opentext™

OpenText™ Fortify Jenkins Plugin

Software Version: 23.1

User Guide

Document Release Date: November 2023
Software Release Date: November 2023

Legal Notices

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Copyright Notice

Trademark Notices

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

This document was produced on November 29, 2023. To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support/documentation

Contents

Preface

Contacting Fortify Customer Support

Visit the Support website to:

- Manage licenses and entitlements
- Create and manage technical assistance requests
- Browse documentation and knowledge articles
- Download software
- Explore the Community

https://www.microfocus.com/support

For More Information

For more information about Fortify software products:

https://www.microfocus.com/cyberres/application-security

About the Documentation Set

The Fortify Software documentation set contains installation, user, and deployment guides for all Fortify Software products and components. In addition, you will find technical notes and release notes that describe new features, known issues, and last-minute updates. You can access the latest versions of these documents from the following OpenText Product Documentation website:

https://www.microfocus.com/support/documentation

To be notified of documentation updates between releases, subscribe to Fortify Product Announcements on the OpenText Community:

https://community.microfocus.com/cyberres/fortify/w/fortify-product-announcements

Fortify Product Feature Videos

You can find videos that highlight Fortify products and features on the Fortify Unplugged YouTube channel:

https://www.youtube.com/c/FortifyUnplugged

Change Log

The following table lists changes made to this document. Revisions to this document are published between software releases only if the changes made affect product functionality.

Software Release / Document Version	Changes
23.1	Updated: • Added information about using a trusted certificate (see "Preparing to Work with Fortify Software Security Center and Fortify ScanCentral SAST" on page 14)
22.1	Updated: • Added the ability to use Jenkins Credential to provide authentication tokens for connecting to Fortify Software Security Center and Fortify ScanCentral SAST (see "Configuring Global Settings for the Fortify Jenkins Plugin" on page 14) • You can now specify proxy settings in the Jenkins Plugin Manager for connecting to Fortify Software Security Center and the Fortify Rulepack update server (see "Configuring Global Settings for the Fortify Jenkins Plugin" on page 14) • Option added to accept the public key for downloading Fortify Software Security Content from Fortify Software Security Center in Freestyle and pipeline projects • Improved the selection of a filter set for uploading local scan results to Fortify Software Security Center in Freestyle projects • The fortifyUpload pipeline step now returns the number of issues that match a specified build failure criteria (see "fortifyUpload Step" on page 61)
21.2	Updated: • New field added for maximum number of application versions per list and option to disable local scans ("Configuring Global Settings for the Fortify Jenkins Plugin" on page 14)

Software Release / Document Version	Changes
	• New ability to translate with MSBuild and new option to skip the project preparation build step (see "Creating a Post-Build Action to Translate and Scan Remotely" on page 22 and "fortifyRemoteAnalysis Step" on page 41) • New ability to use a Gradle or Maven executable configured with Jenkins Global Tool Configuration (see "Creating a Post-Build Action to Translate Locally and Scan Remotely" on page 25 and "Creating a Post-Build Action to Translate and Scan Locally" on page 30) • The timeout option is now available in the Snippet Generator (see "fortifyUpload Step" on page 61)
21.1.0	Added: • "Using a File to Configure Global Settings for the Fortify Jenkins Plugin" on page 19

Chapter 1: Introduction

Use the Fortify Jenkins Plugin in your continuous integration builds to identify security issues in your source code with OpenText™ Fortify Static Code Analyzer. A Fortify Static Code Analyzer security analysis consists of the following phases:

- Translate all source code files into intermediate files
- Perform analysis on the source and produce scan results

The Fortify Jenkins Plugin provides three ways to analyze your source code:

- Offload the complete analysis (translation and analysis phases) to OpenText™ Fortify ScanCentral SAST
- Perform a translation on the local system and then offload the more CPU-intensive analysis phase to Fortify ScanCentral SAST
- Perform the complete analysis (translation and analysis phases) on the local system

  You can run the analysis locally with Gradle, Maven, MSBuild, and Visual Studio (devenv). You can also analyze your source code without a build tool.

After the Fortify Static Code Analyzer analysis is complete, you can upload the scan results to a OpenText™ Fortify Software Security Center server.

For complete analysis run locally only: If you upload the scan results to a Fortify Software Security Center server, you can view the analysis result details within Jenkins. The results provide metrics for each build and an overview of the results, without requiring you to log into Fortify Software Security Center.

This guide provides instructions for how to install, configure, and use the plugin.

This section contains the following topics:

Software Requirements

The Fortify Jenkins Plugin works with the software packages listed in the following table. Your specific requirements depend on the build tools you are using. This table also provides information to help you prepare to add Fortify Static Code Analyzer analysis to your jobs.

Software	Version	Notes
Fortify Static Code Analyzer	18.20 or later	To scan your project locally with Fortify Static Code Analyzer, you must either have the path to the Fortify Static Code Analyzer installation directory so you can specify it in the configuration or make sure that the PATH environment variable includes the sourceanalyzer executable (see "Configuring Global Settings for the Fortify Jenkins Plugin" on page 14). For supported versions of Gradle, Maven, MSBuild, and Visual Studio, see the *Fortify Software System Requirements* document for your version of Fortify Static Code Analyzer in Fortify Static Code Analyzer and Tools Documentation. **Note:** Performing remote analysis requires Fortify Static Code Analyzer version 19.2.0 or later.
Fortify Software Security Center (Optional)	18.20 or later	To upload scan results to Fortify Software Security Center, to trigger a build failure based on the scan results, and to see scan results in Jenkins, make sure that you have: • The Fortify Software Security Center URL • A Fortify Software Security Center authentication token of type CIToken (see "Preparing to Work with Fortify Software Security Center and Fortify ScanCentral SAST" on page 14) To perform a remote analysis, make sure that you have a Fortify Software Security Center authentication of type ScanCentralCtrlToken (see "Preparing to Work with Fortify Software Security Center and Fortify ScanCentral SAST" on page 14).
Fortify ScanCentral SAST or Micro	19.2.0 or later	To perform a Fortify Static Code Analyzer analysis on a remote system using Fortify ScanCentral SAST, make sure

Software	Version	Notes
Focus Fortify CloudScan (Optional)		that you have properly configured Fortify ScanCentral SAST and you have the ScanCentral Controller URL. **Note:** If you will upload remote scan results to Fortify Software Security Center, then you do not need to provide a ScanCentral Controller URL. The Fortify Jenkins Plugin automatically determines the ScanCentral Controller that is associated with Fortify Software Security Center. For languages that are supported for remote translation and general system requirements for Fortify ScanCentral SAST, see the *Fortify Software System Requirements* document for your version of Fortify ScanCentral SAST in Fortify Software Security Center Documentation.

To integrate the scan with Maven, you must install the Fortify Maven plugin, which is available when you install Fortify Static Code Analyzer. OpenText recommends that you use the same Fortify Maven Plugin version as the Fortify Static Code Analyzer version and that you install the source version of the Fortify Maven Plugin rather than the binary version. You must install the Fortify Maven Plugin for the same user who is running Jenkins. If you use a proxy, then you must configure proxy settings for the Fortify Maven Plugin. For information, see the Settings Reference at https://maven.apache.org. For more information about build integration with the Fortify Maven Plugin, see the *OpenText™ Fortify Static Code Analyzer User Guide*.

To integrate the scan with devenv, you must install the Fortify Extension for Visual Studio. For more information, see the *OpenText™ Fortify Extension for Visual Studio User Guide* in Fortify Static Code Analyzer and Tools Documentation.

Related Documents

This topic describes documents that provide information about Fortify software products.

Note: You can find the Fortify Product Documentation at https://www.microfocus.com/support/documentation. Most guides are available in both PDF and HTML formats.

Fortify ScanCentral SAST

The following document provides information about Fortify ScanCentral SAST. Unless otherwise noted, this document is available on the Product Documentation website at https://www.microfocus.com/documentation/fortify-software-security-center.

Document / File Name	Description
OpenText™ Fortify ScanCentral SAST Installation, Configuration, and Usage Guide SC_SAST_Guide_<*version*>.pdf	This document provides information about how to install, configure, and use Fortify ScanCentral SAST to streamline the static code analysis process. It is written for anyone who intends to install, configure, or use Fortify ScanCentral SAST to offload the resource-intensive translation and scanning phases of their Fortify Static Code Analyzer process.

Fortify Software Security Center

The following document provides information about Fortify Software Security Center. Unless otherwise noted, this document is available on the Product Documentation website at https://www.microfocus.com/documentation/fortify-software-security-center.

Document / File Name	Description
OpenText™ Fortify Software Security Center User Guide SSC_Guide_<*version*>.pdf	This document provides Fortify Software Security Center users with detailed information about how to deploy and use Fortify Software Security Center. It provides all of the information you need to acquire, install, configure, and use Fortify Software Security Center. It is intended for use by system and instance administrators, database administrators (DBAs), enterprise security leads, development team managers, and developers. Fortify Software Security Center provides security team leads with a high-level overview of the history and current status of a project.

Fortify Static Code Analyzer

The following documents provide information about Fortify Static Code Analyzer. Unless otherwise noted, these documents are available on the Product Documentation website at https://www.microfocus.com/documentation/fortify-static-code.

Document / File Name	Description
OpenText™ Fortify Static Code Analyzer User Guide SCA_Guide_<*version*>.pdf	This document describes how to install and use Fortify Static Code Analyzer to scan code on many of the major programming platforms. It is intended for people responsible for security audits and secure coding.
OpenText™ Fortify Static Code Analyzer Applications and Tools Guide SCA_Apps_Tools_<*version*>.pdf	This document describes how to install Fortify Static Code Analyzer applications and tools. It provides an overview of the applications and command-line tools that enable you to scan your code with Fortify Static Code Analyzer, review analysis results, work with analysis results files, and more.
OpenText™ Fortify Static Code Analyzer Custom Rules Guide SCA_Cust_Rules_Guide_<*version*>.zip	This document provides the information that you need to create custom rules for Fortify Static Code Analyzer. This guide includes examples that apply rule-writing concepts to real-world security issues. **Note:** This document is included only with the product download.
OpenText™ Fortify License and Infrastructure Manager Installation and Usage Guide LIM_Guide_<*version*>.pdf	This document describes how to install, configure, and use the Fortify License and Infrastructure Manager (LIM), which is available for installation on a local Windows server and as a container image on the Docker platform.

Chapter 2: Installation and Configuration

This chapter describes how to install and configure the Fortify Jenkins Plugin.

This section contains the following topics:

Installing the Fortify Jenkins Plugin

To install the Fortify Jenkins Plugin, you must have Jenkins installed on your system. See the *Fortify Software System Requirements* document for the supported Jenkins versions.

Note: These instructions describe a third-party product and might not match the specific, supported version you are using. See your product documentation for the instructions for your version.

To install the Fortify Jenkins Plugin:

1. From Jenkins, select **Manage Jenkins > Manage Plugins**.
2. On the **Plugin Manager** page, select the **Available** tab.
3. In the **Filter** box, type `Fortify`.
4. Select the check box for the **Fortify** plugin, and then click either **Install without restart** or **Download and install after restart**.

For more information about how to install Jenkins plugins, see the Jenkins website.

Verifying the Fortify Jenkins Plugin Installation

Note: These instructions describe a third-party product and might not match the specific, supported version you are using. See your product documentation for the instructions for your version.

To verify that the Fortify Jenkins Plugin is installed:

1. Open a browser window and navigate to the Jenkins server URL.
2. From the Jenkins menu, select **Manage Jenkins > Manage Plugins**.

3. On the **Plugin Manager** page, select the **Installed** tab.
4. Verify that the list of installed plugins includes the **Fortify** plugin.

Preparing to Work with Fortify Software Security Center and Fortify ScanCentral SAST

If Fortify Software Security Center or the Fortify ScanCentral SAST Controller uses an SSL connection from an internal certificate authority or a self-signed certificate, you must place the certificate into the Java Runtime certificate store in the JRE of the Jenkins controller.

To perform either of the following tasks, you need to have an authentication token created in Fortify Software Security Center. You use this authentication token to configure the Fortify Jenkins Plugin to communicate with Fortify Software Security Center or Fortify ScanCentral SAST. The following table describes the tasks and the token type needed to perform the task.

Task	Token Type
Upload local Fortify Static Code Analyzer scan results to Fortify Software Security Center	CIToken
Use Fortify ScanCentral SAST to perform a remote Fortify Static Code Analyzer analysis (this includes the ability to upload the scan results to Fortify Software Security Center)	ScanCentralCtrlToken

Obtain an authentication token from your Fortify Software Security Center administrator or see the *OpenText™ Fortify Software Security Center User Guide* in Fortify Software Security Center Documentation for instructions.

Configuring Global Settings for the Fortify Jenkins Plugin

Note: These instructions describe a third-party product and might not match the specific, supported version you are using. See your product documentation for the instructions for your version.

To configure your Jenkins server so that it can analyze your project and upload results to Fortify Software Security Center using the Fortify Jenkins Plugin:

1. Open a browser window and navigate to the Jenkins server URL.
2. From the Jenkins menu, select **Jenkins > Manage Jenkins > System**.

3. To analyze (translate or scan) your project locally with Fortify Static Code Analyzer, you can create a Jenkins environment variable to specify the location of the Fortify Static Code Analyzer executable. Otherwise, the Fortify Jenkins Plugin looks for the executable on the system Path variable.

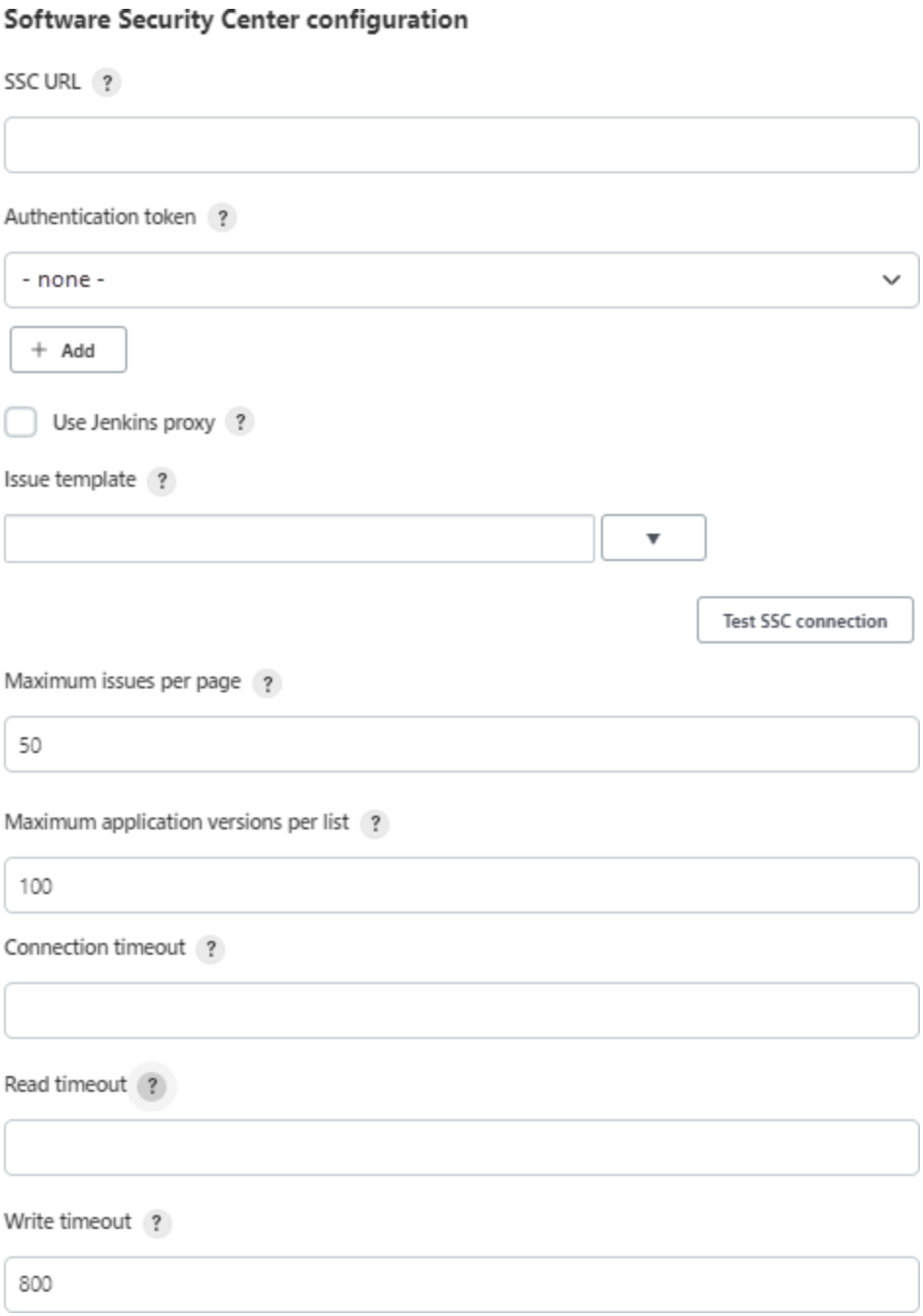
   You can use build tools that you have set up with the Jenkins Global Tool Configuration in the Fortify Jenkins Plugin. Alternatively, you can create Jenkins environment variables to specify the location of a required build tool executable.

   The following table describes the environment variables you can create in **Global properties**.

   Note: Do not use paths that end in `/bin`. The Fortify Jenkins Plugin looks for the location of an executable in the following order: Jenkins Global Tool Configuration (Gradle and Maven only), Jenkins environment variable, the PATH system environment variable, and lastly the build's workspace.

Name	Value	Description
`FORTIFY_HOME`	*<sca_install_dir>*	Specify the path where Fortify Static Code Analyzer is installed. For example, the default location on Windows for versions 23.1.0 or later is `C:\Program Files\Fortify\Fortify_SCA_<sca_version>`.
`GRADLE_HOME`	*<gradle_install_dir>*	(Optional) Specify the path where Gradle is installed.
`M2_HOME` or `MAVEN_HOME`	*<maven_install_dir>*	(Optional) Specify the path where Apache Maven is installed. **Note:** The Fortify Maven Plugin must be installed. See "Software Requirements" on page 9 for more information.

4. To upload results from a local analysis to Fortify Software Security Center, scroll down to the **Fortify Assessment** section, and then do the following in the **Software Security Center configuration** section:



   a. In the **SSC URL** box, type the Fortify Software Security Center server URL provided by your Fortify Software Security Center administrator.
   b. Provide a Fortify Software Security Center **Authentication token** by doing the following:
      i. Click **Add > Jenkins** to open the Jenkins Credentials Provider dialog box.
      ii. From the **Kind** menu select **Fortify Connection Token**.

iii. Provide description information so you can easily identify the credential.

iv. In the **Token** box, type the decoded value of a Fortify Software Security Center authentication token of type CIToken.

v. Click **Add**.

c. (Optional) To connect to Fortify Software Security Center with a proxy server, select **Use Jenkins proxy**.

The Fortify Jenkins Plugin will use the proxy settings configured in the Jenkins Plugin Manager when connecting to a Fortify Software Security Center server or a Fortify Rulepack update server. The Jenkins Plugin Manager allows you to exclude servers from using a proxy in case a proxy is only required for one server and not the other.

d. (Optional) To test the connection to Fortify Software Security Center, click **Test SSC connection**.

e. From the **Issue template** list, select the appropriate issue template for your projects.

Fortify Software Security Center uses the selected issue template when it creates new applications. The issue template optimizes the categorization, summary, and reporting of the application version data.

Note: If no issue template is specified, Fortify Jenkins Plugin creates the application version using the default issue template settings in Fortify Software Security Center.

f. (Optional) To specify the maximum number of issues to display per page in the results breakdown table, type a number in the **Maximum issues per page** box.

Note: This setting controls the **Issue Breakdown** table view. The default is 50 issues per page.

g. (Optional) To specify the maximum number of application versions to display in lists for a Fortify Assessment post-build action configuration, type a number in the **Maximum application versions per list** box.

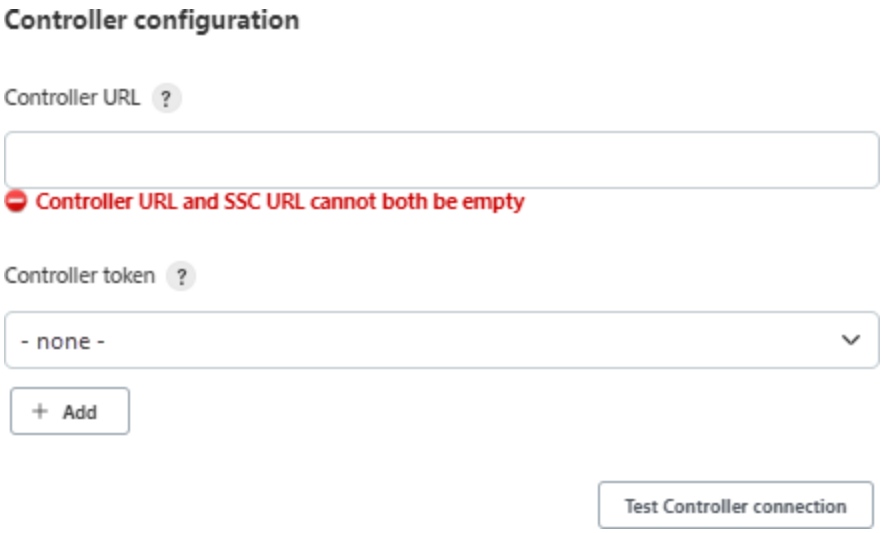
Note: The default is 100 application versions per list. You can type or search for the application name if the application that you want does not appear within the maximum application versions listed.

h. (Optional) To specify how long to wait to connect to Fortify Software Security Center before timing out, type the time in seconds the **Connection timeout** box. A value of 0 means no timeout. The default is 10 seconds.

i. (Optional) To specify how long to wait for a response from Fortify Software Security Center before timing out, type the time in seconds in the **Read timeout** box. A value of 0 means no timeout. The default is 10 seconds.

j. (Optional) To specify how long to allow request data to be sent to Fortify Software Security Center before timing out, type the time in seconds in the **Write timeout** box. A value of 0 means no timeout. The default is 10 seconds.

5. To perform a Fortify Static Code Analyzer analysis on a remote system, do the following in the **Controller configuration** section:



   a. In the **Controller URL** box, type the ScanCentral Controller URL.

   > **Note:** If you specify a URL in the **Software Security Center configuration** section (**SSC URL**), then the Fortify Jenkins Plugin automatically determines the ScanCentral Controller URL from Fortify Software Security Center and you do not need to provide a ScanCentral Controller URL.

   The format for the ScanCentral Controller URL is: `<protocol>://<controller_host>:<port>/scancentral-ctrl` (for example: `https://myControllerHost.com:8443/scancentral-ctrl`).

   b. Provide a Fortify Software Security Center **Controller token** by doing the following:
      i. Click **Add > Jenkins** to open the Jenkins Credentials Provider dialog box.
      ii. From the **Kind** menu select **Fortify Connection Token**.
      iii. Provide description information so you can easily identify the credential.
      iv. In the **Token** box, type the decoded value of a Fortify Software Security Center authentication token of type ScanCentralCtrlToken.
      v. Click **Add**.

   c. (Optional) To test the connection to Fortify ScanCentral SAST, click **Test Controller connection**.

6. To disable Fortify Static Code Analyzer scans on the local system, select the **Disable local scans** check box.

   > **Note:** Fortify Static Code Analyzer translation on the local system is still allowed with this setting selected.

7. Click **Save**.

Using a File to Configure Global Settings for the Fortify Jenkins Plugin

If you have the Jenkins Configuration as Code plugin installed and configured, you can set up a text-based configuration for the Fortify Jenkins Plugin. The Fortify Jenkins Plugin entries belong in the `unclassified` root element. For more about the Jenkins Configuration as Code project, see the Jenkins documentation.

The following is an example Fortify Jenkins Plugin global configuration YAML file:

```
unclassified:
  fortifyPlugin:
    url: "https://MySscHost:8443/ssc"
    sscTokenCredentialsId: "MySscHost_CIToken"
    isProxy: false
    projectTemplate: "Prioritized High Risk Issue Template"
    breakdownPageSize: 50
    connectTimeout: 10
    readTimeout: 10
    writeTimeout: 10
    ctrlUrl: "https://MyControllerHost:8443/scancentral-ctrl"
    ctrlTokenCredentialsId: "MyControllerHost_ScanCentralCtrlToken"
    disableLocalScans: true
    appVersionListLimit: 50
```

The following table describes the keys the Fortify Jenkins Plugin uses.

Key	Description
`url`	A Fortify Software Security Center URL.
`sscTokenCredentialsId`	The ID for a Jenkins credential of the type Fortify Connection Token that contains the Fortify Software Security Center authentication token of type CIToken.
`isProxy`	Whether to use the Jenkins Plugin Manager proxy settings to connect to Fortify Software Security Center and to update Fortify security content. The valid values are `true` and `false`.
`projectTemplate`	A Fortify Software Security Center issue template.
`breakdownPageSize`	The maximum number of issues to display per page in the breakdown table. The default is 50 issues per page.

Key	Description
`connectTimeout`	Time in seconds to wait for a connection to be established with before timing out.
`readTimeout`	Time in seconds to wait for a response from Fortify Software Security Center before timing out.
`writeTimeout`	Time in seconds for request data to be sent to Fortify Software Security Center before timing out.
`ctrlUrl`	The Controller URL is required to perform a Fortify Static Code Analyzer analysis on a remote system. You can leave this empty if you specify a Fortify Software Security Center URL with the `url` key. **Note:** If a Fortify Software Security Center URL is specified (with the `url` key), then the Fortify Jenkins Plugin uses the Controller associated with Fortify Software Security Center and any Controller URL specified here is ignored.
`ctrlTokenCredentialsId`	The ID for a Jenkins credential of the type Fortify Connection Token that contains the decoded value of a Fortify Software Security Center authentication token of type ScanCentralCtrlToken.
`disableLocalScans`	Whether to disable local Fortify Static Code Analyzer scans from being performed. By default, local scans are enabled. The valid values are `true` and `false`.
`appVersionListLimit`	The maximum number of application versions to display in lists for a Fortify Assessment post-build action configuration. The default is 100 application versions per list.

Note: If you downgrade to a previous version of the Fortify Jenkins Plugin that does not support the Jenkins Configure as Code plugin, you must remove the fortifyPlugin entries from the JCasC YAML file.

Preparing Docker to Work with the Fortify Jenkins Plugin

If you run Jenkins in a Docker container, mount the Fortify Static Code Analyzer installation directory in the container to make Fortify Static Code Analyzer executables accessible from Docker. The following example command includes the flag to mount the installation directory in the container:

```
docker container run –p 8080:8080 –v /home/admin/Fortify/Fortify_SCA_23.1.0:/var/jenkins_
home/Fortify/Fortify_SCA_23.1.0 --name=Jenkins jenkins/jenkins –d
```

In the previous example, the value of `FORTIFY_HOME` is `/var/jenkins_home/Fortify/Fortify_SCA_23.1.0`.

Chapter 3: Configuring Fortify Analysis with Freestyle Projects

The Fortify Jenkins Plugin supports Freestyle and Multi-configuration projects. This section describes how to add Fortify analysis as a post-build action for your job.

Note: The Fortify Jenkins Plugin also supports Jenkins Pipeline. For instructions, see "Configuring Fortify Analysis with Pipeline Jobs" on page 38.

This section contains the following topics:

Creating a Post-Build Action to Translate and Scan Remotely

To configure a post-build action to perform a complete analysis on a remote system:

1. From Jenkins, select an existing job to view or create a new job.
   If you selected an existing job, click **Configure** on the job page.
2. In the **Post-build Actions** section, click **Add post-build action**, and then select **Fortify Assessment**.
3. Select **Remote translation & remote scan**.
4. From the **Application type** list, select the type of project you want to analyze. The following table provides instructions for each application type.

Application Type	Description
.NET MSBuild	a. In the **Solution or project file** box, type a project or a solution file for analysis.

Application Type	Description
	b. To exclude disabled projects from the translation, select the **Exclude disabled projects** check box. **Note:** This setting is only valid for Fortify ScanCentral SAST versions 21.1.x or earlier.
Gradle	a. In the **Build file** box, type the name of the build file if it is different than the default of `build.gradle`. b. To include the test source set (for Java projects only) in the scan, select the **Include tests** check box. c. To skip the build invocation used to obtain all project dependencies, select the **Skip build** check box. Use this option if you have a build step earlier in the pipeline and do not want to run the build again. **Note:** This setting is only valid for Fortify ScanCentral SAST versions 20.2.0 or later.
Maven	a. In the **Build file** box, type the name of the build file if it is different than the default of `pom.xml`. b. To include a test scope (for Java projects only) in the scan, select the **Include tests** check box. c. To skip the build invocation used to obtain all project dependencies, select the **Skip build** check box. Use this option if you have a build step earlier in the pipeline and do not want to run the build again. **Note:** This setting is only valid for Fortify ScanCentral SAST versions 20.2.0 or later.
PHP	a. In the **PHP version** box, type the PHP version used in the project.
Python	a. In the **Python version** box, select the Python version used in the project. The default version is 2. b. In the **Python virtual environment** box, type the location (directory) of the Python virtual environment.

Application Type	Description
	c. In the **Python requirements file** box, type the name of the Python project requirements file used to install and collect dependencies.
Other	Use this option to translate and scan other languages.

5. (Optional) To specify Fortify Static Code Analyzer translation options, click **Advanced**, and then specify translation options.

   For descriptions of the available translation options, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation.

   Note: Enclose each option and parameter in double quotes. For example, this option excludes test files from the translation: `"-exclude" "C:/ProjA/tests/*"`.

6. (Optional) To specify Controller settings, add Fortify Static Code Analyzer scan options, custom Rulepacks, or a scan filter file, click **Optional configuration**. The following table describes the optional configuration settings.

Field	Description
Sensor pool	Specify a sensor pool UUID defined in Fortify Software Security Center. By default, Fortify ScanCentral SAST uses the default sensor pool as defined in Fortify Software Security Center.
Notification email	Specify the email address to which the Controller will send notifications.
Fortify SCA scan options	Specify Fortify Static Code Analyzer scan options. For descriptions of the available scan options, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation. **Note:** Enclose each option and parameter in double quotes. In the following example, two analyzers are enabled and verbose status messages are sent to the console for the scan: `"-analyzers" "controlflow,dataflow" "-verbose"`.
Custom Rulepacks	Specify custom rules files (`*.xml`) separated by spaces or a directory that contains custom rules.
Fortify SCA scan	Specify the name of a filter file. You can use a file to filter out specific vulnerability categories, rules, and vulnerability instances from the analysis.

Field	Description
filter file	For more information, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation.

7. To upload the scan results to Fortify Software Security Center:
   a. Select the **Upload Fortify SCA scan results to Fortify Software Security Center** check box.
   b. Specify an application name and an application version from the list of existing applications in Fortify Software Security Center. Provide both application name and application version.

      To search for an application name or version, type any part of the name or version in the box, and then click **Search**. You can also leave the name or version box empty, and then click **Search**. This provides a list of all application names or versions within the configured limit. This search is case-insensitive.

      Note: The number of application names and application versions displayed in both these lists is limited by the maximum application versions per list value specified in the Fortify Jenkins Plugin configuration.
8. Click **Save**.

Creating a Post-Build Action to Translate Locally and Scan Remotely

To configure a post-build action to perform the translation phase on the local system and the scan phase on a remote system:

1. From Jenkins, select an existing job to view or create a new job.

   If you selected an existing job, click **Configure** on the job page.
2. In the **Post-build Actions** section, click **Add post-build action**, and then select **Fortify Assessment**.
3. Select **Local translation & remote scan**.
4. To download Fortify Software Security Content before the scan:
   a. Select the **Update Fortify Security Content** check box.
   b. In the **Update server URL** box, type the URL for the Fortify Rulepack update server.

      The default Fortify Rulepack update server URL is https://update.fortify.com. You can download Fortify Software Security Content from Fortify Software Security Center by specifying a Fortify Software Security Center URL in this box.

      Note: To connect to the Fortify Rulepack update server or the Fortify Software Security Center server with a proxy, you can use the proxy settings configured in Jenkins (see

"Configuring Global Settings for the Fortify Jenkins Plugin" on page 14).

c. (Optional) In the **Locale** box, select the locale for the Fortify Software Security Content.

   The default is English.

d. To automatically accept the public key when you update Fortify Software Security Content from a Fortify Software Security Center server, select the **Accept public key for SSC** server check box.

5. In the **Build ID** box, type a unique identifier for the analysis.
6. (Optional) In the **Maximum heap memory** box, specify the maximum heap memory (in megabytes) as an integer only.

   For example, to specify 48 GB, type `49152`. By default, Fortify Static Code Analyzer automatically allocates memory based on the physical memory available on the system. If you specify the amount of memory in this box, it overrides the default automatic memory allocation.
7. (Optional) In the **Additional JVM options** box, you can add JVM commands.

   Note: Enclose each option and parameter in double quotes in boxes where you can specify multiple values.

   For example: `"-build-label" "label" "-disable-source-bundling"`
8. From the **Application type** list, select the type of application you want to analyze.

   Note: The Fortify Jenkins Plugin looks for the location of the Gradle, Maven, devenv, and MSBuild executables in the following order: Jenkins Global Tool Configuration (Gradle and Maven only), Jenkins environment variable, the PATH system environment variable, and lastly the build's workspace.

Application Type	Description
.NET Devenv	a. In the **Solution or project file** box, type the solution or project file name (or the path to the file). b. (Optional) Specify any additional devenv options.
.NET MSBuild	a. In the **Solution or project file** box, type the solution or project file name (or the path to the file). b. (Optional) Specify any additional MSBuild options.
.NET source code scan	a. In the **.NET framework version** box, specify the .NET framework version used to compile the code. b. (Optional) In the **Libdirs** box, specify a semicolon-separated list of directories where referenced system or third-party DLLs are located.

Application Type	Description
	c. (Optional) In the **Fortify SCA translation options** box, specify any additional Fortify Static Code Analyzer translation options. See the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation for detailed information about the available translation options. d. In the **Source files** box, specify the source files to translate.
Gradle	a. Select whether to use a specific version of Gradle installed on the agent or the Gradle Wrapper. To use a version specific version of Gradle installed on the agent: i. Select **Invoke Gradle**. ii. From the **Gradle version** list, select a version. **Note:** The **Gradle version** option is only available if at least one Gradle version is configured with the Jenkins Global Tool Configuration. Make sure that you select a Gradle version that Fortify Static Code Analyzer supports. For supported Gradle versions, see the *Fortify Software System Requirements* document for your version of Fortify Static Code Analyzer in Fortify Static Code Analyzer and Tools Documentation. To use the Gradle executable defined by the GRADLE_HOME environment variable (see "Configuring Global Settings for the Fortify Jenkins Plugin" on page 14), select **(Default)**. b. In the **Gradle tasks** box, type the Gradle tasks required for your project. c. (Optional) In the **Gradle options** box, type the Gradle options required for your project.
Java	Specify the Java source version, source path, class path, the source files, and any additional Fortify Static Code Analyzer translation options. The only required field is **Source files**. See the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation for more detailed information about the Java translation options.

Application Type	Description
Maven	a. If you have at least one Maven version configured with the Jenkins Global Tool Configuration, you can select a version from the **Maven version** list. Select **(Default)** to use the Maven executable defined by the MAVEN_HOME environment variable (see "Configuring Global Settings for the Fortify Jenkins Plugin" on page 14). **Note:** The **Maven version** option is only available if at least one Maven version is configured with the Jenkins Global Tool Configuration. Make sure that you select a Maven version that Fortify Static Code Analyzer supports. For supported Maven versions, see the *Fortify Software System Requirements* document for your version of Fortify Static Code Analyzer in Fortify Static Code Analyzer and Tools Documentation. b. If you did not run the build previously, then in the **Maven options** box, type `package`. Otherwise, leave this box empty. **Note:** The translation log is in the /target directory that is created when the "package" runs from Maven. Any log file location specified in the Fortify Jenkins Plugin is ignored when the Fortify Maven Plugin performs the translation.
Other	a. (Optional) Provide all the Fortify Static Code Analyzer translation options in the **Fortify SCA translation options** box. See the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation for detailed information about the available translation options. b. Specify the source code to scan in the **Includes list** box.
Advanced	Select **Advanced** if you are familiar with the Fortify Static Code Analyzer command-line interface or want to specify all the translation options without any guidance. Specify all the Fortify Static Code Analyzer translation options including source files. For detailed information about the translation options, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation.

9. (Optional) To exclude files or directories from the translation, add them to the **Exclude list** box.
10. (Optional) Enable the debug or verbose logging options.

11. (Optional) To specify Controller settings, add Fortify Static Code Analyzer scan options, custom Rulepacks, or a scan filter file, click **Optional configuration**. The following table describes the optional configuration settings.

Field	Description
Sensor pool	Specify a sensor pool UUID defined in Fortify Software Security Center. By default, Fortify ScanCentral SAST uses the default sensor pool as defined in Fortify Software Security Center.
Notification email	Specify the email address to which the Controller will send notifications.
Fortify SCA scan options	Specify Fortify Static Code Analyzer scan options. For descriptions of the available scan options, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation. **Note:** Enclose each option and parameter in double quotes. In the following example, two analyzers are enabled and verbose status messages are sent to the console for the scan: `"-analyzers" "controlflow,dataflow" "-verbose"`.
Custom Rulepacks	Specify custom rules files (`*.xml`) separated by spaces or a directory that contains custom rules.
Fortify SCA scan filter file	Specify the name of a filter file. You can use a file to filter out specific vulnerability categories, rules, and vulnerability instances from the analysis. For more information, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation.

12. To upload the scan results to Fortify Software Security Center:
    a. Select the **Upload Fortify SCA scan results to Fortify Software Security Center** check box.
    b. Specify an application name and an application version from the list of existing applications in Fortify Software Security Center. Provide both application name and application version.

       To search for an application name or version, type any part of the name or version in the box, and then click **Search** (🔎). You can also leave the name or version box empty, and then click **Search**. This provides a list of all application names or versions within the configured limit. This search is case-insensitive.

       Note: The number of application names and application versions displayed in both these lists is limited by the maximum application versions per list value specified in the Fortify Jenkins Plugin configuration.
13. Click **Save**.

Creating a Post-Build Action to Translate and Scan Locally

To configure a post-build action to perform a complete analysis on the local system:

1. From Jenkins, select an existing job to view or create a new job.

   If you selected an existing job, click **Configure** on the job page.
2. In the **Post-build Actions** section, click **Add post-build action**, and then select **Fortify Assessment**.
3. Select **Local translation & local scan**.
4. To download Fortify Software Security Content before the scan:
   a. Select the **Update Fortify Security Content** check box.
   b. In the **Update server URL** box, type the URL for the Fortify Rulepack update server.

      The default Fortify Rulepack update server URL is https://update.fortify.com. You can download Fortify Software Security Content from Fortify Software Security Center by specifying a Fortify Software Security Center URL in this box.

      Note: To connect to the Fortify Rulepack update server or the Fortify Software Security Center server with a proxy, you can use the proxy settings configured in Jenkins (see "Configuring Global Settings for the Fortify Jenkins Plugin" on page 14).
   c. (Optional) In the **Locale** box, select the locale for the Fortify Software Security Content.

      The default is English.
   d. To automatically accept the public key when you update Fortify Software Security Content from a Fortify Software Security Center server, select the **Accept public key for SSC** server check box.
5. In the **Build ID** box, type a unique identifier for the analysis.
6. (Optional) In the **Results file** box, type a name for the Fortify Project Results (FPR) file. For example, `MyProjectA.fpr`.

   Note: You do not need to specify the `.fpr` file extension.

   If you do not provide a results file name:

   - If you are running a Fortify Static Code Analyzer scan, scan results are written to `scan.fpr` in the workspace.

     Note: If this file already exists, it will be overwritten.
   - If you are not running a Fortify Static Code Analyzer scan and you are uploading results to Fortify Software Security Center, the Fortify Jenkins Plugin searches "./**/*.fpr" in the workspace for the most recently modified FPR file .

7. (Optional) In the **Maximum heap memory** box, specify the maximum heap memory (in megabytes) as an integer only.

   For example, to specify 48 GB, type `49152`. By default, Fortify Static Code Analyzer automatically allocates memory based on the physical memory available on the system. If you specify the amount of memory in this box, it overrides the default automatic memory allocation.
8. (Optional) In the **Additional JVM options** box, you can add JVM commands.

   Note: Enclose each option and parameter in double quotes in boxes where you can specify multiple values.

   For example: `"-build-label" "label" "-disable-source-bundling"`
9. From the **Application type** list, select the type of application you want to analyze.

   Note: The Fortify Jenkins Plugin looks for the location of the Gradle, Maven, devenv, and MSBuild executables in the following order: Jenkins Global Tool Configuration (Gradle and Maven only), Jenkins environment variable, the PATH system environment variable, and lastly the build's workspace.

Application Type	Description
.NET Devenv	a. In the **Solution or project file** box, type the solution or project file name (or the path to the file). b. (Optional) Specify any additional devenv options.
.NET MSBuild	a. In the **Solution or project file** box, type the solution or project file name (or the path to the file). b. (Optional) Specify any additional MSBuild options.
.NET source code scan	a. In the **.NET framework version** box, specify the .NET framework version used to compile the code. b. (Optional) In the **Libdirs** box, specify a semicolon-separated list of directories where referenced system or third-party DLLs are located. c. (Optional) In the **Fortify SCA translation options** box, specify any additional Fortify Static Code Analyzer translation options. See the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation for detailed information about the available translation options. d. In the **Source files** box, specify the source files to translate.

Application Type	Description
Gradle	a. Select whether to use a specific version of Gradle installed on the agent or the Gradle Wrapper. To use a version specific version of Gradle installed on the agent: i. Select **Invoke Gradle**. ii. From the **Gradle version** list, select a version. **Note:** The **Gradle version** option is only available if at least one Gradle version is configured with the Jenkins Global Tool Configuration. Make sure that you select a Gradle version that Fortify Static Code Analyzer supports. For supported Gradle versions, see the *Fortify Software System Requirements* document for your version of Fortify Static Code Analyzer in Fortify Static Code Analyzer and Tools Documentation. To use the Gradle executable defined by the GRADLE_HOME environment variable (see "Configuring Global Settings for the Fortify Jenkins Plugin" on page 14), select **(Default)**. b. In the **Gradle tasks** box, type the Gradle tasks required for your project. c. (Optional) In the **Gradle options** box, type the Gradle options required for your project.
Java	Specify the Java source version, source path, class path, the source files, and any additional Fortify Static Code Analyzer translation options. The only required field is **Source files**. See the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation for more detailed information about the Java translation options.
Maven	a. If you have at least one Maven version configured with the Jenkins Global Tool Configuration, you can select a version from the **Maven version** list. Select **(Default)** to use the Maven executable defined by the MAVEN_HOME environment variable (see "Configuring Global Settings for the Fortify Jenkins Plugin" on page 14). **Note:** The **Maven version** option is only available if at least one

Application Type	Description
	Maven version is configured with the Jenkins Global Tool Configuration. Make sure that you select a Maven version that Fortify Static Code Analyzer supports. For supported Maven versions, see the *Fortify Software System Requirements* document for your version of Fortify Static Code Analyzer in Fortify Static Code Analyzer and Tools Documentation. b. If you did not run the build previously, then in the **Maven options** box, type `package`. Otherwise, leave this box empty. **Note:** The translation log is in the /target directory that is created when the "package" runs from Maven. Any log file location specified in the Fortify Jenkins Plugin is ignored when the Fortify Maven Plugin performs the translation.
Other	a. (Optional) Provide all the Fortify Static Code Analyzer translation options in the **Fortify SCA translation options** box. See the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation for detailed information about the available translation options. b. Specify the source code to scan in the **Includes list** box.
Advanced	Select **Advanced** if you are familiar with the Fortify Static Code Analyzer command-line interface or want to specify all the translation options without any guidance. Specify all the Fortify Static Code Analyzer translation options including source files. For detailed information about the translation options, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation.

10. (Optional) To exclude files or directories from the translation, add them to the **Exclude list** box.
11. (Optional) Enable the debug or verbose logging options.
12. (Optional) To specify a custom location for the Fortify Static Code Analyzer log file, type a file name (or a full path) in the **Log file location** box.

    By default, the log file is written to the workspace in `/.fortify/sca<version>/log`.

13. To run a scan, select the **Run Fortify SCA scan** check box, and then specify the scan settings:
    a. (Optional) In the **Custom Rulepacks** box, specify custom rules.

       Specify custom rules files (`*.xml`) separated by spaces or a directory that contains custom rules.
    b. (Optional) Specify any additional scan options.

       For detailed information about the scan options, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation.

       Note: Enclose each option and parameter in double quotes.

       In the following example, two analyzers and quick scan mode are enabled for the scan: `"-analyzers" "controlflow,dataflow" "-quick"`.
    c. (Optional) Enable the debug or verbose logging options.
    d. (Optional) To specify a custom location for the Fortify Static Code Analyzer log file, type a file name (or a full path) in the **Log file location** box.

       By default, the log file is written to the workspace in `/.fortify/sca<version>/log`.
14. To upload the scan results to Fortify Software Security Center, select the **Upload Fortify SCA scan results to Fortify Software Security Center** check box, and then specify the upload settings:
    a. Specify an application name and an application version.

       Always specify both application name and application version. If you have a successful connection to a Fortify Software Security Center server, you can search for an existing application version. To search for an application name or version, type any part of the name or version in the box, and then click **Search**. You can also leave the name or version box empty, and then click **Search**. This provides a list of all application names or versions within the configured limit. This search is case-insensitive.

       Note: The number of application names and application versions displayed in both these lists is limited by the maximum application versions per list value specified in the Fortify Jenkins Plugin configuration.

       You can type an application name and version that does not exist in Fortify Software Security Center. Fortify Jenkins Plugin will create it upon a successful build.
    b. (Optional) Specify a filter set to use when retrieving scan results for display in Jenkins. If left blank, the Fortify Jenkins Plugin uses the default filter set configured in Fortify Software Security Center.

       Note: If you specify an application version in the previous step that does not yet exist in Fortify Software Security Center, then the **Filter set** list will be empty. You can configure a filter set on the next job run.

       The failure criteria and the Normalized Vulnerability Score (NVS) calculation depend on the issues filtered by the filter set. For example, if a Quick View filter is applied to the project issues (and no critical or high issues are found), then the failure criteria determines that there

is no reason to set this build to unstable and the NVS is set to zero. The graph summary also shows zero.

c. (Optional) To trigger a build status of unstable based on the scan results, type a search query in the **Build failure criteria** box.

   For example, the following search query causes the build to fail if any critical issues exist in the scan results:

   ```
   [fortify priority order]:critical
   ```

   See the *OpenText™ Fortify Software Security Center User Guide* in Fortify Software Security Center Documentation for a description of the search query syntax.

d. (Optional) To specify the length of time to poll Fortify Software Security Center to determine if FPR processing is finished, type the time (in minutes) in the **Timeout** box.

   If no value or a value of `0` is specified, polling continues until FPR processing finishes or stops due to errors. The valid values are 0–10080.

e. (Optional) To specify the frequency that the Fortify Jenkins Plugin polls Fortify Software Security Center to check FPR processing is finished, do the following:

   i. Click **Advanced settings**.

   ii. In the **Polling interval** box, specify an interval (in minutes).

   The valid values are 1–60 and the default value is 1 minute.

   Note: The Fortify Jenkins Plugin will poll until the processing is complete or the amount of time specified for **Timeout** is reached. The **Polling Interval** must be less than the **Timeout** value.

   The Fortify Jenkins Plugin runs the NVS calculation after the FPR is processed.

   Important! If the FPR processing requires approval, then this step will not complete until the approval is granted through Fortify Software Security Center.

15. Click **Save**.

Creating a Post-Build Action to Upload Scan Results to Fortify Software Security Center

To configure a post-build action that only uploads existing scan results to Fortify Software Security Center:

1. From Jenkins, select an existing job to view or create a new job.

   If you selected an existing job, click **Configure** on the job page.

2. In the **Post-build Actions** section, click **Add post-build action**, and then select **Fortify Assessment**.
3. Select **Upload existing Fortify SCA scan results to Fortify Software Security Center**.

4. (Optional) In the **Results file** box, type the name of the Fortify Project Results (FPR) file that you want to upload.

   You can specify a file name or an absolute path. If a file is not specified, the Fortify Jenkins Plugin searches "./**/*.fpr" in the build's workspace for the most recently modified FPR file.

   Note: You can also upload third-party artifacts in a ZIP file. For detailed instructions about preparing the ZIP file, see the *OpenText™ Fortify Software Security Center User Guide* in Fortify Software Security Center Documentation for a how to upload scan artifacts.

5. Specify an application name and an application version.

   Always specify both application name and application version. If you have a successful connection to a Fortify Software Security Center server, you can search for an existing application version. To search for an application name or version, type any part of the name or version in the box, and then click **Search**. You can also leave the name or version box empty, and then click **Search**. This provides a list of all application names or versions within the configured limit. This search is case-insensitive.

   Note: The number of application names and application versions displayed in both these lists is limited by the maximum application versions per list value specified in the Fortify Jenkins Plugin configuration.

   You can type an application name and version that does not exist in Fortify Software Security Center. Fortify Jenkins Plugin will create it upon a successful build.

6. (Optional) Specify a filter set to use when retrieving scan results for display in Jenkins. If left blank, the Fortify Jenkins Plugin uses the default filter set configured in Fortify Software Security Center.

   Note: If you specify an application version in the previous step that does not yet exist in Fortify Software Security Center, then the **Filter set** list will be empty. You can configure a filter set on the next job run.

   The failure criteria and the Normalized Vulnerability Score (NVS) calculation depend on the issues filtered by the filter set. For example, if a Quick View filter is applied to the project issues (and no critical or high issues are found), then the failure criteria determines that there is no reason to set this build to unstable and the NVS is set to zero. The graph summary also shows zero.

7. (Optional) To trigger a build status of unstable based on the scan results, type a search query in the **Build failure criteria** box.

   For example, the following search query causes the build to fail if any critical issues exist in the scan results:

   ```
   [fortify priority order]:critical
   ```

   See the *OpenText™ Fortify Software Security Center User Guide* in Fortify Software Security Center Documentation for a description of the search query syntax.

8. (Optional) To specify the length of time to poll Fortify Software Security Center to determine if FPR processing is finished, type the time (in minutes) in the **Timeout** box.

   If no value or a value of `0` is specified, polling continues until FPR processing finishes or stops due to errors. The valid values are 0–10080.
9. (Optional) To specify the frequency that the Fortify Jenkins Plugin polls Fortify Software Security Center to check FPR processing is finished, do the following:
   a. Click **Advanced settings**.
   b. In the **Polling interval** box, specify an interval (in minutes).

      The valid values are 1–60 and the default value is 1 minute.

   Note: The Fortify Jenkins Plugin will poll until the processing is complete or the amount of time specified for **Timeout** is reached. The **Polling Interval** must be less than the **Timeout** value.

   The Fortify Jenkins Plugin runs the NVS calculation after the FPR is processed.

   Important! If the FPR processing requires approval, then this step will not complete until the approval is granted through Fortify Software Security Center.
10. Click **Save**.

Chapter 4: Configuring Fortify Analysis with Pipeline Jobs

The Fortify Jenkins Plugin supports both Declarative and Scripted Pipeline syntax. The advantage of using Jenkins Pipeline is that you can check your script into source control, and you can have multiple Fortify Static Code Analyzer translation or upload requests (for example) within the same Jenkinsfile script. See the Jenkins documentation for additional information about pipelines.

This section contains the following topics:

Pipeline Steps to Translate and Scan Remotely

There are two Pipeline steps available to perform the analysis remotely. The following table lists these Fortify Jenkins Plugin Pipeline steps. Each section describes the parameters and contains examples.

Task	Pipeline Step
Set options for remote translation and scan. This step is optional and if used should precede a fortifyRemoteAnalysis step.	"fortifyRemoteArguments Step" on page 40
Send a project to a remote system for analysis.	"fortifyRemoteAnalysis Step" on page 41

The following is an example Jenkinsfile that sends a Java project that uses Gradle to a remote system for analysis. After the remote analysis is complete, the Controller uploads the scan results to Fortify Software Security Center.

```
node {
  stage('Fortify Remote Arguments') {
    fortifyRemoteArguments transOptions: '-Xmx4G',
      scanOptions: '"-analyzers" "dataflow"'
  }
  stage('Fortify Remote Analysis') {
    fortifyRemoteAnalysis remoteAnalysisProjectType: fortifyGradle(),
      remoteOptionalConfig: [notifyEmail: 'joe@xyzCo.com',
      customRulepacks: 'MyRules.xml'],
      uploadSSC: [appName: 'MyJavaApp', appVersion: '3.1']
  }
}
```

The following Declarative Pipeline script has the same functionality as the previous example:

```
pipeline {
  agent any
  stages {
   stage('Fortify Remote Arguments') {
      steps {
        fortifyRemoteArguments transOptions: '-Xmx4G',
          scanOptions: '"-analyzers" "dataflow"'
      }
    }
    stage('Fortify Remote Analysis') {
      steps {
        fortifyRemoteAnalysis remoteAnalysisProjectType: fortifyGradle(),
          remoteOptionalConfig: [notifyEmail: 'joe@xyzCo.com',
          customRulepacks: 'MyRules.xml'],
          uploadSSC: [appName: 'MyJavaApp', appVersion: '3.1']
      }
    }
  }
}
```

fortifyRemoteArguments Step

Use this step to specify Fortify Static Code Analyzer translation and scan options in a settings file for remote analysis. This step is optional. To start a remote analysis, follow this step with a fortifyRemoteAnalysis step.

Parameter	Description
`transOptions`	Optional (String). Specifies any additional Fortify Static Code Analyzer translation options. Enclose each option and parameter in double quotes.
`scanOptions`	Optional (String). Specifies any additional Fortify Static Code Analyzer scan options. Enclose each option and parameter in double quotes.

See Also

"fortifyRemoteArguments Example" below

"fortifyRemoteAnalysis Step" on the next page

fortifyRemoteArguments Example

The following example specifies 4 GB for the translation and excludes SQL. Only Control Flow and Dataflow analyzers are used and the default Rulepacks are not processed for the scan phase.

```
node {
  stage('Fortify Remote Arguments') {
    fortifyRemoteArguments transOptions: '"-Xmx4G"
      "-disable-language" "sql"',
       scanOptions: '"-analyzers" "controlflow,dataflow"
       "-no-default-rules"'
  }
}
```

See Also

"fortifyRemoteArguments Step" above

fortifyRemoteAnalysis Step

Use this step to send a project to a remote system for analysis (translation and scan). To add additional translation or scan options for the analysis, precede this step with the fortifyRemoteArguments step.

Parameter	Description	Default Value
`RemoteAnalysisProjectType`	Required (String). The project type is one of the following: `fortifyGradle`, `fortifyMaven`, `fortifyMSBuild`, `fortifyPHP`, `fortifyPython`, or `fortifyOther`.	(none)
Gradle Parameters		
`buildFile`	Optional (String). Specifies the build file name.	`build.gradle`
`includeTests`	Optional (boolean). Specifies whether to include a test source set.	`false`
`skipBuild`	Optional (boolean). Specifies whether to skip the build invocation used to obtain all project dependencies. Use this option if you have a build step earlier in the pipeline and do not want to run the build again. **Note:** This setting is only valid for Fortify ScanCentral SAST versions 20.2.0 or later.	`false`
Maven Parameters		

Parameter	Description	Default Value
`buildFile`	Optional (String). Specifies the build file name.	`pom.xml`
`includeTests`	Optional (boolean). Specifies whether to include a test scope.	`false`
`skipBuild`	Optional (boolean). Specifies whether to skip the build invocation used to obtain all project dependencies. Use this option if you have a build step earlier in the pipeline and do not want to run the build again. **Note:** This setting is only valid for Fortify ScanCentral SAST versions 20.2.0 or later.	`false`
MSBuild Parameters		
`dotnetProject`	Required (String). Specifies the project or solution file to analyze.	(none)
`excludeDisabledProjects`	Optional (boolean). Specifies whether to exclude disabled projects in the solution from the analysis. **Note:** This parameter is only valid for Fortify ScanCentral SAST versions 21.1.x or earlier.	`false`
PHP Parameters		

Parameter	Description	Default Value
`phpVersion`	Optional (Number). Specifies the PHP version used in the project.	The default version is defined by Fortify Static Code Analyzer. For example, in Fortify Static Code Analyzer version 23.1.0, the default PHP version is 7.4. See the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation for specific version information.
Python Parameters		
`pythonVersion`	Optional (String). Specifies the Python version used in the project. The valid values are 2 and 3. This parameter is ignored if you also provide the `pythonVirtualEnv` parameter.	`2`
`pythonVirtualEnv`	Optional (String). Specifies the location (directory) of the Python virtual environment.	(none)
`pythonRequirementsFile`	Optional (String). Specifies the Python project requirements file used to install and collect dependencies.	(none)
remoteOptionalConfig Parameters		
`sensorPoolUUID`	Optional (String). Specifies which sensor pool to submit the job.	(none)
`notifyEmail`	Optional (String). Specifies	(none)

Parameter	Description	Default Value
	the email address to which the Controller will send notifications.	
`customRulepacks`	Optional (String). Specify custom rules files (`*.xml`) separated by spaces or a directory that contains custom rules.	(none)
`filterFile`	Optional (String). Specifies a file used to filter out specific vulnerability categories, rules, and vulnerability instances from the analysis. For more information about filter files, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation.	(none)
uploadSSC Parameters		
`appName`	Required (String). Specifies an existing application name for which to store the results in Fortify Software Security Center.	(none)
`appVersion`	Required (String). Specifies an existing application version for which to store the results in Fortify Software Security Center.	(none)

See Also

"fortifyRemoteAnalysis Example" on the next page

"fortifyRemoteArguments Step" on page 40

fortifyRemoteAnalysis Example

Specify a function name for the `RemoteAnalysisProjectType` parameter. The valid function names are: `fortifyGradle`, `fortifyMaven`, `fortifyMSBuild`, `fortifyPHP`, `fortifyPython`, and `fortifyOther`.

The following example uploads a Python 3 project to a remote system for translation and scan. Controller notifications are emailed to joe@xyzCo.com. After the analysis is complete, the Fortify Jenkins Plugin uploads the project to Fortify Software Security Center.

```
node {
  stage('Get Src Code') {
     git credentialsId: '3e58c50d-cd4a-6e28-ff44-cb164dec13f2',
       url: 'https://github.xyzCo.com/MyDept/projA.git'
   }

  stage('Fortify Remote Analysis') {
    fortifyRemoteAnalysis
      remoteAnalysisProjectType: fortifyPython: (pythonVersion: '3',
        pythonRequirementsFile: 'C:\\projA\\requirements.txt',
        pythonVirtualEnv: 'C:\\projA\\my_project'),
      remoteOptionalConfig: [notifyEmail: 'joe@xyzCo.com'],
      uploadSSC: [appName: 'ProjA', appVersion: '2.3Beta']
  }
}
```

See Also

"fortifyRemoteAnalysis Step" on page 41

Pipeline Steps to Translate Locally and Scan Remotely

The following table lists the available Fortify Jenkins Plugin Pipeline steps to perform local translation, remote scan, and upload to Fortify Software Security Center. Each section describes the parameters and contains examples.

Project Build Step	Pipeline Step
Run a local Fortify Static Code Analyzer clean	"fortifyClean Step" on page 53
Run a local Fortify Static Code Analyzer translation	"fortifyTranslate Step" on page 54
Run a Remote Fortify Static Code Analyzer scan	"fortifyRemoteScan Step" on page 47

Note: If any Fortify Jenkins Plugin Pipeline step in a script fails to execute, then the build fails. You do have the option to implement your own exception-catch mechanism to ignore a step failure.

The following is an example Jenkinsfile that performs the Fortify Static Code Analyzer translation for a Java project on the local system, uploads the project to a remote system for scanning, and then uploads the scan results to Fortify Software Security Center:

```
node {
  stage('Fortify Clean') {
    fortifyClean buildID: 'MyJavaApp', logFile: 'MyJavaAppFortify.log'
  }
  stage('Fortify Translate') {
    fortifyTranslate buildID: 'MyJavaApp',
      logFile: 'MyJavaApp-translate.log',
      projectScanType: fortifyJava(javaSrcFiles:
       'src\\main\\java\\com\\projectA',
      javaVersion: '17')
  }
  stage('Remote Fortify Scan Upload to SSC') {
    fortifyRemoteScan buildID: 'MyJavaApp',
      remoteOptionalConfig: [notifyEmail: 'joe@xyzCo.com',
        scanOptions:"-analyzers" "controlflow"],
        uploadSSC: [appName: 'JavaAppA', appVersion: '3']
  }
}
```

The following Declarative Pipeline script has the same function as the previous example:

```
pipeline {
  agent any
  stages {
    stage('Fortify Clean') {
      steps {
        fortifyClean buildID: 'MyJavaApp',
          logFile: 'MyJavaAppFortify.log'
       }
     }
    stage('Fortify Translate') {
      steps {
        fortifyTranslate buildID: 'MyJavaApp',
          logFile: 'MyJavaApp-translate.log',
          projectScanType: fortifyJava(javaSrcFiles:
          'src\\main\\java\\com\\projectA', javaVersion: '17')
      }
    }
    stage('Remote Fortify Scan Upload to SSC') {
      steps {
        fortifyRemoteScan buildID: 'MyJavaApp',
          remoteOptionalConfig: [notifyEmail: 'joe@xyzCo.com',
            scanOptions:"-analyzers" "controlflow"],
          uploadSSC: [appName: 'JavaAppA', appVersion: '3']
       }
    }
  }
}
```

fortifyRemoteScan Step

Use this step to send a locally translated project to a remote system for the scan phase.

Parameter	Description
`buildID`	Required (String). A unique identifier for the analysis.
remoteOptionalConfig Parameters	
`sensorPoolUUID`	Optional (String). Targets a specific sensor pool for the scan request.
`notifyEmail`	Optional (String). Specifies the email address to which the Controller will send notifications.
`scanOptions`	Optional (String). Fortify Static Code Analyzer scan options. For descriptions of the available scan options, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation.

Parameter	Description
	Note: Enclose each option and parameter in double quotes. In the following example, two analyzers and quick scan mode are enabled for the scan: `"-analyzers" "controlflow,dataflow" "-quick"`.
`customRulepacks`	Optional (String). Specify custom rules files (*.xml) separated by spaces or a directory that contains custom rules.
`filterFile`	Optional (String). Specifies a file used to filter out specific vulnerability categories, rules, and vulnerability instances from the analysis. For more information about filter files, see the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation.
uploadSSC Parameters	
`appName`	Required (String). Specifies an existing application name for which to store the results in Fortify Software Security Center.
`appVersion`	Required (String). Specifies an existing application version for which to store the results in Fortify Software Security Center.

See Also

"fortifyRemoteScan Example" below

fortifyRemoteScan Example

The following example uploads a locally translated project with a build ID of MyAppA to the remote system for scanning. After the scan is complete, the Fortify Jenkins Plugin uploads the project to Fortify Software Security Center.

```
node {
  stage('Remote Fortify Scan Upload to SSC') {
    fortifyRemoteScan buildID: 'MyAppA',
      remoteOptionalConfig: [notifyEmail: 'joe@xyzCo.com',
      scanOptions:"-quick"],
      uploadSSC: [appName: 'AppA', appVersion: 'version1']
  }
}
```

See Also

"fortifyRemoteScan Step" on the previous page

Pipeline Steps to Translate and Scan Locally

The following table lists the available Fortify Jenkins Plugin Pipeline steps to update Fortify Software Security Content, run a local translation, run a local scan, and upload analysis results to Fortify Software Security Center. Each section describes the parameters and contains examples.

Project Build Step	Pipeline Step
Update Fortify Software Security Content to use for local translation and scan	"fortifyUpdate Step" on page 51
Run a local Fortify Static Code Analyzer clean	"fortifyClean Step" on page 53
Run a local Fortify Static Code Analyzer translation	"fortifyTranslate Step" on page 54
Run a local Fortify Static Code Analyzer scan	"fortifyScan Step" on page 60
Upload local Fortify Static Code Analyzer scan results to Fortify Software Security Center	"fortifyUpload Step" on page 61

Note: If any Fortify Jenkins Plugin Pipeline step in a script fails to execute, then the build fails. You do have the option to implement your own exception-catch mechanism to ignore a step failure.

The following is an example Jenkinsfile that updates Fortify Software Security Content, performs a complete Fortify analysis of a Java project, and then uploads the scan results to Fortify Software Security Center:

```
node {
  stage('Fortify Update') {
    fortifyUpdate  updateServerURL: 'https://update.fortify.com'
  }
  stage('Fortify Clean') {
    fortifyClean buildID: 'MyJavaApp', logFile: 'MyJavaAppFortify.log'
  }
  stage('Fortify Translate') {
    fortifyTranslate buildID: 'MyJavaApp',
      logFile: 'MyJavaApp-translate.log',
      projectScanType: fortifyJava(javaSrcFiles:
        'src\\main\\java\\com\\projectA', javaVersion: '17')
  }
  stage('Fortify Scan') {
    fortifyScan buildID: 'MyJavaApp', resultsFile: 'MyJavaApp.fpr'
      customRulepacks: 'MyRules.xml', logFile: 'MyJavaApp-scan.log'
  }
  stage('Fortify Upload') {
    fortifyUpload appName: 'JavaAppA', appVersion: '3',
      resultsFile: 'MyJavaApp.fpr'
  }
}
```

The following Declarative Pipeline script has the same function as the previous example:

```
pipeline {
  agent any
  stages {
    stage('Fortify Update') {
      steps {
        fortifyUpdate  updateServerURL: 'https://update.fortify.com'
      }
    }
    stage('Fortify Clean') {
      steps {
        fortifyClean buildID: 'MyJavaApp',
          logFile: 'MyJavaAppFortify.log'
       }
    }
    stage('Fortify Translate') {
      steps {
        fortifyTranslate buildID: 'MyJavaApp',
          logFile: 'MyJavaApp-translate.log',
          projectScanType: fortifyJava(javaSrcFiles:
          'src\\main\\java\\com\\projectA', javaVersion: '17')
      }
    }
    stage('Fortify Scan') {
      steps {
        fortifyScan buildID: 'MyJavaApp',
          resultsFile: 'MyJavaApp.fpr'
          customRulepacks: 'MyRules.xml',
          logFile: 'MyJavaApp-scan.log'
       }
    }
    stage('Fortify Upload') {
      steps {
        fortifyUpload appName: 'JavaAppA', appVersion: '3',
          resultsFile: 'MyJavaApp.fpr'
       }
    }
  }
}
```

fortifyUpdate Step

Use this step to update the local copy of the Fortify Software Security Content used by the Fortify translation and scan steps. To connect to the Fortify Rulepack update server or the Fortify Software Security Center server with a proxy, you can use the proxy settings configured in Jenkins.

Parameter	Description	Default Value
`updateServerURL`	Optional (String). Specifies the URL for the Fortify Rulepack update	`https://update.fortify.com`

Parameter	Description	Default Value
	server. You can download Fortify security content from Fortify Software Security Center by specifying a Fortify Software Security Center URL as the value for this parameter.	
`locale`	Optional (String). Specifies the locale for the Fortify Rulepack. Use one of the following locale codes: • English: `en` • Chinese (Simplified): `zh_CN` • Chinese (Traditional): `zh_TW` • Portuguese (Brazil): `pt_BR` • Korean: `ko` • Spanish: `es`	`en`
acceptKey	Optional (boolean). Specifies whether to accept the public key when updating Fortify Software Security Content from Fortify Software Security Center.	`false`

See Also

"fortifyUpdate Example" below

fortifyUpdate Example

The following example updates the Fortify Software Security Content from the Fortify Rulepack update server in Spanish:

```
node {
  stage('Fortify Update') {
    fortifyUpdate  updateServerURL: 'https://update.fortify.com', locale: 'es'
  }
}
```

See Also

"fortifyUpdate Step" on the previous page

fortifyClean Step

Use this step to remove any temporary files from a previous scan for a specific build ID.

Parameter	Description	Default Value
`buildID`	Required (String). A unique identifier for the scan.	(none)
`maxHeap`	Optional (int). The maximum heap size for the JVM (-Xmx).	By default, Fortify Static Code Analyzer automatically allocates memory based on the physical memory available on the system.
`addJVMOptions`	Optional (String). Specifies additional JVM commands.	(none)
`debug`	Optional (boolean). Specifies whether to include debug information in the Fortify Support log file.	`false`
`verbose`	Optional (boolean). Specifies whether to send verbose status messages to the console and to the Fortify Support log file.	`false`
`logFile`	Optional (String). Specifies the log file location and file name.	The default file name is `sca.log` and the default location is in the workspace directory.

See Also

"fortifyClean Example" below

fortifyClean Example

The following example removes all the temporary files for the MyJavaApp build ID:

```
node {
  stage('Fortify Clean') {
    fortifyClean buildID: 'MyJavaApp', logFile: 'MyJavaAppFortify.log'
  }
}
```

See Also

"fortifyClean Step" on the previous page

fortifyTranslate Step

Use this step to translate the project source code on the local system.

Parameter	Description	Default Value
General Parameters		
`buildID`	Required (String). A unique identifier for the analysis.	(none)
`projectScanType`	Required. (String). The project scan type is one of the following: `fortifyAdvanced`, `fortifyDevenv`, `fortifyDotnetSrc`, `fortifyGradle`, `fortifyJava`, `fortifyMaven3`, `fortifyMSBuild`, or `fortifyOther`.	(none)
`excludeList`	Optional (String). Specifies a list of directories or files to exclude from translation.	(none)
`maxHeap`	Optional (int). The maximum heap size for the JVM (-Xmx).	By default, Fortify Static Code Analyzer automatically allocates memory based on the physical memory available on the system.
`addJVMOptions`	Optional (String). Additional JVM commands.	(none)
`debug`	Optional (boolean). Specifies whether to	`false`

Parameter	Description	Default Value
	include debug information in the Fortify Support log file.	
`verbose`	Optional (boolean). Specifies whether to send verbose status messages to the console and to the Fortify Support log file.	`false`
`logFile`	Optional (String). Specifies the log file location and file name.	The default file name is `sca.log` and the default location is the workspace directory.
fortifydevenv and fortifyMSBuild Parameters		
`dotnetProject`	Required (String). Specifies a solution or a project file.	(none)
`dotnetAddOptions`	Optional (String). Specifies any additional Fortify Static Code Analyzer options for translating .NET code.	(none)
fortifyDotnetSrc Parameters		
`dotnetFrameworkVersion`	Required (int). Specifies the .NET framework version.	(none)
`dotnetSrcFiles`	Required (String). Specifies the location of the .NET source files.	(none)
`dotnetLibdirs`	Optional (String). Specifies a semicolon-separated list of directories where referenced system or	(none)

Parameter	Description	Default Value
	third-party DLLs are located.	
`dotnetAddOptions`	Optional (String). Specifies any additional devenv or MSBuild options required for your project.	(none)
fortifyMaven3 Parameters		
`mavenInstallationName`	Optional (String). Specifies the version of Maven using a name that you configured with the Jenkins Global Tool Configuration.	`(Default)` Uses the Maven executable defined by the MAVEN_HOME environment variable.
`mavenOptions`	Optional (String). Specifies any additional Maven options required for your project.	(none)
fortifyGradle Parameters		
`gradleInstallationName`	Optional (String). Specifies the version of Gradle using a name that you configured with the Jenkins Global Tool Configuration.	`(Default)` Uses the Gradle executable defined by the GRADLE_HOME environment variable.
`useWrapper`	Optional (boolean). Specifies whether to use a Wrapper.	`false`
`gradleTasks`	Required (String). Specifies the Gradle tasks required for your project.	(none)

Parameter	Description	Default Value
`gradleOptions`	Optional (String). Specifies any additional Gradle options required for your project.	(none)
fortifyJava Parameters		
`javaSrcFiles`	Required (String). Specifies the location of the Java source files.	(none)
`javaVersion`	Optional (String). Specifies the JDK version for which the Java code is written.	The default version is defined by Fortify Static Code Analyzer. For example, in Fortify Static Code Analyzer version 23.2.0, the default JDK version is 11. See the *OpenText™ Fortify Static Code Analyzer User Guide* in Fortify Static Code Analyzer and Tools Documentation for specific version information.
`javaClasspath`	Optional (String). Specifies the class path as a colon- or semicolon-separated list of directories to use for analyzing Java source code.	(none)
`javaAddOptions`	Optional (String). Specifies any additional Fortify Static Code Analyzer options for translating Java code.	(none)
fortifyOther Parameters		
`otherIncludesList`	Required (String). Specifies the location of the source files.	(none)

Parameter	Description	Default Value
`otherOptions`	Optional (String). Specifies any additional Fortify Static Code Analyzer options required for your project.	(none)
fortifyAdvanced Parameters		
`advOptions`	Required (String). Specifies all the Fortify Static Code Analyzer options that are necessary to translate the project.	(none)

See Also

"fortifyTranslate Examples" below

fortifyTranslate Examples

Specify a function name for the `projectScanType` parameter. The valid function names are: `fortifyAdvanced()`, `fortifyDevenv()`, `fortifyDotnetSrc()`, `fortifyGradle()`, `fortifyJava()`, `fortifyMaven3()`, `fortifyMSBuild()`, and `fortifyOther()`.

The following example translates a Java project and excludes some files from the translation:

```
node {
  stage('Fortify Translate') {
    fortifyTranslate buildID: 'MyJavaApp',
    excludeList: '"src\\main\\java\\com\\projectA\\command\\Config.java"
        "src\\main\\java\\com\\projectA\\command\\Test*.java"',
    logFile: 'MyJavaApp-translate.log',
    projectScanType: fortifyJava(javaSrcFiles:
     'src\\main\\java\\com\\projectA',javaVersion: '17')
  }
}
```

The following example uses Maven to translate a Java project:

```
node {
  stage('Fortify Translate') {
    fortifyTranslate buildID: 'MyJavaApp',
```

```
    excludeList: '"src\\main\\java\\com\\projectA\\command\\Config.java"
        "src\\main\\java\\com\\projectA\\command\\Test*.java"',
    logFile: 'MyJavaApp.log', maxHeap: '4800',
    projectScanType: fortifyMaven3(mavenOptions: 'package')
  }
}
```

The following example uses MSBuild to translate a .NET solution:

```
node {
  stage('Fortify Translate') {
    fortifyTranslate buildID: 'MyDotNetApp', ,
    logFile: 'MyJavaApp.log', maxHeap: '4800',
    projectScanType: fortifyMSBuild(dotnetProject: 'MyDotNetApp.sln',
      dotnetAddOptions: '/t:rebuild')
  }
}
```

The following example translates a Python 3 project:

```
node {
  stage('Fortify Translate') {
    fortifyTranslate buildID: 'MyPythonApp',
    excludeList: '"src\\**\\Test*.py"',
    logFile: 'MyPythonApp-translate.log',
    projectScanType: fortifyAdvanced(advOptions: '"-python-version" "3"
      "-python-path" "C:\\Python33\\lib\\site-packages"
      "src\\main\\pythonApp" ')
  }
}
```

The following example translates a JavaScript application:

```
node {
  stage ('Fortify Translate') {
    fortifyTranslate buildID: 'JS_App',
    logFile: 'JS_App-translate.log', projectScanType:
    fortifyOther(otherIncludesList: ‘./**/*.js’)
  }
}
```

See Also

"fortifyTranslate Step" on page 54

fortifyScan Step

Use this step to run a scan on all the translated files with the specific build ID.

Parameter	Description	Default Value
`buildID`	Required (String). A unique identifier for the scan.	(none)
`maxHeap`	Optional (number). The maximum heap size for the JVM (-Xmx).	By default, Fortify Static Code Analyzer automatically allocates memory based on the physical memory available on the system.
`addJVMOptions`	Optional (String). Specifies additional JVM commands.	(none)
`resultsFile`	Optional (String). Specifies a name for the Fortify results file (FPR). For example, `MyProjectA.fpr`.	`scan.fpr`
`customRulepacks`	Optional (String). Specifies custom rules (XML files).	(none)
`addOptions`	Optional (String). Specifies any additional scan options. Enclose each option and parameter in double quotes.	(none)
`debug`	Optional (boolean). Specifies whether to include debug information in the Fortify Support log file.	false
`verbose`	Optional (boolean). Specifies whether to send verbose status messages to the console and to the Fortify Support log file.	false
`logFile`	Optional (String). Specifies the log file location and file name.	The default file name is `sca.log` and the default location is the workspace directory.

See Also

"fortifyScan Example" below

fortifyScan Example

The following example scans the previously-translated project with the MyJavaApp build ID:

```
node {
  stage('Fortify Scan') {
    fortifyScan buildID: 'MyJavaApp', resultsFile: 'MyJavaApp.fpr'
    customRulepacks: 'MyRules.xml', logFile: 'MyJavaApp-scan.log'
  }
}
```

See Also

"fortifyScan Step" on the previous page

fortifyUpload Step

Use this step to upload the scan results (FPR) to Fortify Software Security Center. The information to connect to Fortify Software Security Center is obtained from the **Fortify Assessment** section in the Jenkins global settings (see "Configuring Global Settings for the Fortify Jenkins Plugin" on page 14). After the upload is complete, you can view the results in Jenkins (see "Viewing Scan Results" on page 65). To connect to the Fortify Software Security Center server with a proxy, you can use the proxy settings configured in Jenkins.

Parameter	Description	Default Value
`appName`	Required (String). Specifies the application name for which to store the results in Fortify Software Security Center.	(none)
`appVersion`	Required (String). Specifies the application version for which to store the results in Fortify Software Security Center.	(none)
`resultsFile`	Optional (String). Specifies a name for the FPR file. For example, `MyProjectA.fpr`. **Note:** You can also upload third-party artifacts in a ZIP file. For detailed instructions about preparing the ZIP file, see the *OpenText™ Fortify Software*	If you ran a Fortify Static Code Analyzer scan, the default file is `scan.fpr`, otherwise the Fortify Jenkins Plugin searches "./**/*.fpr" in the workspace for the FPR

Parameter	Description	Default Value
	Security Center User Guide in Fortify Software Security Center Documentation for a how to upload scan artifacts.	file with the latest modified date.
`filterSet`	Optional (String). Specifies the ID of a filter set to use when retrieving scan results for display in Jenkins. The filter set ID for Quick View is `32142c2d-3f7f-4863-a1bf-9b1e2f34d2ed` and the filter set ID for Security Auditor View is `a243b195-0a59-3f8b-1403-d55b7a7d78e6`.	The default filter set configured in Fortify Software Security Center.
`failureCriteria`	Optional (String). Specifies a search query to use on the scan results to trigger a build status of unstable. For example, `[fortify priority order]:critical`. The fortifyUpload step returns the number of issues that satisfy the failure criteria. A return value of zero indicates no matches were found and the build status is not marked as unstable.	(none)
`timeout`	Optional (int). Specifies the time (in minutes) to poll Fortify Software Security Center to check if FPR processing is finished. If no value or a value of 0 is specified, polling continues until FPR processing finishes or stops due to an error. The valid values are 0–10080.	(none)

Parameter	Description	Default Value
`pollingInterval`	Optional (int). Specifies the interval (in minutes) at which the Fortify Jenkins Plugin polls Fortify Software Security Center to check if FPR processing is finished. The valid values are 0–60. **Note:** The Fortify Jenkins Plugin will poll until the processing is complete or the amount of time specified for `timeout` is reached. The `pollingInterval` must be less than the `timeout` value. **Important!** If the FPR processing requires approval, then this step will not complete until the approval is granted through Fortify Software Security Center.	`1`

See Also

"fortifyUpload Examples" below

fortifyUpload Examples

The following example uploads the Fortify scan results for the MyJavaApp project to version 3 of the MyJavaApp application on Fortify Software Security Center. The upload will abort if not completed within 15 minutes. The Fortify Jenkins Plugin will poll SSC every minute (default) to determine if the FPR processing is complete.

```
node {
  stage('Fortify Upload') {
    fortifyUpload appName: 'MyJavaApp', appVersion: '3',
    resultsFile: 'MyJavaApp.fpr', timeout: '15'
  }
}
```

The following example uploads the Fortify scan results to version 1.2 of the MyJavaCode application on Fortify Software Security Center. The pipeline script reports if there are any issues in the scan results with a critical Fortify Priority Order.

```
node {
  stage('ReportCriticals') {
    steps {
      script {
        def criticalCount = fortifyUpload(appName: 'MyJavaCode', appVersion: '1.2',
          failureCriteria: '[fortify priority order]:Critical')
        if (criticalCount > 0) {
          echo "Detected ${criticalCount} critical vulnerabilities"
        }
      }
    }
  }
```

See Also

"fortifyUpload Step" on page 61

Chapter 5: Viewing Scan Results

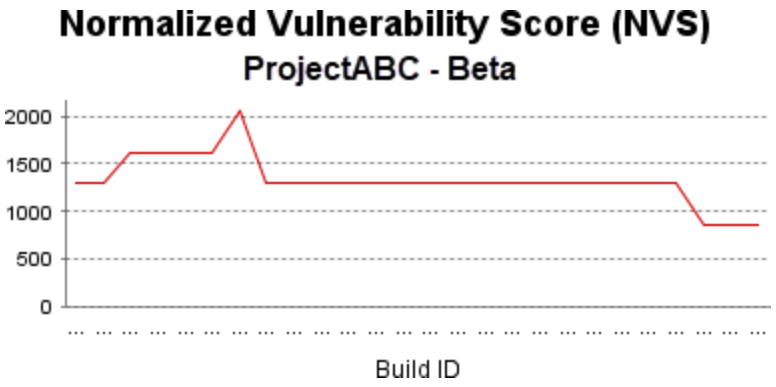
When you perform the Fortify analysis on the local system and if you uploaded Fortify Static Code Analyzer results to Fortify Software Security Center, you can view a security vulnerability graph for your project and a summary of the issues from Jenkins.

Note: When an analysis is performed on a remote system and you uploaded the Fortify Static Code Analyzer results to Fortify Software Security Center, you can view the results in Fortify Software Security Center.

This section contains the following topics:

Security Vulnerability Graph for your Project

The project page displays a Normalized Vulnerability Score (NVS) graph. The NVS is a normalized score that gives you an estimate of the security vulnerability of your project.



The Fortify Jenkins Plugin calculates the NVS with the following formula:

```
NVS = ((CFPO * 10) + (HFPO * 5) + (MFPO * 1) + (LFPO * 0.1)) * 0.5 +
      ((P1 * 2) + (P2 * 4) + (P3 * 16) + (PABOVE *64)) * 0.5
```

where:

- CFPO = Number of critical vulnerabilities (unless audited as Not an Issue)
- HFPO = Number of high vulnerabilities (unless audited as Not an Issue)
- MFPO = Number of medium vulnerabilities (unless audited as Not an Issue)
- LFPO = Number of low vulnerabilities (unless audited as Not an Issue)

and:

- PABOVE = Exploitable
- P3 = Suspicious
- P2 = Bad practice
- P1 = Reliability issue

The total issues count is not especially useful. For example, if Application A has no critical issues and ten low issues, the total issue count is ten. If Application B has five critical issues and no low issues, the total issue count is five. These values might mislead you to think that Application B is less vulnerable than Application A, when it is not.

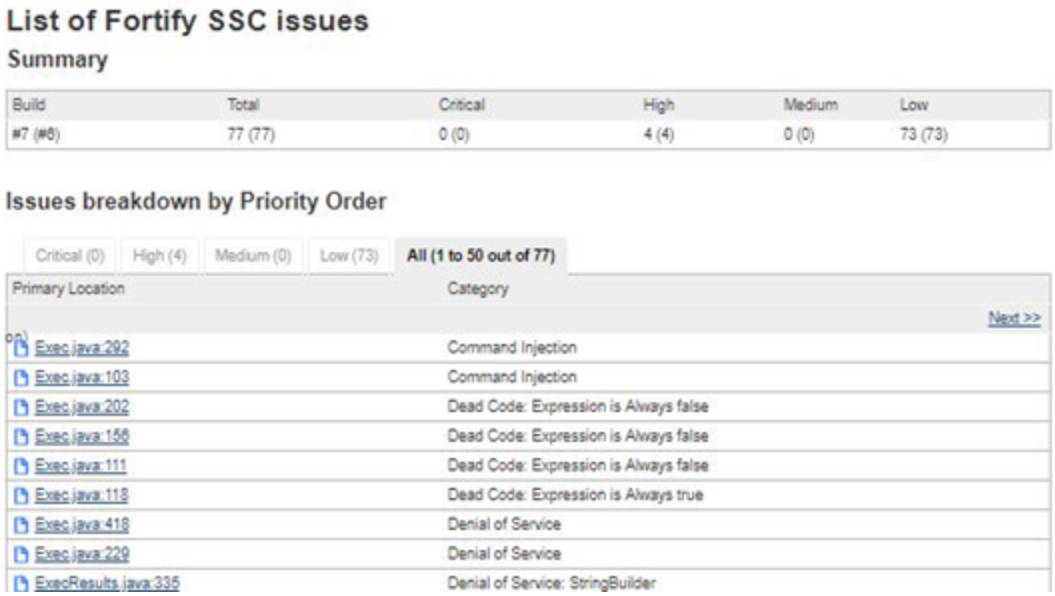
The NVS calculated for the two example applications provides a different picture (simplified equation):

- Application A: NVS = 0*10 + 10*0.1 = 1
- Application B: NVS = 5*10 + 0*0.1 = 50

Viewing Issues

To see the issues for a Fortify Static Code Analyzer analysis that you have uploaded to Fortify Software Security Center, open your job in Jenkins and click **Fortify Assessment** on the left.

The interactive **List of Fortify SSC issues** page displays the **Summary** and **Issues breakdown by Priority Order** tables.



The **Summary** table shows the difference in the number of issues in different categories between the two most recent builds. A blue arrow next to a value indicates that the number in that category has decreased, and a red arrow indicates that the number in that category has increased.

The **Issues breakdown by Priority Order** table shows detailed information about the issues for the specified location and category in each priority folder. Wait for the table to load. If the data load takes longer than expected, you might need to refresh the browser window.

By default, you see the critical issues first. To see all issues, select the **All** tab.

Note: The more issues a page shows, the longer it takes to load. Fortify recommends that you not use the **All** tab for large projects.

The first and the second columns show the file name and line number of the issue and the full path to this file. The last column displays the category of each vulnerability.

By default, issues are sorted by primary location. To organize them by category, click the **Category** column header.

To see more details about or to audit a specific issue, click the file name in the first column. The link takes you directly to the details for that issue on the Fortify Software Security Center server. If you are not logged in to Fortify Software Security Center, you are prompted to log in.

Configuring the Number of Issues Displayed on a Page

By default, the page displays up to 50 issues. To navigate to all the issues, use **Next>>** and **<<Previous** on the top and bottom of the table. To increase the maximum number of issues displayed to 100 per page, from the **50 | 100 | All** section at the bottom of the page, click **100**.

To control the number of the issues shown on a page from the **Configure System** page:

- In the **Fortify Assessment** section, change the value in the **Maximum issues per page** box.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email.

Note: If you are experiencing a technical issue with our product, do not email the documentation team. Instead, contact Customer Support at https://www.microfocus.com/support so they can assist you.

If an email client is configured on this computer, click the link above to contact the documentation team and an email window opens with the following information in the subject line:

Feedback on User Guide (Fortify Jenkins Plugin 23.1)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to fortifydocteam@opentext.com.

We appreciate your feedback!