

**FlexNet Code Insight 2018 R3
Plugin Guide**

Legal Information

Book Name: FlexNet Code Insight 2018 R3 Plugin Guide
Part Number: FNCI-2018R3-PG00
Product Release Date: October 2018

Copyright Notice

Copyright © 2018 Flexera.

This publication contains proprietary and confidential information and creative works owned by Flexera and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera is strictly prohibited. Except where expressly provided by Flexera in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera, must display this notice of copyright and ownership in full.

Intellectual Property

For a list of trademarks and patents that are owned by Flexera, see <https://www.flexera.com/producer/company/about/intellectual-property/>. All other brand and product names mentioned in Flexera products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.

Contents

- 1 FlexNet Code Insight 2018 R3 Plugin Guide 5**
- 2 Using & Developing FlexNet Code Insight Plugins 7**
 - Scan Agent Plugins..... 7**
 - Standard Scan Plugins 7
 - Prerequisites for the Generic Scan Agent Plugin 8
 - Scan Agent Framework 8**
 - Available Classes and Methods. 9
 - Property Settings. 10
 - Contents of the Generic Scan Agent Plugin and SDK 11
 - Running the Generic Plugin. 11
 - Writing a Custom Scan Agent Plugin. 11
 - Deploying a Custom Scan Agent Plugin 12

1

FlexNet Code Insight 2018 R3 Plugin Guide

FlexNet Code Insight empowers organizations to take control of and manage their use of open source software (OSS) and third-party components. It helps development, legal and security teams use automation to create a formal OSS strategy that balances business benefits and risk management.

Contacting Support

If you need further support, please submit your questions through our online **Customer Community** portal:

<https://flexeracommunity.force.com/customer/>

If you do not have a login to the Customer Community, you can request one on the **Login Request** page of our site:

<https://flexeracommunity.force.com/customer/CCLoginRequest>

If you are unable to use the steps above, please visit the following site for other options to reach out to Flexera Support:

<https://flexeracommunity.force.com/customer/CCContactSupport>

2

Using & Developing FlexNet Code Insight Plugins

This chapter provides information about developing a FlexNet Code Insight scan agent plugin to perform remote scanning on an Engineering server containing the specific codebase to scan. The following topics are covered in this chapter:

- [Scan Agent Plugins](#)
- [Scan Agent Framework](#)

Scan Agent Plugins

FlexNet Code Insight provides a Scan Agent Framework to help you create your own scan plugins to perform scanning on a remote codebase (one that is not on the Code Insight server) as part of the software development process. The scan plugin determines the appropriate codebase to be scanned within the context of an Engineering application (such as an IDE, source management, artifact repository, CI, build, testing, installation, or other application). It then invokes a scan and sends results back to FlexNet Code Insight for inventory review, management, and security alerting. The generated inventory items can be automatically reviewed based on license policies or manually reviewed by legal and security. Additionally, security vulnerability alerts are sent for any inventory items that have new security vulnerabilities. For information about security vulnerability alerts, see “Security Vulnerability Alerts” in the *FlexNet Code Insight User guide*.

FlexNet Code Insight offers a set of standard scan agent plugins that can be deployed immediately, as described in the next section. However, to author custom plugins, you need to use the Scan Agent Framework. You can create the custom plugin from scratch or use the generic scan agent plugin as a template, all of which is described in later sections in this chapter.

Standard Scan Plugins

FlexNet Code Insight provides the following standard scan agent plugins:

- **Jenkins:** Allows automated scanning of a Jenkins workspace as part of the build process.
- **Artifactory:** Allows automated scanning of Artifactory repositories to identify non-compliant artifacts.
- **Docker:** Allows automated scanning of Docker images on a Docker server.
- **Bamboo:** Allows automated scanning of a Bamboo workspace as part of the build process.

- **Maven:** Allows automated scanning of Maven projects as part of the build process.
- **Gradle:** Allows automated scanning of Gradle projects as part of the build process.
- **Ant:** Allows automated scanning of Apache Ant as part of the build process.
- **Visual Studio:** Allows automated scanning of VSTS workspace as part of the build process.
- **TeamCity:** Allows automated scanning of TeamCity projects as part of the build process.
- **GitLab:** Allows automated scanning of GitLab projects as part of the build process.
- **Generic scan plugin:** Enables you to scan arbitrary file systems of your choice or to create your own scan plugin (as described in this guide). It also easily integrates with other Engineering systems to perform scans as part of a build process (as described in the “Scan Integration With Build Environments Through the Generic Scan Plugin” section in the *FlexNet Code Insight Installation and Configuration Guide*)

The output of each of these plugins is automatically uploaded to FlexNet Code Insight for inventory review, management, and security alerting. These functions can be performed via the FlexNet Code Insight user interface.



Note • Before attempting to run any of the plugins, ensure that you have installed and configured FlexNet Code Insight and the necessary plugin. The scan plugins require a minimum of 2GB of heap size to operate. See “Installing & Configuring FlexNet Code Insight Plugins” in the *FlexNet Code Insight Installation and Configuration guide*.

Prerequisites for the Generic Scan Agent Plugin

In addition to the prerequisites for the FlexNet Code Insight application itself, which must already have been properly installed, the generic scan agent plugin has a few other prerequisites:

- At least 2 GB of Java heap space allocated to the JVM under which the scan agent plugin will be executing
- 64-bit Java Runtime Environment JRE 8+
- FlexNet Code Insight Server 2017 R3 or later
- An inventory-only project is created in FlexNet Code Insight (see “Creating a Project” in the *FlexNet Code Insight User Guide*).



Note • Plugins may require a larger heap size, depending upon the scanning needs of your environment. For more information, see “Installing & Configuring FlexNet Code Insight Plugins” in the *FlexNet Code Insight Installation and Configuration guide*.

Scan Agent Framework

The Scan Agent framework provides the ability to scan codebases at various phases of the development process on remote engineering systems, in the appropriate context. The framework provides the backbone for the processing that the user-created scan agent plugins require.

The Scan Agent Framework is itself a set of Java APIs. The source code for a generic scan plugin is provided to demonstrate the API flow. By creating a plugin that takes advantage of the Scan Agent Framework, you can tailor FlexNet Code Insight’s powerful scanning capabilities to your computing environment and incorporate them into your business process flow.

Specifically, the framework provides the following features for the scan agent plugin:

- Tests the connection from your plugin to the FlexNet Code Insight server and provides error handling with error messages. These messages include the FlexNet Code Insight version, any invalid URLs passed, invalid user access tokens, and invalid project names.
- Passes environmental and system properties.
- Downloads and installs a remote scanner on the engineering system where the scan agent plugin is executed.
- Invokes the scan called for in the plugin, and sends the scan results back to FlexNet Code Insight.
- Processes and displays logging content to a system console in the scan agent plugin environment.
- Generates a verbose scanner log for further information and debugging of failed scans.


Available Classes and Methods

The Scan Agent Framework provides the following classes and methods that can be used to build a scan plugin:

Table 2-1 • Available Classes and Methods

Class	Description	Settings/Methods
ExecutionContext	Stores all the information needed for a scan to run and publish results to FlexNet Code Insight server. This class should only be initialized by calling <code>ExecutionContext.getInstance(...)</code> method.	<ul style="list-style-type: none"> • ExecutionContext <code>getInstance(Properties props, PrintStream logger)</code>: Initializes ExecutionContext for the current scan <p>Parameters:</p> <p><code>props</code>: properties required to scan and publish results</p> <p><code>logger</code>: output stream where all the useful logging will be sent; if logger is null, all logging will be redirected to console output. For more information about parameters for this class, see Property Settings.</p>

Table 2-1 • Available Classes and Methods

Class	Description	Settings/Methods
ScanExecutor	<p>Executes scan and publish results request made by client plugins.</p>  <p>Note • This class requires a valid <i>ExecutionContext</i> instance to be able to operate.</p>	<ul style="list-style-type: none"> ● ScanExecutor(ExecutionContext executionContext): constructor ● String testConnection(): Validates the connection to the FlexNet Code Insight server and user authorization token. <p>Valid returned strings:</p> <ul style="list-style-type: none"> ● Success ● Server Not Found ● Invalid Auth Token Found ● User not authorized to access the project ● String scanCodebase(List<String> paths): Updates the embedded scanner if updates are found, scans the paths provided as input, publishes the results to the Code Insight server as provided in <i>ExecutionContext</i>. <p>Parameters:</p> <ul style="list-style-type: none"> ● paths: list of absolute paths to be scanned <p>Return:</p> <ul style="list-style-type: none"> ● SUCCESS or FAILURE as string, for more details look at logger output stream <p>See also:</p> <ul style="list-style-type: none"> ● ExecutionContext

Property Settings

The following table lists property settings that you can update:

Table 2-2 • Property Settings

Property	Required/Optional?	Description
codeinsight.server	Required	Code Insight server URI (for example, http(s)://your.codeinsight.server:port/).
codeinsight.auth.token	Required	Authorized user token generated from the FlexNet Code Insight server web UI.
codeinsight.project.name	Required	The FlexNet Code Insight project name where all the findings will be published.
plugin.root.path	Required	Local root path that can be replaced with path prefix.

Table 2-2 • Property Settings

Property	Required/Optional?	Description
plugin.project.name	Optional	Name of the plugin (for example, "Codeinsight Jenkins plugin").
plugin.project.description	Optional	Plugin's instance name/build number or tag can be sent to the Code Insight server.
plugin.path.prefix	Optional	Prefix to be added to file paths for display to the user. For example, it could be the URL to a Jenkins workspace.

Contents of the Generic Scan Agent Plugin and SDK

The FlexNet Code Insight generic scan agent SDK includes the following files:

- **/readme.txt**: Instructions on how to use the SDK.
- **/lib**: All dependent jar files to run the plugin. (This directory is located in both the /generic-plugin-binary and the /generic-plugin-source folders)
- **/generic-plugin-binary**: Example binary to execute the plugin.
- **/generic-plugin-source**: Example code used to build the plugin.
- **/generic-plugin-source/pom.xml+assembly.xml**: Maven build files to compile and build the plugin.

Running the Generic Plugin



Task *To run the generic scan agent plugin, do the following:*

1. Extract the contents of the *CodeInsight2018Plugins.zip* file. This file is available from the Customer Community page of the Flexera website: <https://flexeracommunity.force.com/customer/>.
2. Locate the `code-insight-agent-sdk-generic/example` folder and copy it to your hard drive.
3. Run the following from a command line as a Java application:

```
java -Dflx.agent.logLevel=info -jar codeinsight-generic-VERSION.jar -server "<HOST>:<PORT>" -token "Bearer <JWT_TOKEN>" -proj "<Codeinsight Project Name>" -root "</path/to/the/codebase>" -scandirs "</path/to/the/codebase/PROJECT>"
```

Alternatively, you can set options, and run the application using `run_scan.bat` or `run_scan.sh`.

Writing a Custom Scan Agent Plugin

To write a custom scan agent plugin, perform the following tasks:

- Review the example generic scan agent plugin for the code structure and build configuration.

- Identify the Engineering system with which you will be integrating the FlexNet Scan Agent. For example, it could be the Jenkins Continuous Integration server. This Engineering system will be referred to as the Integration Server.
- Using the APIs provided by the integration server, write code to pull in all the codebase files that you want to scan into a single folder. This will be the folder path that you will be passing to the plugin.
- Use the scan agent framework APIs to connect to the FlexNet Code Insight server and scan the folder that contains the desired code base files.



Note • *The plugin will typically execute on the same computer system as the Integration Server.*

Deploying a Custom Scan Agent Plugin

To deploy a custom Scan Agent plugin, do the following:

- Ensure you follow the system pre-requisites for the plugins. See [Prerequisites for the Generic Scan Agent Plugin](#).
- Deploy the plugin on the appropriate Integration Server.
- Install and configure the FlexNet Code Insight server.
- Invoke the remote scans on the integration servers, and review the results in the FlexNet Code Insight UI.