it is the last user to save the configuration, until the administrator sets the creator to a different user in the configuration.
- Added support for the Firefox browser.
- Hub server configuration options are now located on the **Administration** > **Add-ons** > **Hub Server** page. The **Hub Admin** plugin is no longer required, and should be uninstalled where applicable.
- New options for fields for in the **Hub Field** selector in the **Field Mapping** section are:
    - Project version nicknames
    - Component license names
    - Component usage
- License names are now hyperlinked to the corresponding license text in the Hub.

Changes in Release 3.1.1

- The Hub JIRA plugin now checks for component violation using the `approvalStatus` on the `bomComponentVersionPolicyStatus`.
- Addressed an issue wherein a policy violation notification could be incorrectly created for an ignored component.
- Addressed an issue wherein the documented methods for changing the logging level may be ineffective.
- Addressed an issue wherein a Hub JIRA task may fail if the user receives notifications about projects to which they did not have access.
- Addressed an issue wherein the Hub JIRA task may fail with the message *Could not find the link 'policy-rule', these are the available links.*
- Default logging level is now set to INFO for com.blackducksoftware.integration, and WARN for

everything else called by the plugin. This avoids highly-verbose INFO-level SQL logging.

- Auto-generated comments now display as *Black Duck Hub JIRA-plugin auto-generated comment*.

Changes in Release 3.1.0

- You can now disable Hub JIRA tickets for improved security.
- There is now an option for *Do not create vulnerabilities*.
- When a custom field requires updating, it is now performed without deleting the associated data for that field.
- The **Configuration** page is now split into three tabs: **Admin**, **Basic**, and **Advanced**.
- The version of the plugin now displays on the **Configuration** page.
- Added separate avatars for policy violation and vulnerability issues.
- New issues now contain a link to the Hub project vulnerable components page.
- Improved the link display text for new vulnerability issues on the link to the Hub vulnerable components page.
- Added the ability to create a mapping of Hub-source fields to JIRA issue fields. Therefore, the data from the Hub is written into both fields.
- The **Configuration** page features improvements for the display of error messages.

Changes in Release 3.0.2

- Addressed an issue that could prevent the Add-on **Installing** page from automatically closing after the plugin is installed. This was caused by an attempt to open a log file that could fail, depending on directory permissions.
- Addressed an issue that could prevent policy violation issues from being created. This was caused by a `NullPointerException` in `HubFieldpageSchemeSetup.addHubTabTopage()`.

Changes in Release 3.0.1

- You can now configure the automatic state update to be on or off.
- For vulnerability issues, URLs now display for vulnerable components and individual vulnerabilities.
- Added the ability to safely disable BDS Hub Plugin Workflow to turn off automation state transitions.
- Addressed an issue with rare configuration save errors when policy rule descriptions introduced certain non-alphanumeric characters.
- Addressed an issue wherein tickets may be created, even though no policy rules were enabled.
- Addressed an issue wherein the time stamp was truncated when it occurred in events.
- For components with vulnerabilities, the description now has a link to the Hub.

Changes in Release 3.0.0

- Requires JIRA version 7.1.x or higher.

Changes in Release 2.0.1

- Requires JIRA versions 6.4.0 to 7.0.11.
- Added **Configuration** page **Reset** button.

- Added configurable user groups.
- Added auto-resolution of tickets when the vulnerability count reaches 0.

Changes in Release 2.0.0

- Added support for JIRA versions 6.4.0 to 7.0.11.
- Added Black Duck-specific issue types: Custom Fields / Page / Configurations / Workflows.
- In handling vulnerability notifications, all vulnerabilities generated for mapped projects now generate tickets.
- When handling deletions, any removal of policy violations, overrides, or vulnerabilities now resolves tickets. Note that this functionality requires Hub version 3.3.1 or higher.
- Hub JIRA configuration now shows errors occurring during job runs; for example, Hub connection, ticket creation, permission errors, and others.

Changes in Release 1.0.0

- Hub JIRA configuration is now located in the **Add-Ons** section.
- Configurable interval between notification checks and ticket creation.
- Configurable mapping from JIRA projects to Hub projects. Supports a many-to-many mapping.
- Configurable policy violations for which to create tickets.
- Creates tickets when a component in a project/version violates a defined policy rule.
- Resolves tickets for components violating a policy rule when that violation is overridden.
- Re-opens tickets when a new notification is sent out for the same Hub project/version, component, and violation.

6.2 Hub JIRA Plugin Known Issues

The following are known issues for the Hub JIRA plugin.

Known Issues for Hub JIRA Release 3.1.0

- The link to the Hub project/version was added in Hub version 3.4.2, but unless you are already signed in to the Hub in the same browser, a JSON error of `{"errorMessage":"Invalid username or password","arguments":{},"errors":null,"errorCode":"{core.rest.unauthenticated}` may occur because you are not authenticated in that browser.
- When non-sysadmin (system administrator) Hub users are logged in, the plugin displays plain text instead of live hyperlinks to the Hub in the **Description** field for the project version and vulnerabilities.

Known Issues for Hub JIRA Releases 2.0.1 - 3.0.0

- When upgrading from a previous version of the Hub JIRA plugin, the plugin does not recognize JIRA issues created by an earlier version of the plugin. As a result, it may create JIRA issues that duplicate those created by an earlier version of the plugin, and does not close issues created by an earlier version of the plugin.
- When upgrading from a previous version of the Hub JIRA plugin, you may need to clear your browser cache to correctly render the Hub JIRA configuration page.

Known Issues for Hub JIRA Releases 1.0.0 - 3.0.0

- If a Hub policy rule description contains an embedded linefeed character, attempting to save the Hub JIRA configuration fails. Trailing newlines do not cause a problem.

Known Issues for Hub JIRA Release 2.0.0 - 2.0.1

- For JIRA version 7.0: In the Hub-specific issue types, the Black Duck Hub web panel does not display. Therefore, the Hub-specific custom fields display just like any other custom fields in the **Details** section.

Known Issues for Hub JIRA Release 2.0.0

- If more than one process tries to perform the CLI install at the same time to the same directory (on the same machine), then the processes collide with each other and start deleting files while another creates them.
    - Workaround: When a process finds that the currently installed CLI must be updated (downloaded for the first time or to change the current files), then it should create a `.locked` file (or a similar and appropriate file name), then perform the update, and remove the `.locked` file when the update is complete. If another process starts to perform the install, it should first check for the `.locked` file, if it exists, then it should wait until the `.locked` file is deleted before performing its check.

Chapter 7: 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 or integration product.
- Black Duck product name and version number.
- Third-party integrated product and version; for example:
    - Visual Studio
    - MSBuild
    - TFS
    - Artifactory
    - Eclipse
    - Jenkins
    - Maven, and others.
- 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, MSBuild project files, or Project Object Model (POM) XML files, as applicable.

7.1 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.

7.2 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.