



BLACKDUCK

Hub Plugin for Python
Version 1.0.0



This edition of the *Hub Plugin for Python* refers to version 1.0.0 of the Black Duck Hub Plugin for Python.

This document created or updated on Monday, April 10, 2017.

Please send your comments and suggestions to:

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

Copyright © 2017 by **Black Duck Software, Inc.**

All rights reserved. All use of this documentation is subject to the license agreement between Black Duck Software, Inc. and the licensee. No part of the contents of this document may be reproduced or transmitted in any form or by any means without the prior written permission of Black Duck Software, Inc.

Black Duck, Know Your Code, and the Black Duck logo are registered trademarks of Black Duck Software, Inc. in the United States and other jurisdictions. Black Duck Code Center, Black Duck Code Sight, Black Duck Export, Black Duck Hub, Black Duck Protex , and Black Duck Suite are trademarks of Black Duck Software, Inc. All other trademarks or registered trademarks are the sole property of their respective owners.

Chapter 1: Hub Python Plugin Installation	1
1.1 Python Plugin Requirements	1
1.2 Downloading and Installing the Hub Python Plugin	1
Chapter 2: Configuring the Python Plugin	3
2.1 Hub Python Configuration File Options	3
2.2 Additional Hub Python Configuration Options	4
Chapter 3: Using the Hub Python Plugin	5
3.1 Using the Hub Python Plugin without Setup Tools	5
Chapter 4: Hub Python Plugin Release Notes	6
Chapter 5: Black Duck Support	7
5.1 Training	8
5.2 Services	8

Chapter 1: Hub Python Plugin Installation

The procedure for installing the Hub Python plugin is as follows:

1. Download and install the Python plugin.
2. Configure the plugin.

Instructions for both steps are in the following sections.

1.1 Python Plugin Requirements

Software Requirements

Before installing the Hub Python plugin, your system must meet the following prerequisites:

- Black Duck Hub version 3.5 or higher.
- Python versions:
 - 2.7.x or higher
 - 3.6 or higher

For Hub installation instructions and system requirements, refer to the *Hub Installation Guide*.

For additional Python information, refer to:

<https://docs.python.org/3/>

Network Requirements

The Hub Python plugin requires internet connectivity. The machine hosting your Python environment must be able to connect to the Hub server.

1.2 Downloading and Installing the Hub Python Plugin

You can download and install the Hub Python plugin using the following process.

* To download and install the Hub Python plugin:

1. *(Recommended)* Run the command: **pip install hub-pip**. You must run this for command for each build environment. *The following installation scenarios are optional.*
2. Navigate to the PyPI repository at <https://pypi.python.org/pypi>.
3. Search for Black Duck.
4. In the search results, select *hub-pip<version number>*, and download.
5. *Optional installation option:* You can also run the command `pip install <path_to_`

`download.tar.gz>.`

6. *Optional:* If you use setup tools, you must perform these steps to generate the Black Duck Input/Output file (BDIO):
 - a. Run **python setup.py install** Running this command builds your project and installs it to your local system. The Hub Python plugin can then access project metadata: project name, version, and requirements. Run this command each time you want to build a new project version.
 - b. Then run **python setup.py hub_pip -c your_config.ini**.

Chapter 2: Configuring the Python Plugin

You can configure the Hub Python plugin for your environment using the following procedure.

Note: You can use the Hub Python plugin without having a Hub user name and password. However, if you do not have Hub credentials, you cannot deploy or check policies.

* To configure the Python plugin:

1. To generate a configuration file, run the following command: `hub_pip create-config <output_path>`.
2. Edit the configuration file using the command line. You can override any values in the configuration file.
3. Supply the following values for the configuration file:
 - a. Your Hub user name.
 - b. Your Hub password.
 - c. Your Hub server URL.
4. If you have a proxy, supply the following proxy values:
 - a. Your proxy host.
 - b. Your proxy port.
 - c. Your proxy user name.
 - d. Your proxy password.
 - e. Your Hub server URL.
5. Optional configuration values are:
 - a. Value for API request timeout: Hub-Timeout: default is 120 seconds.
 - b. Hub-ScanTimeout = 300 seconds; this value determines how long to wait for matching.

2.1 Hub Python Configuration File Options

The configuration file contains various options you can utilize to match your environment and workflow requirements. You can edit the configuration file in the text editor of your choice.

Configuration file fields and values

- Hub-Url = None
- Hub-Username = None
- Hub-Password = None

- Hub-Proxy-Host = None
- Hub-Proxy-Port = None
- Hub-Proxy-Username = None
- Hub-Proxy-Password = None
- Hub-Timeout = 120
- Hub-ScanTimeout = 300
- Hub-CodeLocationName = None
- OutputDirectory = build/output/
- RequirementsFile = None
- IgnoreFailure = False
- CreateFlatDependencyList = True
- CreateTreeDependencyList = True
- CreateHubBdio = True
- DeployHubBdio = True
- CheckPolicies = True

2.2 Additional Hub Python Configuration Options

After you have configured your Hub Python plugin, you may want to use the following optional configuration settings, depending on your environment and workflow needs.

- **Options:** you can override the options value in your configuration file by entering `--OutputDirectory=/testing/`.
- **Passwords:** If you do not want your password stored in the configuration file, you can specify `Hub-Password = None` in the configuration file, and on the command line, enter `--Hub-Password=<your_password>`.
- **Output path:** If you do not pass in an output file, the plugin uses the `OutputDirectory` value specified in the configuration file as the BDIO output location, and shows flat lists and tree lists.
- In the configuration file: if you have specified `DeployHubBdio = True`, the plugin automatically sends the BDIO to the Hub and automatically builds your Bill of Materials (BOM).

Note: Entries in the configuration file are case-sensitive.

Chapter 3: Using the Hub Python Plugin

The Hub Python plugin is accessed from the command line, and has no graphical user interface. After you have configured your Hub Python plugin, you can call it as follows.

* To use the Hub Python plugin:

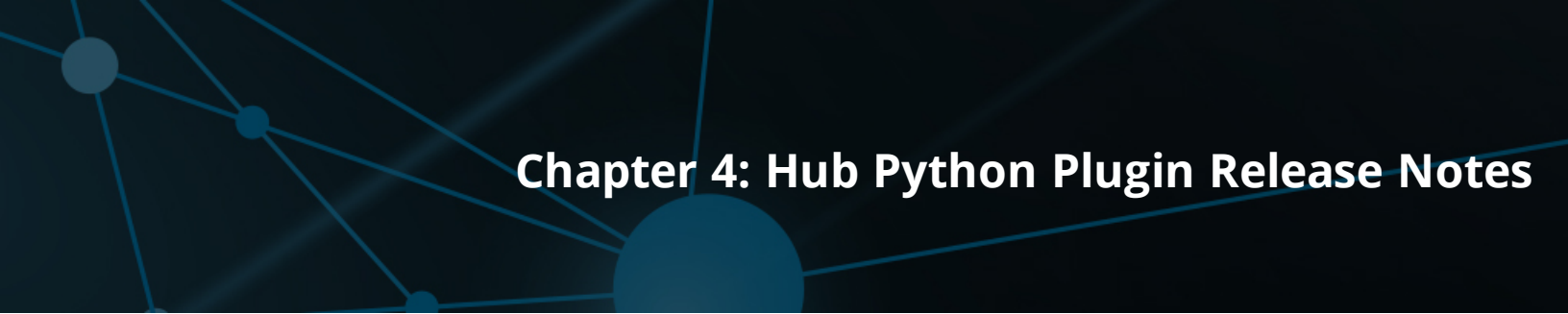
1. On the command line, navigate to the root directory of your project.
2. Run the following command for each build to generate and deploy the BDIO: `python setup.py hub_pip -c your_config.ini`.

3.1 Using the Hub Python Plugin without Setup Tools

If you are not using setup tools, you can still run the Hub Python plugin. With this method you must specify the project name and version that is installed to your pip-cache.

* To use the Hub Python plugin without setup tools:

1. Install your project to the pip cache using your build tool of choice.
2. Run the command: `hub_pip <project_name> <project_version> -c <path/to/hub_config.ini>`.



Chapter 4: Hub Python Plugin Release Notes

Release 1.0.0

- First release of Hub Python plugin.

Chapter 5: 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.

5.1 Training

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

View the full catalog of online courses and try some free courses at: <http://academy.blackducksoftware.com>.

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

5.2 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.