BLACKDUCK

Installing, Configuring, and Using the Hub Artifactory Plugin

Version 2.0.0

This edition of the *Installing, Configuring, and Using the Hub Artifactory Plugin* refers to version 2.0.0 of the Black Duck Hub Plugin for Artifactory.

This document created or updated on Thursday, February 09, 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 Artifactory Plugin Overview

The Black Duck Hub Artifactory plugin is a new tool designed to help you manage the logistics of your Artifactory repositories and contents. The Black Duck Hub Artifactory plugin can also help you uncover security and compliance risks associated with the open source artifacts stored in Artifactory repositories.

As of Hub Artifactory 2.0.0 and higher, there are two operational mode options: Scanner and Inspector. Having two modes which you can use separately or together brings further flexibility, enabling you to get even more from the Hub Artifactory plugin. Your environment and workflow preferences determines your choice and utilization of the two modes.

As a Hub and Artifactory user, the Hub Artifactory plugin enables you to:

- Run an artifact scan in an Artifactory job:
    - Scan multiple artifacts within your Artifactory repositories.
    - Create projects and releases in Black Duck Hub through the Artifactory job.
    - Schedule and automate the scanning of your Artifactory Repositories.

After a scan is complete, the results are available on the Hub server. Additionally, scanned artifacts include new properties in the Artifactory user interface indicating the most recent scan time, scan status, and a URL to view the scan results in Black Duck Hub.

Chapter 2: Installation Overview

The following sections provide a step-by-step guide to downloading and installing your Black Duck Hub Artifactory plugin.

Note: You can find Artifactory's documentation at: https://www.jfrog.com/confluence/display/RTF/Welcome+to+Artifactory

2.1 Installation Prerequisites

Before you install the Hub Artifactory plugin, ensure that:

- Hub Artifactory Pro version 4.14.0 or higher.
- Your Artifactory instance is up-to-date and fully patched.
- You know the host name and port for the Hub server.
- You have a user account with administrator privileges on the Hub system that you can use for the integration.
- You have connectivity to the internet. The machine that hosts your Artifactory server must be able to connect to the Hub server.

2.2 Downloading the Hub Artifactory Plugin

Download the Black Duck Hub Artifactory plugin from the specified GitHub location. The plugin is contained in a `.zip` file.

To download the Hub Artifactory plugin:

1. Navigate to https://github.com/blackducksoftware/hub-artifactory/releases to download the latest version of the Black Duck Hub Artifactory plugin.
2. Download and save the `.zip` file to a temporary location.

2.3 Hub Artifactory Scanner and Inspector Modes

As of Hub Artifactory 2.0.0 and higher, there are two operational mode options: **Scanner** and **Inspector**.

You may want to use either or both, depending on your organizational structure.

- If you use Artifactory to deploy your final build artifact, use Scanner.
- If you use Artifactory to cache your dependencies, use Inspector on those cached repositories.
- If you do both, you can use both Scanner and Inspector to provide the necessary context to enforce policy management.

Usage examples

Inspector: If your organization uses a given package manager, and has archival downloads from this package manager, Inspector creates a Bill of Materials (BOM) of all the dependencies downloaded by your developers. However, using only the Inspector mode, you do not have full context of what has been shipped to your customers.

Scanner: Scanner creates a Bill of Materials (BOM) that represents your shipped code. So, by using Scanner and Inspector together, you can have a full accounting of everything in your repositories and your shipped product.

For more information on each mode, refer to Hub Artifactory Scanner Overview on page 6 and Hub Artifactory Inspector Overview on page 6.

2.4 Hub Artifactory Scanner Overview

The Hub Artifactory Scanner deploys by using the groovy files on the Artifactory instance. On an archive you specify, it downloads the CLI, and performs the scan on that archive. It also detects other binary information, such as components and dependencies.

2.5 Hub Artifactory Inspector Overview

The Hub Artifactory Inspector is part of the executable JAR file. The major difference from the Scanner is that it has no scanning component, and therefore does not need to be run on Artifactory. Inspector is a stand-alone that integrates with the Hub API and the Artifactory API. Unlike Scanner, Inspector looks at specific repositories and analyzes each package inside that repository.

2.5.1 Package Managers Supported by the Hub Artifactory Plugin

As of Hub Artifactory version 2.0, the following package managers are supported.

- Ruby Gems
- NPM (Node Package Manager)
- NuGet
- PyPI
- Maven/Gradle

Future Hub Artifactory plugin releases will contain support for additional package managers.

2.6 Installing the Hub Artifactory Plugin

After you have downloaded the Hub Artifactory zip file as described in Downloading the Hub Artifactory Plugin on page 5, use the following process to install your Hub Artifactory plugin. As of Hub Artifactory 2.0.0 and higher, there are two installation options:

- **Scanner**: This contains the standard Hub Artifactory functionality and features. **Scanner** scans the end artifact only.
- **Inspector**: Inspects the repository on a package-by-package basis.

The installation option you choose begins with the unzipping options. Both options are discussed in the

following sections.

2.6.1 Installing the Hub Artifactory Scanner Option

To install the Hub Artifactory plugin with the Scanner option:

1. SSH to the server hosting your Artifactory instance.
2. Navigate to the location where you saved the Hub Artifactory zip file.
3. Unzip and extract the zip file to a directory on the machine hosting your Artifactory installation. The structure of the `.zip` file is:

```
/
-blackDuckScanForHub.groovy
-lib
    /
    -jarFile.jar
    -jarFile2.jar
```

4. Place the `blackDuckScanForHub.groovy` file in: `.../path/to/your/artifactory/etc/plugins/`.
5. Create a `lib` folder in: `.../path/to/your/artifactory/etc/plugins/`, so that the path to the `lib` folder is: `.../path/to/your/artifactory/etc/plugins/lib`.
6. There are multiple `.jar` files contained within the `.zip` file. Place all `.jar` files from the `.zip` file in the `lib` folder you created in the previous step.
7. Schedule the scans of your Artifactory repositories by configuring the CRON job in the `blackDuckScanForHub.groovy` file. This is discussed in Configuring the Scanning Schedule on page 9.
8. Restart Artifactory using the commands:
    - `service artifactory stop`
    - `service artifactory start`

2.6.2 Installing the Hub Artifactory Inspector Option

Unlike the Hub Artifactory Scanner, the Hub Artifactory Inspector option doesn't require an installation process. Instead, to run in Inspector mode, use the following process.

To run the Hub Artifactory plugin in Inspector mode:

1. Unzip the Hub Artifactory zip file you previously downloaded.
2. Locate and run the `hub-artifactory-2.0.0.jar` file.

Chapter 3: Configuring the Hub Artifactory Plugin

The following sections explain and detail the options for configuring both the **Scanner** and **Inspector** options for your Hub Artifactory plugin.

3.1 Configuring the Hub Artifactory Plugin *Scanner* Option

If you have selected the Hub Artifactory **Scanner** option, you can configure it using the following process.

Configure the `blackDuckScanForHub.groovy` file to customize your Hub Artifactory plugin. This enables you to control the file types for scanning, and when the Artifactory properties are updated. This enables you specify which Artifactory repositories and artifact types are scanned. You can also configure the Hub server where scan results are sent. Additionally, you can also configure the scheduling of scans.

3.2 Configuring the Hub Artifactory Plugin *Inspector* Option

If you have selected the Hub Artifactory **Inspector** option, you can configure it using the following process. Note that default values for common fields are provided. You can choose to accept these, or provide your own, based on your environment and workflow preferences.

✱ **To configure the Hub Artifactory Inspector option:**

1. Unzip the installation file you downloaded.
2. Using your command line utility, select and run the Inspector JAR file. For example, run: `java - jar <name of jar file>`. In Hub Artifactory 2.0.0, the name of the file to specify is `hub-artifactory-2.0.0.jar`.

3.3 The Config Scan Mode

The **Config Scan** utility updates and modifies the Groovy file located in the scanner directory with the values you input into the automated CLI tool. This gives you the choice to set up your environment directly from a command line, or to manually edit the Groovy file. For more information on manually editing the Groovy file, refer to Configuring Preferences Using the Hub Artifactory Groovy File on page 10.

✱ **To configure your preferences using the Config Scan mode:**

1. Run the automated configuration tool using the command:
   `java -jar hub-artifactory-2.0.0.jar --mode=configure-scanner`

   After this command, you are greeted with a command line prompt similar to the following

example.

```
:: Spring Boot :: (v1.4.3.RELEASE)

2017-02-09 13:24:18.422 INFO 16856 --- [ main]
c.b.i.hub.artifactory.Application : You are running in an
interactive mode - if configuration is needed, you should be
prompted to provide it.

Updating Config - just hit enter to make no change to a value:
```

2. Complete the following prompts:
   a. `Enter Hub Server Url`: Type the URL for your Hub server.
   b. `Enter Hub Server Username`: Type your Hub user name.
   c. `Enter Hub Server Password`: Type your Hub password.
   d. `Enter Hub Server Timeout`: Type the number of seconds for system timeout; the default is 120.
   e. `Enter Local Working Directory`: Type the name of your local working directory.
3. After completing these steps, the Config Scan utility automatically verifies your configuration.

Note: You can change your configuration at any time by re-running the Config Scan utility. Your current values display at each prompt; you can change each one, or press Enter to accept the current value.

3.4 Configuring the Scanning Schedule

You can configure the CRON job to determine when to automatically run the Hub Artifactory plugin. This enables you to set up Hub Artifactory scans to run automatically at your predetermined intervals.

To configure the Hub Artifactory plugin CRON settings:

1. Log in to Hub Artifactory as administrator.
2. Go to `.../path/to/your/artifactory/etc/plugins/`, and locate the `artifactory-plugin.groovy` file.
3. Open the `blackDuckScanForHub.groovy` file in a text editor such as Notepad.
4. Locate a row in the `artifactory-plugin.groovy` file similar to:
   `scanForHub(cron: "0 0/1 * 1/1 * ?")`
   and schedule the job for a specific running time by editing the CRON parameters as follows:

CRON parameters

From left to right:

- 1 - Seconds
- 2 - Minutes

- 3 - Hours
- 4 - Day-of-Month
- 5 - Month
- 6 - Day of week
- 7 - Year (optional field)

Examples:

`"0 42 10 * * ?"` - Builds a trigger that fires daily at 10:42 am.

`"0 0/2 8-17 * * ?"` - Builds a trigger that fires every other minute, between 8am and 5pm, every day.

Tip: For more information on building your customized CRON settings, and to use an automated CRON building tool to make building your custom CRON settings easier, visit http://www.cronmaker.com/. You can build your CRON expressions using the automated builder, and then copy them into the `artifactory-plugin.groovy` file.

3.5 How the Hub Artifactory Properties are Updated

The Hub Artifactory plugin updates your properties metadata as follows.

- *Main function*: The Hub Artifactory plugin searches your Artifactory repositories, and scans your artifacts.
- *After scanning*: The Hub Artifactory plugin continuously updates your Artifactory metadata through and by using the properties. Afterward, all properties are updated.

Note: Black Duck scan time prevents additional scans of the same artifact until the last modified is updated. In other words: for each scan, the Hub Artifactory plugin only scans new or modified artifacts.

3.6 Configuring Preferences Using the Hub Artifactory Groovy File

You can customize your Hub Artifactory plugin by editing the `blackDuckScanForHub.groovy` file. The `groovy` file provides wide flexibility in customization based on your environment, and enables you to fine-tune your scans based on workflow, preferences, and changing day-to-day requirements.

You can also setup or edit your system preferences using the **Config Scan** command line utility if you prefer not to manually edit the Groovy file. For more information, refer to The Config Scan Mode on page 8.

Caution: Changing values on fields not listed below can cause severe issues with your Hub Artifactory instance.

To configure the Hub Artifactory Groovy file:

1. Log in to Hub Artifactory as administrator.
2. Go to `.../path/to/your/artifactory/etc/plugins/`, and locate the `blackDuckScanForHub.groovy` file.
3. Open the `blackDuckScanForHub.groovy` file in your preferred text editor; for example, Notepad in Windows.
4. Edit the following properties. Note that all proxy fields are optional.

- `@Field final String HUB_URL=" "`: Provide the path to your Hub server.
- `@Field final int HUB_TIMEOUT=120`: System timeout in seconds; 120 is the default.
- `@Field final String HUB_USERNAME=" "`: Type your Hub user name.
- `@Field final String HUB_PASSWORD=" "`: Type your Hub password.
- `@Field final String HUB_PROXY_HOST=" "`: Type your Hub proxy host URL.
- `@Field final int HUB_PROXY_PORT= 0`: Type your Hub proxy port number.
- `@Field final String HUB_PROXY_IGNORED_PROXY_HOSTS=" "`: Type your Hub ignored proxy URLs.
- `@Field final String HUB_PROXY_USERNAME=" "`: Type your Hub proxy user name.
- `@Field final String HUB_PROXY_PASSWORD=" "`: Type your Hub proxy password.
- `@Field final int HUB_SCAN_MEMORY=4096`: Type the amount of physical memory (in MB) to be allocated for Hub scanning. The default is 4096.
- `@Field final boolean HUB_SCAN_DRY_RUN=false`: A dry run scan enables you to create scan files, but the scan files are not submitted to the Hub. Values are `false` (default) or `true`. Note that scan results are not submitted to the Hub unless the value is `true`.
- `@Field final List<String> ARTIFACTORY_REPOS_TO_SEARCH=[`

  ```
  "ext-release-local",
  "libs-release"
  ]
  ```

  Type the Artifactory repositories to be scanned.

- `@Field final List<String> ARTIFACT_NAME_PATTERNS_TO_SCAN=[`

  ```
  "*.war",
  "*.zip",
  "*.tar.gz",
  "*.hpi"
  ]
  ```

Determines the files to be scanned. Specify the Artifactory file name patterns to include in your scans. You can specify only the archive file types (as shown in this example), or you can use wildcard searches

for more specificity. For example, you could search on file names to locate any files with *MyProject* in the name by specifying `MyProject.*`.

- `@Field final boolean logVerboseCronLog = false`: Turns the logging for CRON jobs on (`true`), or off (`false`). The default setting is false, which means that no date/time stamps for your automated scanning jobs are captured (logged). Turning on CRON job logging captures the start times of your automated Hub Artifactory scans. This can be helpful for verifying that CRON jobs are running as expected. Note that for `logVerbose`, only the starting time is logged.

5. Save and close the `blackDuckScanForHub.groovy` file.
6. Restart Artifactory.

3.7 What Happens After Scanning

After you have completed your Hub Artifactory Scanner scan, the following post-scanning results are available on the Black Duck Hub. This is in the form of a project representing your archive.

1. The project/version is created for each artifact.
2. You can drill down into the project/version to see the open source inventory, along with security and compliance risks.
3. If policy violations occur, they are highlighted.

3.8 What Happens After Inspecting

After completing your Hub Artifactory Inspector inspection, the following post-inspection results are available on the Black Duck Hub. This is in the form of a project representing your Artifactory archive.

To view the inspection results:

1. Navigate to the Black Duck Hub.
2. Locate the version name link in the **Version** column for the desired inspection. You only have one dated version per day; in other words, if you run more than one inspection on one day, it updates that version date name's contents and shows a new time stamp in the **Last Scanned** column, but does not change the version name. This is the default naming behavior; in the automated CLI utility, the name you specified overrides the date version naming scheme.
3. The project/version is created for each artifact. Click the version name link to show the new Bill of Materials for the inspected repository. This also displays the total number of matched components in the inspected repository. This inspection produces BDIO that is uploaded to the Hub server.
4. You can drill down into the project/version to see the open source inventory, along with security and compliance risks.
5. If policy violations occur, they are highlighted.

Chapter 4: 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.

4.1 Testing Your Configuration

After you have installed and configured your Hub Eclipse instance, you can test your connection to the Hub server using a REST API call. This test determines:

- If a successful connection can be established with the Hub server in your configuration.
- If the repositories named in your properties configuration can be located.
- The number of artifacts in each repository.
- The most recent entries in the CRON log file, if you have CRON logging enabled.

To test your configuration:

1. Open a command line window.
2. At the command prompt, type the command:
   `curl -X GET -u admin:password"http://Eclipse_SERVER/Eclipse/api/plugins/execute/testConfig"`
   Where `admin:password` is your user name and password, and the URL points to your environment.
3. Check the CLI output as follows; this is an example of the resulting output for a successful test.

```
canConnectToHub: OK
artifactsFound: 4
loggedCronRuns:
addPolicyStatus 2016-12-15T01:52:00.005
scanForHub 2016-12-15T01:53:00.003
addPolicyStatus 2016-12-15T01:53:00.004
addProjectVersionUrl 2016-12-15T01:53:00.004
scanForHub 2016-12-15T01:54:00.004
addPolicyStatus 2016-12-15T01:54:00.004
addProjectVersionUrl 2016-12-15T01:54:00.005
```

```
scanForHub 2016-12-15T01:55:00.002
addProjectVersionUrl 2016-12-15T01:55:00.003
addPolicyStatus 2016-12-15T01:55:00.006
```

- For `canConnectToHub`, a result of `OK` indicates a successful test. Otherwise, there is an issue with your Hub configuration.
- For `artifactsFound`, a value greater than `0` indicates a successful test. Otherwise, there is an issue with the name patterns and repositories you have entered.
- For `loggedCronRuns`, the test results may be blank. A blank test results indicates that either there have been no successfully logged CRON jobs, or the CRON logging is turned off, which is its default value. To enable CRON logging, open the `blackDuckScanForHub.groovy` file, and change the following values as shown.

  From:

  ```
  @Field final boolean logVerboseCronLog = false
  ```

  To:

  ```
  @Field final boolean logVerboseCronLog = true
  ```

  Then, save the `blackDuckScanForHub.groovy` file, and restart Eclipse. If after performing these steps, re-testing, and still nothing displays in the logged CRON jobs, there is an issue with your CRON expressions.

For more information on editing the `blackDuckScanForHub.groovy` file, refer to Configuring the Hub Artifactory Plugin Scanner Option on page 8.

For more information regarding CRON expressions, refer to Configuring the Scanning Schedule on page 9.

4.2 Development and Testing Errors

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.

4.3 Error: Project Does Not Exist on the Hub

- If you try to use the Hub plugin 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.

    Solution:

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

4.4 Hub Artifactory Log Files

Hub Artifactory log files may be helpful when troubleshooting. The log files are located in:

```
/var/opt/jfrog/artifactory/logs
```

Logging Level Options

The default log level for user plugins is *warn*. You can change a plugin log level to either:

- info
- debug

To change the logging level, edit the file `${ARTIFACTORY_HOME}/etc/logback.xml` as follows:

- To change to the *info* logging level:
  `<logger name="blackDuckScanForHub"> <level value="info"/> </logger>`
- To change to the *debug* logging level:
  `<logger name="blackDuckScanForHub"> <level value="debug"/> </logger>`

CRON Job Logging

Time/date stamps for your automated scans are captured in the `blackduck_cron_history` log file. The default setting for CRON job logging is *false* (off). If you open the `blackduck_cron_history` log file, and it is blank, this is because your CRON job logging value is set to *false*. Set the value to *true* for capturing the start and stop times for all CRON jobs. For more information, refer to Configuring

Preferences Using the Hub Artifactory Groovy File on page 10.

4.5 Clearing Your Log Files

As of Hub Artifactory version 1.2.0 and higher, you can clear your REST and CLI log files. This prevents your log files from consuming extra disk space, and makes it easier to see the latest logging entries.

To clear your log files:

1. Open a command line window.
2. At the command prompt, type the command:
   `curl -X POST -u admin:password "http://ARTIFACTORY_SERVER/artifactory/api/plugins/execute/testConfig"`
   Where `admin:password` is your user name and password, and the URL points to your environment.

Your old log files are now deleted, and are replaced with blank log files.

Chapter 5: Hub Artifactory Plugin Release Notes

Release 2.0.0

- ARTIFACT CUTOFF DATE now accounts for invalid input.
- New **Scanner** and **Inspector** modes.

Release 1.2.0

- Added a new REST API call to enable the testing of multiple elements of the Hub Artifactory configuration.
- Added a new REST API call to clear the REST and CLI log files.
- Added a new parameter to the `groovy` file which allows you to set a scanning cut-off date.

Release 1.1.0

- Eliminated the `Properties` file by moving all `Properties` file functionality into the `groovy` file.
- The default behavior of scanning is now `dryRun=false`. This parameter does not push the scan results to the Hub until the value is changed to `true`.
- A new log file is added which captures the date/time stamps of all automated scans (CRON jobs).

Release 1.0.0

- Because this is the first release of the Black Duck Hub Artifactory plugin, there are no release note items.

Chapter 6: Black Duck Support

If you have questions, need help, or find issues, contact Black Duck Software. 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, the following third-party integrated products are supported:
    - Jenkins
    - TeamCity
    - Bamboo
    - Team Foundation Server (TFS)
    - Artifactory
    - JIRA
    - Atlassian
    - JFrog Xray
- 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.

6.1 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.

6.2 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.