

---

# Micro Focus Fortify Jenkins Plugin

Software Version: 20.2.0

## User Guide

Document Release Date: Revision 1: December 2020

Software Release Date: November 2020



## Legal Notices

Micro Focus  
The Lawn  
22-30 Old Bath Road  
Newbury, Berkshire RG14 1QN  
UK

<https://www.microfocus.com>

## Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

## Restricted Rights Legend

Confidential computer software. Except as specifically indicated otherwise, a valid license from Micro Focus is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

## Copyright Notice

© Copyright 2014 - 2020 Micro Focus or one of its affiliates

## Trademark Notices

All trademarks, service marks, product names, and logos included in this document are the property of their respective owners.

## 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 24, 2020. 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 .....	5
Contacting Micro Focus Fortify Customer Support .....	5
For More Information .....	5
About the Documentation Set .....	5
Change Log .....	6
Chapter 1: Introduction .....	8
Software Requirements .....	9
Related Documents .....	10
Micro Focus Fortify ScanCentral SAST .....	11
Micro Focus Fortify Software Security Center .....	11
Micro Focus Fortify Static Code Analyzer .....	12
Chapter 2: Installation and Configuration .....	13
Installing the Fortify Jenkins Plugin .....	13
Verifying the Fortify Jenkins Plugin Installation .....	13
Preparing Fortify Software Security Center to Work with the Fortify Jenkins Plugin .....	14
Configuring Global Settings for the Fortify Jenkins Plugin .....	15
Preparing Docker to Work with the Fortify Jenkins Plugin .....	18
Chapter 3: Using the Fortify Jenkins Plugin .....	19
Configuring Fortify Analysis with Freestyle Projects .....	19
Creating a Post-Build Action to Translate and Scan Remotely .....	19
Creating a Post-Build Action to Translate Locally and Scan Remotely .....	21
Creating a Post-Build Action to Translate and Scan Locally .....	24
Creating a Post-Build Action to Upload Scan Results to Fortify Software Security Center ...	29
Configuring Fortify Analysis with Pipeline Jobs .....	30
Pipeline Steps to Translate and Scan Remotely .....	30
fortifyRemoteArguments Step .....	31
fortifyRemoteArguments Example .....	32

fortifyRemoteAnalysis Step .....	32
fortifyRemoteAnalysis Example .....	34
Pipeline Steps to Translate Locally and Scan Remotely .....	34
fortifyRemoteScan Step .....	36
fortifyRemoteScan Example .....	36
Pipeline Steps to Translate and Scan Locally .....	37
fortifyUpdate Step .....	39
fortifyUpdate Example .....	40
fortifyClean Step .....	40
fortifyClean Example .....	41
fortifyTranslate Step .....	41
fortifyTranslate Examples .....	44
fortifyScan Step .....	46
fortifyScan Example .....	47
fortifyUpload Step .....	47
fortifyUpload Example .....	48
Viewing Analysis Results .....	49
Security Vulnerability Graph for Your Project .....	49
Viewing Issues .....	50
Configuring the Number of Issues Displayed on a Page .....	51
Send Documentation Feedback .....	52

# Preface

## Contacting Micro Focus 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/solutions/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 Micro Focus Product Documentation website:

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

# 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.

<b>Software Release / Document Version</b>	<b>Changes</b>
20.2.0 / Revision 1: December 2020	Updated: <ul style="list-style-type: none"><li>Updated the Fortify ScanCentral product name to reflect the change to Fortify ScanCentral SAST</li></ul>
20.2.0	Added: <ul style="list-style-type: none"><li><a href="#">"Preparing Docker to Work with the Fortify Jenkins Plugin" on page 18</a></li><li><a href="#">"Creating a Post-Build Action to Upload Scan Results to Fortify Software Security Center" on page 29</a></li></ul> Updated: <ul style="list-style-type: none"><li><a href="#">"Configuring Global Settings for the Fortify Jenkins Plugin" on page 15</a> - Described optional Jenkins environment variables and added fields for connection timeout, read timeout, and write timeout</li><li><a href="#">"Creating a Post-Build Action to Translate and Scan Locally" on page 24</a> and <a href="#">"fortifyUpload Step" on page 47</a> - Added timeout field</li></ul>
20.1.0	Updated: <ul style="list-style-type: none"><li>Replaced all references to Micro Focus Fortify CloudScan with the new product name: Micro Focus Fortify ScanCentral</li><li><a href="#">"Creating a Post-Build Action to Translate and Scan Locally" on page 24</a> and <a href="#">"fortifyUpdate Step" on page 39</a> - Added ability to specify Security Content locale</li></ul>
19.2.0	Updated: <ul style="list-style-type: none"><li>Changes were made throughout this document to describe the new ability to perform a Fortify Static Code Analyzer analysis on a remote system using Micro Focus Fortify CloudScan.</li></ul>
19.1.0	Added: <ul style="list-style-type: none"><li><a href="#">"Configuring Fortify Analysis with Pipeline Jobs" on page 30</a></li></ul>

<b>Software Release / Document Version</b>	<b>Changes</b>
	<p>Updated:</p> <ul style="list-style-type: none"><li>• <a href="#">"Installing the Fortify Jenkins Plugin" on page 13</a> - Instructions now describe how to obtain the plugin from the Jenkins website</li><li>• <a href="#">"Creating a Post-Build Action to Translate and Scan Locally" on page 24</a><ul style="list-style-type: none"><li>• New field added to specify source files for .NET type projects, the upload wait time setting was changed to polling interval, and other minor changes</li><li>• Changes were made to the description of how to configure uploading results to Fortify Software Security Center</li></ul></li></ul>

# Chapter 1: Introduction

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

- Translate all source code files into intermediate files
- Scan the source to complete the security analysis

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

- Offload the complete analysis (translation and scan) to Fortify ScanCentral SAST  
See "[Software Requirements](#)" on the next page for a list of languages that this method of analysis supports.
- Perform a translation on the local system and then offload the more CPU-intensive scan phase to Fortify ScanCentral SAST
- Perform the complete analysis (translation and scan) 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 analysis results to a Fortify Software Security Center server.

**For complete analysis run locally only:** If you upload the analysis results to a Micro Focus 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.

## 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
Micro Focus Fortify Static Code Analyzer	18.20 or later	<p>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 "<a href="#">Configuring Global Settings for the Fortify Jenkins Plugin</a>" on page 15).</p> <p>For supported versions of Maven, MSBuild, and Visual Studio, see the <i>Micro Focus Fortify Software System Requirements</i> document for your version of Fortify Static Code Analyzer (<a href="https://www.microfocus.com/documentation/fortify-static-code">https://www.microfocus.com/documentation/fortify-static-code</a>).</p> <p><b>Note:</b> Performing remote analysis requires Fortify Static Code Analyzer version 19.2.0 or later.</p>
Micro Focus Fortify Software Security Center (Optional)	18.20 or later	<p>To upload scan results to Fortify Software Security Center, to trigger a build failure based on scan results, and to see results in Jenkins, make sure that you have:</p> <ul style="list-style-type: none"><li>• The Fortify Software Security Center URL</li><li>• A Fortify Software Security Center authentication token of type CIToken (see "<a href="#">Preparing Fortify Software Security Center to Work with the Fortify Jenkins Plugin</a>" on page 14)</li></ul> <p>To perform a remote analysis, make sure that you have:</p> <ul style="list-style-type: none"><li>• A Fortify Software Security Center authentication of type ScanCentralCtrlToken (see "<a href="#">Preparing Fortify Software Security Center to Work with the Fortify Jenkins Plugin</a>" on page 14)</li></ul> <p><b>Note:</b> The ScanCentralCtrlToken was introduced in Fortify Software Security Center version 20.1.0. Use the CloudCtrlToken with Fortify Software Security Center versions 19.2.x and earlier.</p>

Software	Version	Notes
Micro Focus Fortify ScanCentral SAST (Optional)	19.2.0 or later	<p>To perform a Fortify Static Code Analyzer analysis on a remote system using Fortify ScanCentral SAST, make sure that you have properly configured Fortify ScanCentral SAST and you have the ScanCentral Controller URL.</p> <p><b>Note:</b> If you plan to 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.</p> <p>Fortify ScanCentral SAST supports offloading project translation for .NET applications in C# and VB.NET (.NET Core, .NET Standard, and ASP.NET), ABAP, Apex, Classic ASP, ColdFusion, Java (including Gradle and Maven projects), JavaScript, PHP, PL/SQL, Python, Ruby, T-SQL, TypeScript, and Visual Basic.</p> <p><b>Note:</b> Translation of .NET requires the following:</p> <ul style="list-style-type: none"><li>• Fortify CloudsScan version 19.2.0 requires .NET Framework 4.6.1 or later.</li><li>• Fortify ScanCentral version 20.1.0 or later requires .NET Framework version 4.7.2 or later.</li></ul>

To integrate the scan with Maven, you must install the Fortify Maven plugin, which is available when you install Fortify SCA and Apps. Fortify 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 need to 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 *Micro Focus Fortify Static Code Analyzer User Guide*.

To integrate the scan with devenv, you must install the Micro Focus Fortify Extension for Visual Studio. For more information, see the *Micro Focus Fortify Extension for Visual Studio User Guide*.

## Related Documents

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

**Note:** You can find the Micro Focus Fortify Product Documentation at <https://www.microfocus.com/support/documentation>.

## Micro Focus Fortify ScanCentral SAST

The following document provides information about Fortify ScanCentral SAST. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at <https://www.microfocus.com/documentation/fortify-software-security-center>.

Document / File Name	Description
<i>Micro Focus Fortify ScanCentral SAST Installation, Configuration, and Usage Guide</i> 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.

## Micro Focus Fortify Software Security Center

The following document provides information about Fortify Software Security Center. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at <https://www.microfocus.com/documentation/fortify-software-security-center>.

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

## Micro Focus Fortify Static Code Analyzer

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

Document / File Name	Description
<i>Micro Focus Fortify Static Code Analyzer User Guide</i> 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.
<i>Micro Focus Fortify Static Code Analyzer Custom Rules Guide</i> 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.  <b>Note:</b> This document is included only with the product download.

# Chapter 2: Installation and Configuration

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

This section contains the following topics:

Installing the Fortify Jenkins Plugin .....	13
Preparing Fortify Software Security Center to Work with the Fortify Jenkins Plugin .....	14
Configuring Global Settings for the Fortify Jenkins Plugin .....	15
Preparing Docker to Work with the Fortify Jenkins Plugin .....	18

## Installing the Fortify Jenkins Plugin

To install the Fortify Jenkins Plugin, you must have Jenkins installed on your system. See the *Micro Focus 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, click 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.  
By default, the Jenkins URL is `http://localhost:8080`.
2. From the Jenkins menu, select **Manage Jenkins > Manage Plugins**.

3. On the **Plugin Manager** page, click the **Installed** tab.
4. Verify that **Fortify Jenkins Plugin** is included in the list of installed plugins.

## Preparing Fortify Software Security Center to Work with the Fortify Jenkins Plugin

To perform the following tasks, you need to obtain an authentication token created in Fortify Software Security Center. You will 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
Perform a remote Fortify Static Code Analyzer analysis using Fortify ScanCentral SAST (this includes the ability to upload the remote scan results to Fortify Software Security Center)	ScanCentralCtrlToken <b>Note:</b> Do not use the CloudCtrlToken type in Fortify Software Security Center versions 20.1.0 and later.

You can generate the authentication token from either the Administration view in Fortify Software Security Center or from the command-line with the `fortifyclient` utility.

**Note:** If you generate the token from Fortify Software Security Center, use the decoded token to configure the Fortify Jenkins Plugin.

The following instructions describe how to create the authentication token with the `fortifyclient` utility. For information about how to create an authentication token from Fortify Software Security Center, see the *Micro Focus Fortify Software Security Center User Guide*.

To create an authentication token using the `fortifyclient` utility:

1. From the `<ssc_install_dir>/Tools/fortifyclient/bin` directory, run the following:

```
fortifyclient token -gettoken <token_type> -url <ssc_url> -user <user_name>  
[-daysToLive <number_of_days>]
```

**Note:** Find the `Tools` folder in the directory where the Fortify Software Security Center WAR file was extracted.

where:

- `<token_type>` is either:
  - `CIToken` to upload locally-created FPR files to Fortify Software Security Center
  - `ScanCentralCtrlToken` to offload the analysis to a remote system
- `<ssc_url>` includes both the port number and the context path `/ssc`. For example, `http://my.domain.com:8080/ssc`.
- `<user_name>` is the Fortify Software Security Center user name of an account that has the required privileges to read or write information from or to Fortify Software Security Center.
- `<number_of_days>` is the number of days before the token expires. The default is 365.

You are prompted for a password.

2. Type the password for `<user_name>`.

The `fortifyclient` utility displays a token of the general form:

```
cb79c492-0a78-44e3-b26c-65c14df52e86.
```

3. Copy the returned token to use when you configure the Fortify Jenkins Plugin (see ["Configuring Global Settings for the Fortify Jenkins Plugin" below](#)).

## 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 > Configure System**.
3. If you are analyzing (translate or scan) your project locally with Fortify Static Code Analyzer, you can create a Jenkins environment variable 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 provide paths that end in `/bin`. The Fortify Jenkins Plugin looks for the location of a build tool executable in the following order: Jenkins environment variable, the `PATH` system environment variable, and lastly the build's workspace.

Environment Variable	Applies To
<p><b>Name:</b> FORTIFY_HOME</p> <p><b>Value:</b> <code>&lt;sca_install_dir&gt;</code></p> <p>where <code>&lt;sca_install_dir&gt;</code> is the path where Fortify Static Code Analyzer is installed. For example, the default location on Windows is <code>C:\Program Files\Fortify\Fortify_SCA_and_Apps_&lt;sca_version&gt;</code>.</p>	Local analysis
<p><b>Name:</b> GRADLE_HOME</p> <p><b>Value:</b> <code>&lt;Gradle_install_dir&gt;</code></p> <p>where <code>&lt;Gradle_install_dir&gt;</code> is the path where Gradle is installed.</p>	Local analysis of a Gradle project
<p><b>Name:</b> M2_HOME or MAVEN_HOME</p> <p><b>Value:</b> <code>&lt;Maven_install_dir&gt;</code></p> <p>where <code>&lt;Maven_install_dir&gt;</code> is the path where Apache Maven is installed.</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>Note:</b> The Fortify Maven Plugin must be installed. See <a href="#">"Software Requirements" on page 9</a> for more information.</p> </div>	Local analysis of a Maven project

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:

Software Security Center configuration

SSC URL  ?  
▲ URL cannot be empty

Authentication token  ?  
▲ Authentication token cannot be empty

Use proxy

Issue template  ▼ ?

Maximum issues per page  ?

Connection timeout  ?

Read timeout  ?

Write timeout  ?

- a. In the **SSC URL** box, type the Fortify Software Security Center server URL.  
 The correct format for the Fortify Software Security Center URL is:  
`<protocol>://<ssc_host>:<port>/ssc` (for example:  
`https://my.domain.com:8080/ssc`).
- b. In the **Authentication token** box, type the decoded value of a Fortify Software Security Center authentication token of type CIToken.

See ["Preparing Fortify Software Security Center to Work with the Fortify Jenkins Plugin"](#) on page 14.

- c. To connect to Fortify Software Security Center with a proxy server, select **Use proxy**, and then specify the proxy information.
- d. 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.
- e. To test the connection to Fortify Software Security Center, click **Test SSC connection**.
- 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 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.
  - h. (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.
  - i. (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:

#### Controller configuration

Controller URL

⚠ Controller URL cannot be empty

Controller token

⚠ Controller token cannot be empty

Test Controller connection

- 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 correct format for the ScanCentral Controller URL is:

`<protocol>://<controller_host>:<port>/scancentral-ctrl` (for example:  
`https://myControllerHost.com:8443/scancentral-ctrl`).

- b. In the **Controller token** box, type the decoded value of the Fortify Software Security Center authentication token of type ScanCentralCtrlToken.

See "[Preparing Fortify Software Security Center to Work with the Fortify Jenkins Plugin](#)" on [page 14](#) for instructions.

6. Click **Save**.

## Preparing Docker to Work with the Fortify Jenkins Plugin

If you are running 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_and_Apps_20.2.0:/var/jenkins_home/Fortify/Fortify_SCA_and_Apps_20.2.0 --name=Jenkins jenkins/jenkins -d
```

In the previous example, the value of FORTIFY\_HOME is `/var/jenkins_home/Fortify/Fortify_SCA_and_Apps_20.2.0`.

# Chapter 3: Using the Fortify Jenkins Plugin

This section describes how to add Fortify analysis to a project as a post-build action or with a Pipeline script.

This section contains the following topics:

- [Configuring Fortify Analysis with Freestyle Projects](#) ..... 19
- [Configuring Fortify Analysis with Pipeline Jobs](#) ..... 30
- [Viewing Analysis Results](#) ..... 49

## 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 30.

### 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
Gradle	<ol style="list-style-type: none"><li>a. To include the test source set (for Java projects only) with translation, select the <b>Include tests</b> check box.</li><li>b. In the <b>Build file</b> box, type the name of the build file if it is different than the default of <code>build.gradle</code>.</li></ol>

Application Type	Description
Maven	<ul style="list-style-type: none"> <li>a. To include a test scope (for Java projects only) with translation, select the <b>Include tests</b> check box.</li> <li>b. In the <b>Build file</b> box, type the name of the build file if it is different than the default of <code>pom.xml</code>.</li> </ul>
PHP	<ul style="list-style-type: none"> <li>a. In the <b>PHP version</b> box, type the PHP version used in the project.</li> </ul>
Python	<ul style="list-style-type: none"> <li>a. In the <b>Python version</b> box, select the Python version used in the project. The default version is 2.</li> <li>b. In the <b>Python virtual environment</b> box, type the location (directory) of the Python virtual environment.</li> <li>c. In the <b>Python requirements file</b> box, type the name of the Python project requirements file used to install and collect dependencies.</li> </ul>
Other	Use this option to translate and scan other languages. See <a href="#">"Software Requirements" on page 9</a> for the list of languages that Fortify ScanCentral SAST supports for remote translation and scan.

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 *Micro Focus Fortify Static Code Analyzer User Guide*.

**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 ScanCentral 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.

Field	Description
Fortify SCA scan options	Specify Fortify Static Code Analyzer scan options. For descriptions of the available scan options, see the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> .  <b>Note:</b> 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 <i>Micro Focus Fortify Static Code Analyzer User Guide</i> .

- To upload the scan results to Fortify Software Security Center:
  - Select the **Upload Fortify SCA scan results to Fortify Software Security Center** check box.
  - Select 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.
- 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:

- 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.
- In the **Post-build Actions** section, click **Add post-build action**, and then select **Fortify Assessment**.
- Select **Local translation & remote scan**.
- To download Fortify security content:

**Note:** This setting only updates Fortify security content on the local system and is used in the local translation. Make sure that you have the same version of Fortify security content on both the local system and the remote system.

- Select the **Update Fortify Security Content** check box.
- 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>.

**Note:** To connect to the Fortify Rulepack update server with a proxy server, you need to configure Fortify security content update settings with the `scapostinstall` tool. For instructions, see the *Micro Focus Fortify Static Code Analyzer User Guide*.

- c. (Optional) In the **Locale** box, select the locale for the Fortify Rulepack.

The default is English.

- 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 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 an amount of memory in this field, 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 environment variable, the PATH system environment variable, and lastly the build's workspace.

Application Type	Description
.NET Devenv	<ul style="list-style-type: none"> <li>a. In the <b>Solution or project file</b> box, type the solution or project file name (or the path to the file).</li> <li>b. (Optional) Specify any additional devenv options.</li> </ul>
.NET MSBuild	<ul style="list-style-type: none"> <li>a. In the <b>Solution or project file</b> box, type the solution or project file name (or the path to the file).</li> <li>b. (Optional) Specify any additional MSBuild options.</li> </ul>
.NET source code scan	<ul style="list-style-type: none"> <li>a. In the <b>.NET framework version</b> box, specify the .NET framework version used to compile the code.</li> <li>b. (Optional) In the <b>Libdirs</b> box, specify a semicolon-separated list of directories where referenced system or third-party DLLs are located.</li> </ul>

<b>Application Type</b>	<b>Description</b>
	<p>c. (Optional) In the <b>Fortify SCA translation options</b> box, specify any additional Fortify Static Code Analyzer translation options. See the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> for detailed information about the available translation options.</p> <p>d. In the <b>Source files</b> box, specify the source files to translate.</p>
Java	<p>Specify the Java source path, classpath, the source files, and any additional Fortify Static Code Analyzer translation options. The only required field is <b>Source files</b>. See the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> for more detailed information about the Java translation options.</p>
Maven	<p>a. If you did not run the build previously, then in the <b>Maven options</b> box, type package. Otherwise, leave this box empty.</p> <div data-bbox="488 869 1401 1056" style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <p><b>Note:</b> 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.</p> </div>
Gradle	<p>a. To use a Wrapper, select <b>Use Gradle Wrapper</b>.</p> <p>b. In the <b>Gradle tasks</b> box, type the Gradle tasks required for your project.</p> <p>c. In the <b>Gradle options</b> box, type the Gradle options required for your project.</p>
Other	<p>a. (Optional) Provide all the Fortify Static Code Analyzer translation options in the <b>Fortify SCA translation options</b> box. See the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> for detailed information about the available translation options.</p> <p>b. Specify the source code to scan in the <b>Includes list</b> box.</p>
Advanced	<p>Select <b>Advanced</b> if you are familiar with the Fortify Static Code Analyzer command-line interface or you want to specify all the translation options without any guidance. Specify all the Fortify Static Code Analyzer translation options including source files. See the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> for detailed information about the translation options.</p>

9. (Optional) You can exclude files or directories from the translation by adding them to the **Exclude list** box.
10. (Optional) Enable the debug or verbose logging options.

11. (Optional) To specify ScanCentral 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 <i>Micro Focus Fortify Static Code Analyzer User Guide</i> .  <b>Note:</b> 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: <code>"-analyzers" "controlflow,dataflow" "-verbose"</code> .
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 <i>Micro Focus Fortify Static Code Analyzer User Guide</i> .

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. Select 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.
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 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`.

**Note:** To connect to the Fortify Rulepack update server with a proxy server, you need to configure Fortify security content update settings with the `scapostinstall` tool. For instructions, see the *Micro Focus Fortify Static Code Analyzer User Guide*.

- c. (Optional) In the **Locale** box, select the locale for the Fortify Rulepack.  
The default is English.
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 SCA scan, the analysis 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 SCA scan and you are uploading results to Fortify Software Security Center, 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 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 an amount of memory in this field, 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 environment variable, the `PATH` system environment variable, and lastly the build's workspace.

<b>Application Type</b>	<b>Description</b>
.NET Devenv	a. In the <b>Solution or project file</b> 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 <b>Solution or project file</b> 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 <b>.NET framework version</b> box, specify the .NET framework version used to compile the code. b. (Optional) In the <b>Libdirs</b> box, specify a semicolon-separated list of directories where referenced system or third-party DLLs are located. c. (Optional) In the <b>Fortify SCA translation options</b> box, specify any additional Fortify Static Code Analyzer translation options. See the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> for detailed information about the available translation options. d. In the <b>Source files</b> box, specify the source files to translate.
Java	Specify the Java source path, classpath, the source files, and any additional Fortify Static Code Analyzer translation options. The only required field is <b>Source files</b> . See the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> for more detailed information about the Java translation options.
Maven	a. If you did not run the build previously, then in the <b>Maven options</b> box, type package. Otherwise, leave this box empty. <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> 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.</p> </div>
Gradle	a. To use a Wrapper, select <b>Use Gradle Wrapper</b> . b. In the <b>Gradle tasks</b> box, type the Gradle tasks required for your project. c. In the <b>Gradle options</b> box, type the Gradle options required for your project.

<b>Application Type</b>	<b>Description</b>
Other	a. (Optional) Provide all the Fortify Static Code Analyzer translation options in the <b>Fortify SCA translation options</b> box. See the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> for detailed information about the available translation options.  b. Specify the source code to scan in the <b>Includes list</b> box.
Advanced	Select <b>Advanced</b> if you are familiar with the Fortify Static Code Analyzer command-line interface or you want to specify all the translation options without any guidance. Specify all the Fortify Static Code Analyzer translation options including source files. See the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> for detailed information about the translation options.

10. (Optional) You can exclude files or directories from the translation by adding 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.

**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.

If you have a successful connection to a Fortify Software Security Center server, you can select an application name and version from the list. Always specify both application name and application version.

**Note:** If an application with the specified name and version does not exist on Fortify Software Security Center, Fortify Jenkins Plugin creates it for a successful build.

- b. (Optional) Specify the ID of a filter set to use when retrieving scan results for display in Jenkins. If no value is specified, the Fortify Jenkins Plugin uses the default filter set configured in Fortify Software Security Center.

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 fail condition 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 fail condition 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 failure based on 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 *Micro Focus Fortify Software Security Center User Guide* for a description of the search query syntax.

- d. (Optional) To specify how long to poll Fortify Software Security Center to check 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 will continue until FPR processing finishes or stops due to errors. The valid values are 0–10080.
- e. (Optional) To specify the interval that the Fortify Jenkins Plugin polls Fortify Software Security Center to check FPR processing is finished, click **Advanced settings**. In the **Polling interval** field, specify an interval (in minutes). The valid values are 0–60 and the default is 1 minute. The Fortify Jenkins Plugin polls Fortify Software Security Center until the FPR is processed before it runs the NVS calculation.

**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 to only upload 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 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. The value can be a name or a full file 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.
5. Specify an application name and an application version.  
If you have a successful connection to a Fortify Software Security Center server, you can select an application name and version from the list. Always specify both application name and application version.

**Note:** If an application with the specified name and version does not exist on Fortify Software Security Center, Fortify Jenkins Plugin creates it for a successful build.

6. (Optional) Specify the ID of a filter set to use when retrieving scan results for display in Jenkins. If no value is specified, the Fortify Jenkins Plugin uses the default filter set configured in Fortify Software Security Center.  
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 fail condition 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 fail condition 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 failure based on 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 *Micro Focus Fortify Software Security Center User Guide* for a description of the search query syntax.

8. (Optional) To specify how long to poll Fortify Software Security Center to check 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 will continue until FPR processing finishes or stops due to errors. The valid values are 0–10080.

9. (Optional) To specify the interval that the Fortify Jenkins Plugin polls Fortify Software Security Center to determine if the FPR processing is complete, click **Advanced settings**. In the **Polling interval** field, specify an interval (in minutes). The valid values are 0–60 and the default is 1 minute.

The Fortify Jenkins Plugin polls Fortify Software Security Center until the FPR is processed before it runs the NVS calculation.

**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**.

## 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.

### 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 <code>fortifyRemoteAnalysis</code> step.	<a href="#">"fortifyRemoteArguments Step" on the next page</a>
Send a project to a remote system for analysis.	<a href="#">"fortifyRemoteAnalysis Step" on page 32</a>

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 analysis 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 initiate a remote analysis, follow this step with a `fortifyRemoteAnalysis` step (see ["fortifyRemoteAnalysis Step" on the next page](#)).

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

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

### 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"'
  }
}
```

### 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 (see "[fortifyRemoteArguments Step](#)" on the previous page).

Parameter	Description	Default Value
RemoteAnalysisProjectType	Required (String). The project type is one of the following: fortifyGradle, fortifyMaven, fortifyPHP, fortifyPython, or fortifyOther.	
<b>Gradle Parameters</b>		
includeTests	Optional (boolean). Specifies whether to include a test source set.	false
buildFile	Optional (String). Specifies the build file name.	build.gradle
<b>Maven Parameters</b>		
includeTests	Optional (boolean). Specifies whether to include a test scope.	false
buildFile	Optional (String). Specifies the build file name.	pom.xml

Parameter	Description	Default Value
<b>PHP Parameters</b>		
phpVersion	Optional (Number). Specifies the PHP version used in the project.	7.0
<b>Python Parameters</b>		
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)
<b>remoteOptionalConfig Parameters</b>		
sensorPoolUUID	Optional (String). Specifies the sensor pool to which to submit the job.	(none)
notifyEmail	Optional (String). Specifies the email address to which the Controller will send notifications.	(none)
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 <i>Micro Focus Fortify Static Code Analyzer User Guide</i> .	(none)
<b>uploadSSC Parameters</b>		
appName	Required (String). Specifies an existing application name for which to store the results in Fortify Software Security Center.	

Parameter	Description	Default Value
appVersion	Required (String). Specifies an existing application version for which to store the results in Fortify Software Security Center.	

### fortifyRemoteAnalysis Example

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 project is uploaded 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']
  }
}
```

## 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 SCA clean	<a href="#">"fortifyClean Step" on page 40</a>
Run a local Fortify SCA translation	<a href="#">"fortifyTranslate Step" on page 41</a>
Run a Remote Fortify SCA scan	<a href="#">"fortifyRemoteScan Step" on page 36</a>

**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 analysis 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: '11')
  }
  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: '11')
      }
    }
  }
  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	Default Value
buildID	Required (String). A unique identifier for the analysis.	
<b>remoteOptionalConfig Parameters</b>		
sensorPoolUUID	Optional (String). Specifies the sensor pool to which to submit the job.	(none)
notifyEmail	Optional (String). Specifies the email address to which the Controller will send notifications.	(none)
scanOptions	Optional (String). Fortify Static Code Analyzer scan options. For descriptions of the available scan options, see the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> .  <b>Note:</b> Enclose each option and parameter in double quotes. In the following example, two analyzers and quick scan mode are enabled for the scan: <code>"-analyzers" "controlflow,dataflow" "-quick"</code> .	(none)
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 <i>Micro Focus Fortify Static Code Analyzer User Guide</i> .	(none)
<b>uploadSSC Parameters</b>		
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.	

## 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 project is uploaded 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']  
  }  
}
```

## Pipeline Steps to Translate and Scan Locally

The following table lists the available Fortify Jenkins Plugin Pipeline steps to update Fortify 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 security content to use for local translation and scan	<a href="#">"fortifyUpdate Step" on page 39</a>
Run a local Fortify SCA clean	<a href="#">"fortifyClean Step" on page 40</a>
Run a local Fortify SCA translation	<a href="#">"fortifyTranslate Step" on page 41</a>
Run a local Fortify SCA scan	<a href="#">"fortifyScan Step" on page 46</a>
Upload local Fortify SCA scan results to Fortify Software Security Center	<a href="#">"fortifyUpload Step" on page 47</a>

**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 security content, performs a complete Fortify analysis of a Java project, and then uploads the analysis 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: '11')
  }
  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: '11')
      }
    }
    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 security content used by the Fortify translation and scan steps.

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

Parameter	Description	Default Value
	<p>the Fortify Rulepack. Use one of the following locale codes:</p> <ul style="list-style-type: none"> <li>• English: en</li> <li>• Chinese (Simplified): zh_CN</li> <li>• Chinese (Traditional): zh_TW</li> <li>• Portuguese (Brazil): pt_BR</li> <li>• Korean: ko</li> <li>• Spanish: es</li> </ul>	

### fortifyUpdate Example

The following example updates the Fortify security content in English:

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

### 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.	
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

Parameter	Description	Default Value
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 <code>sca.log</code> and the default location is in the workspace directory.

### 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'
  }
}
```

### fortifyTranslate Step

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

Parameter	Description	Default Value
<b>General Parameters</b>		
buildID	Required (String). A unique identifier for the analysis.	
projectScanType	Required. (String). The project scan type is one of the following: <code>fortifyAdvanced</code> , <code>fortifyDevenv</code> , <code>fortifyDotnetSrc</code> , <code>fortifyGradle</code> , <code>fortifyJava</code> , <code>fortifyMaven3</code> , <code>fortifyMSBuild</code> , or <code>fortifyOther</code> .	
maxHeap	Optional (int). The maximum heap size for the JVM ( <code>-Xmx</code> ).	By default, Fortify Static Code Analyzer automatically allocates memory based

Parameter	Description	Default Value
		on the physical memory available on the system.
addJVMOptions	Optional (String). 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 <code>sca.log</code> and the default location is the workspace directory.
excludeList	Optional (String). Specifies a list of directories or files to exclude from translation.	(none)
<b>Java Parameters</b>		
javaSrcFiles	Required (String). Specifies the location of the Java source files.	
javaVersion	Optional (String). Specifies the JDK version for which the Java code is written.	The default version defined by Fortify Static Code Analyzer. For example, in Fortify Static Code Analyzer version 18.20, the default JDK version is 1.8. See the <i>Micro Focus Fortify Static Code Analyzer User Guide</i> for specific version information.
javaClasspath	Optional (String). Specifies the class path as a colon- or semicolon-separated list of	(none)

Parameter	Description	Default Value
	directories to use for analyzing Java source code.	
javaAddOptions	Optional (String). Specifies any additional Fortify Static Code Analyzer options for translating Java code.	(none)
<b>devenv / MSBuild Parameters</b>		
dotnetProject	Required (String). Specifies a solution (*.sln) or a project (*.proj) file.	
dotnetAddOptions	Optional (String). Specifies any additional Fortify Static Code Analyzer options for translating .NET code.	(none)
<b>DotnetSrc Parameters</b>		
dotnetFrameworkVersion	Required (int). Specifies the .NET framework version.	
dotnetSrcFiles	Required (String). Specifies the location of the .NET source files.	
dotnetLibdirs	Optional (String). Specifies a semicolon-separated list of directories where referenced system or third-party DLLs are located.	(none)
dotnetAddOptions	Optional (String). Specifies any additional devenv or MSBuild options required for your project.	(none)
<b>Maven3 Parameters</b>		
mavenOptions	Optional (String). Specifies	(none)

Parameter	Description	Default Value
	any additional Maven options required for your project.	
<b>Gradle Parameters</b>		
gradleTasks	Required (String). Specifies the Gradle tasks required for your project.	
useWrapper	Optional (boolean). Specifies whether to use a Wrapper.	false
gradleOptions	Optional (String). Specifies any additional Gradle options required for your project.	(none)
<b>Other Parameters</b>		
otherIncludesList	Required (String). Specifies the location of the source files.	
otherOptions	Optional (String). Specifies any additional Fortify Static Code Analyzer options required for your project.	(none)
<b>Advanced Parameters</b>		
advOptions	Required (String). Specifies all the Fortify Static Code Analyzer options that are necessary to translate the project.	

### 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: '1.8')
  }
}
```

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')  
  }  
}
```

## 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.	
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	false

Parameter	Description	Default Value
	the console and to the Fortify Support log file.	
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.

### 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'
  }
}
```

### fortifyUpload Step

Use this step to upload the results of a Fortify analysis (FPR) to Micro Focus 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 15](#)). After an upload you can view the results in Jenkins (see ["Viewing Analysis Results" on page 49](#)).

Parameter	Description	Default Value
appName	Required (String). Specifies the application name for which to store the results in Fortify Software Security Center.	
appVersion	Required (String). Specifies the application version for which to store the results in Fortify Software Security Center.	
resultsFile	Optional (String). Specifies a name for the Fortify results file (FPR). For example, MyProjectA.fpr.	If you ran a Fortify SCA scan, the default file is scan.fpr, otherwise the Fortify Jenkins Plugin searches ".*/*.fpr" in the workspace for the FPR file with the latest modified date.

Parameter	Description	Default Value
filterSet	<p>Optional (String). Specifies the ID of a filter set to use when retrieving scan results for display in Jenkins.</p> <p>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.</p>	The default filter set configured in Fortify Software Security Center.
failureCriteria	<p>Optional (String). Specifies a search query to use on the scan results to trigger a build failure. For example, [fortify priority order]:critical.</p>	(none)
timeout	<p>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 will continue until FPR processing finishes or stops due to errors. The valid values are 0–10080.</p> <p><b>Note:</b> This parameter is not available in the Snippet Generator.</p>	(none)
pollingInterval	<p>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.</p> <p><b>Important!</b> If the FPR processing requires approval, then this step will not complete until the approval is granted through Fortify Software Security Center.</p>	1

### fortifyUpload Example

The following example uploads the Fortify analysis results for the MyJavaApp project to version 3 of the MyJavaApp application on Fortify Software Security Center:

```
node {  
  stage('Fortify Upload') {
```

```
fortifyUpload appName: 'MyJavaApp', appVersion: '3',  
resultsFile: 'MyJavaApp.fpr'  
}  
}
```

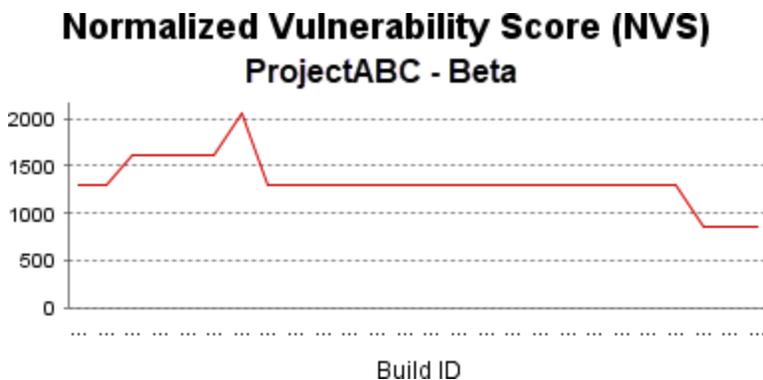
## Viewing Analysis Results

When you perform the Fortify analysis on the local system and if you uploaded Micro Focus Fortify Static Code Analyzer results to Micro Focus 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.

## 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 a rough idea of the security vulnerability of your project.



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

$$\text{NVS} = ((\text{CFPO} * 10) + (\text{HFPO} * 5) + (\text{MFPO} * 1) + (\text{LFPO} * 0.1)) * 0.5 + ((\text{P1} * 2) + (\text{P2} * 4) + (\text{P3} * 16) + (\text{PABOVE} * 64)) * 0.5$$

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 very 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:  $NVS = 0 \cdot 10 + 10 \cdot 0.1 = 1$
- Application B:  $NVS = 5 \cdot 10 + 0 \cdot 0.1 = 50$

## Viewing Issues

To see the issues for a Fortify Static Code Analyzer analysis that you have uploaded to Micro Focus 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.

### List of Fortify SSC issues

#### Summary

Build	Total	Critical	High	Medium	Low
#7 (#8)	77 (77)	0 (0)	4 (4)	0 (0)	73 (73)

#### Issues breakdown by Priority Order

Critical (0) High (4) Medium (0) Low (73) All (1 to 50 out of 77)

Primary Location	Category
<a href="#">Exec.java:292</a>	Command Injection
<a href="#">Exec.java:103</a>	Command Injection
<a href="#">Exec.java:202</a>	Dead Code: Expression is Always false
<a href="#">Exec.java:156</a>	Dead Code: Expression is Always false
<a href="#">Exec.java:111</a>	Dead Code: Expression is Always false
<a href="#">Exec.java:118</a>	Dead Code: Expression is Always true
<a href="#">Exec.java:418</a>	Denial of Service
<a href="#">Exec.java:229</a>	Denial of Service
<a href="#">ExecResults.java:335</a>	Denial of Service: StringBuiler

[Next >>](#)

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, click 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 Micro Focus Fortify 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 20.2.0)**

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@microfocus.com](mailto:FortifyDocTeam@microfocus.com).

We appreciate your feedback!