Micro Focus
Fortify Jenkins Plugin

Software Version: 19.1.0

User Guide

Document Release Date: March 2019

Software Release Date: February 2019

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

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

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-and-services/documentation

Contents

Preface

Contacting Micro Focus Fortify Customer Support

If you have questions or comments about using this product, contact Micro Focus Fortify Customer Support using one of the following options.

To Manage Your Support Cases, Acquire Licenses, and Manage Your Account

https://softwaresupport.softwaregrp.com

To Call Support

1.844.260.7219

For More Information

For more information about Fortify software products:
https://software.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-and-services/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.

Software Release / Document Version	Changes
19.1.0	Added: • "Configuring Fortify Analysis with Pipeline Jobs" on page 17 Updated: • "Installing the Fortify Jenkins Plugin" on page 10 - Instructions now describe how to obtain the plugin from the Jenkins website • "Configuring a Build Step to use the Fortify Jenkins Plugin" on page 13 • New field added to specify source files for .NET type projects and other minor changes and the upload wait time setting was changed to polling interval • Changes were made to the description of how to configure uploading results to Fortify Software Security Center
18.20	Updated to describe the new capability that enables you to scan projects with Fortify Static Code Analyzer as part of the build.
18.10	Updated: • Minor edits to incorporate branding changes • "Preparing Fortify Software Security Center to Work with the Fortify Jenkins Plugin" on page 11 - Updated the token type and the instructions for how to create an authentication token

Fortify Jenkins Plugin

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. After the Fortify Static Code Analyzer analysis is complete, you can upload the results to a Micro Focus Fortify Software Security Center server. The Fortify Jenkins Plugin also enables you to view the analysis result details within Jenkins. It provides metrics for each build and an overview of the results, without requiring you to log into Fortify Software Security Center.

With the Fortify Jenkins Plugin, you can integrate Fortify Static Code Analyzer with the following build tools:

- Gradle
- Maven
- MSBuild
- Visual Studio (devenv)

You can also scan your source code directly without a build tool.

This document provides instructions on how to prepare Fortify Software Security Center to work with the Fortify Jenkins Plugin, and 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 for the configuration of your buld step.

Software	**Version**	**Notes**
Micro Focus Fortify Static Code Analyzer (Optional)	18.20 or later	To scan your project 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 the Fortify Jenkins Plugin" on page 12).

Software	Version	Notes
Micro Focus 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 scan results, and to see 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 Fortify Software Security Center to Work with the Fortify Jenkins Plugin" on page 11)
Maven	3.x	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*.
MSBuild	4.x, 12.0, 14.0, 15.0	
Visual Studio (devenv)	2013, 2015, 2017	

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-and-services/documentation.

Micro Focus Fortify Software Security Center

The following documents provide information about Fortify Software Security Center. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website.

Document / File Name	Description
Micro Focus 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 Software Security Center. It provides all of the information you need to acquire, install, configure, and use Software Security Center. 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.

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.

Document / File Name	Description
Micro Focus Fortify Static Code Analyzer Installation Guide SCA_Install_*<version>*.pdf	This document contains installation instructions for Fortify Static Code Analyzer and Applications.
Micro Focus Fortify Static Code Analyzer User Guide SCA_Guide_*<version>*.pdf	This document describes how to 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.
Micro Focus Fortify Static Code Analyzer Performance Guide	This document provides guidelines for selecting hardware to scan different types of codebases and offers tips for optimizing memory usage and performance.

Document / File Name	Description
SCA_Perf_Guide_<*version*>.pdf	
Micro Focus 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.

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 checkbox 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

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 upload Fortify Static Code Analyzer results to Fortify Software Security Center or to view Fortify Static Code Analyzer results from Jenkins, you need to have an authentication token of type CIToken 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.

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 of type CIToken using the fortifyclient utility:

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

   ```
   fortifyclient token -gettoken CIToken -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:

   - *<ssc_url>* includes both the port number and the context path /ssc. For example, http://*<hostname>*>:*<port>*/ssc.
   - *<user_name>* is the Fortify Software Security Center username 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 the Fortify Jenkins Plugin" on the next page).

Configuring the Fortify Jenkins Plugin

To configure your Jenkins server so that it can analyze your project, update Fortify security content, 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. To analyze your project with Fortify Static Code Analyzer or to update Fortify security content as part of your build, create a Jenkins environment variable to specify the location of the Fortify Static Code Analyzer executables. In **Global properties**, create the following environment variable:

   - **Name:** `FORTIFY_HOME`
   - **Value:** `<sca_install_dir>`

     where `<sca_install_dir>` is the path where Fortify Static Code Analyzer is installed. For example, on Windows the default installation location is `C:\Program Files\Fortify\Fortify_SCA_and_Apps_19.1.0`.

   > **Note:** If the Fortify Jenkins Plugin cannot find the executables (sourceanalyzer and fortifyupdate) using the `FORTIFY_HOME` variable, then it uses the system `PATH` environment variable to find them. Fortify recommends that you specify the full path in Jenkins on Unix systems.
   >
   > You can also set the environment variable on a per-node basis (**Jenkins > Manage Jenkins > Manage Nodes >** ***<node_name>***).
   >
   > If you are using Gradle, you might need to add the following environment variable to ensure access to Fortify Static Code Analyzer:
   >
   > - **Name:** `PATH`
   > - **Value:** `<sca_install_dir>/bin`

4. To upload results to Fortify Software Security Center, scroll down to the **Fortify Assessment** section, and then do the following:

   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:

      `http://<host_IP>:<port>/ssc`.

   b. To connect to Fortify Software Security Center with a proxy server, select **Use Proxy for SSC**, and then specify the proxy information.

      Use the following format for the **Proxy server host:port**:

      `<address>:<port_number>`

   c. In the **Authentication token** box, type the authentication token generated for the Fortify Software Security Center server.

      See "Preparing Fortify Software Security Center to Work with the Fortify Jenkins Plugin" on the previous page.

d. Click **Advanced settings**, and then click **Test Connection**.

The Fortify Jenkins Plugin populates the **Issue Template** list with available Fortify Software Security Center issue templates. 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. From the **Issue template** list, select the appropriate issue template for your projects.

> **Note:** There is no need to specify a value in the **Issue breakdown page size** box at this time. You can change this setting later. This setting controls the **Issue Breakdown** table view. The default is 50 issues per page.

5. Click **Save**.

Configuring a Build Step to use the Fortify Jenkins Plugin

To configure a build step for your project that uses the Fortify Jenkins Plugin:

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

   The Fortify Jenkins Plugin supports Freestyle and Multi-configuration projects.

   > **Note:** The Fortify Jenkins Plugin also supports Jenkins Pipeline. For instructions, see "Configuring Fortify Analysis with Pipeline Jobs" on page 17.

   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. In the **Build ID** box, type a unique identifier for the scan.
4. In the **Results file** box, type a name for the Fortify results file (FPR). For example, `MyProjectA.fpr`.

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

   Specifying the results file name is optional. If you do not provide a 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 FPR file with the latest modified date.

5. (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.
6. (Optional) In the **Additional JVM options** box, you can add additional JVM commands.
7. To download Fortify security content before the scan, select the **Update Fortify Security Content** check box, and specify the following:
   a. 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.
   b. To connect to the Fortify Rulepack update server with a proxy server, select the **Configure update server proxy** check box, and then specify the proxy information.
8. To remove any temporary files from a previous scan for the specified build ID, select the **Run Fortify SCA Clean** check box.

   Fortify recommends that you run the clean phase before each translation unless, for example, you are translating several projects with the same build ID to perform one scan for all the projects and generate a single FPR file.
9. To run translation, select the **Run Fortify SCA translation** check box, and then specify the translation settings.

   You might want to skip the translation if, for example, the security content has changed but the source code has not. If you do skip the translation, make sure that you do not run a Fortify SCA clean.

   > **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"`

   a. Select whether you want to use the basic or advanced configuration.

      Select **Advanced** 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, if needed. See the *Micro Focus Fortify Static Code Analyzer User Guide* for detailed information about the translation options.

      Select **Basic** to be prompted to provide the typical information to scan Java or .NET code or to run a Maven 3, or a Gradle build to perform the translation. The configuration fields dynamically change based on your selection.

      > **Note:** The Fortify Jenkins Plugin uses the PATH environment variable to find the executable for gradle, maven, devenv, and msbuild.

For each of the basic translation configurations, you can exclude files or directories from the translation by including them in the **Exclude list** box. The following table provides instructions for each application type in the basic configuration.

Application Type	Description
Java	Specify the Java source path, classpath, the source files, and any additional Fortify Static Code Analyzer translation options. See the *Micro Focus Fortify Static Code Analyzer User Guide* for more detailed information about the Java translation options.
.NET	i. From the **Scan type** list, select whether to perform a **Project Solution Scan** or a **Source Code Scan**. ii. To translate a solution or a project (**Project Solution Scan**): A. From the **Build type** list, select **devenv** or **MSBuild**. B. In the **Solution or project file** box, type the solution or project file name (or the path to the file). C. Specify any additional devenv or MSBuild options, based on the build type you are using. iii. To translate source code (**Source Code Scan**): A. In the **.NET framework version** box, specify the .NET framework version used to compile the code. B. In the **Libdirs** box, specify a semicolon-separated list of directories where referenced system or third-party DLLs are located. C. In the **Fortify SCA translation options** box, specify any additional Fortify Static Code Analyzer translation options. See the *Micro Focus Fortify Static Code Analyzer User Guide* for detailed information about the available translation options. D. In the **Source files** box, specify the source files to translate.
Maven 3	i. 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 located 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.

Application Type	Description
Gradle	i. To use a Wrapper, select **Use Gradle Wrapper**. ii. In the **Gradle tasks** box, type the Gradle tasks required for your project. iii. In the **Gradle options** box, type the Gradle options required for your project.
Other	This is very similar to the advanced configuration. You must manually provide all the Fortify Static Code Analyzer translation options in the **Fortify SCA translation options** box. See the *Micro Focus Fortify Static Code Analyzer User Guide* for detailed information about the available translation options. Specify the source code to scan in the **Includes list** box.

b. (Optional) Enable the debug or verbose logging options.

c. (Optional) Specify a custom location for the Fortify Static Code Analyzer log file, specify 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`.

10. 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 (XML files).

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) Specify a custom location for the Fortify Static Code Analyzer log file, specify 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`.

11. 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 **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 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 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 performed in Fortify Software Security Center.

12. 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.

The available Pipeline steps match the actions provided in the Fortify Jenkins Plugin project build steps. The following table lists each of the available Fortify Jenkins Plugin Pipeline steps. Each section describes the parameters and contains examples.

Project Build Step	**Pipeline Step**
Update Fortify Security Content	"fortifyUpdate Step" on page 20

Project Build Step	Pipeline Step
Run Fortify SCA clean	"fortifyClean Step" on page 20
Run Fortify SCA translation	"fortifyTranslate Step" on page 21
Run Fortify SCA scan	"fortifyScan Step" on page 26
Upload Fortify SCA scan results to Fortify Software Security Center	"fortifyUpload Step" on page 27

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',
      proxyURL: 'proxy.mycorp.net:8080', useProxy: true,
      proxyUsername: 'admin', proxyPassword: 'pw123'
  }
  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: '1.8')
  }
  stage('Fortify Scan') {
    fortifyScan buildID: 'MyJavaApp', resultsFile: 'MyJavaApp.fpr'
      customRulepacks: 'MyRules.xml', logFile: 'MyJavaApp-scan.log'
  }
  stage('Fortify Upload') {
    fortifyUpload appName: 'MyJavaApp', appVersion: '3',
      resultsFile: 'MyJavaApp.fpr'
  }
}
```

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

```
pipeline {
  agent any
  stages {
    stage('Fortify Update') {
      steps {
        fortifyUpdate  updateServerURL: 'https://update.fortify.com',
          proxyURL: 'proxy.mycorp.net:8080', useProxy: true,
          proxyUsername: 'admin', proxyPassword: 'pw123'
      }
    }
    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: '1.8')
      }
    }
    stage('Fortify Scan') {
      steps {
        fortifyScan buildID: 'MyJavaApp',
          resultsFile: 'MyJavaApp.fpr'
          customRulepacks: 'MyRules.xml',
          logFile: 'MyJavaApp-scan.log'
       }
     }
     stage('Fortify Upload') {
       steps {
          fortifyUpload appName: 'MyJavaApp', 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`
proxyURL	Optional (String). Specifies a URL for the proxy server. Use the following format: *`<address>`*`:`*`<port_number>`*.	(none)
proxyUsername	Optional (String). Specifies a user name for the proxy connection.	(none)
proxyPassword	Optional (String). Specifies a password for the proxy connection.	(none)
useProxy	Optional (boolean). Specifies whether or not to use a proxy to connect to the Fortify Update server.	false

fortifyUpdate Example

The following example updates the Fortify security content using a proxy for the connection:

```
node {
  stage('Fortify Update') {
    fortifyUpdate  proxyURL: 'proxy.mycorp.net:8080',
    proxyUsername: 'admin', proxyPassword: 'pw123', useProxy: true
  }
}
```

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.	

Parameter	Description	Default Value
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 or not 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.

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.

Parameter	Description	Default Value
General parameters		
buildID	Required (String). A unique identifier for the scan.	

Parameter	Description	Default Value
projectScanType	Required. (String). The project scan type is one of the following:fortifyAdvanced, fortifyDevenv, fortifyDotnetSrc, fortifyGradle, fortifyJava, fortifyMaven3, fortifyMSBuild, or fortifyOther.	
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 or not 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.
excludeList	Optional (String). Specifies a list of directories or files to exclude from translation.	(none)
Java parameters		

Parameter	**Description**	**Default Value**
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 *Micro Focus Fortify Static Code Analyzer User Guide* for specific version information.
javaClasspath	Optional (String). Specifies the class path as 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)
devenv / MSBuild parameters		
dotnetProject	Required (String). Specifies a solution (.sln) or a project file (.proj) file.	
dotnetAddOptions	Optional (String). Specifies any additional Fortify Static Code Analyzer options for translating .NET code.	(none)
DotnetSrc parameters		
dotnetFrameworkVersion	Required (int). Specifies the .NET framework version.	
dotnetSrcFiles	Required (String). Specifies the location of the .NET source files.	

Parameter	Description	Default Value
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)
Maven3 parameters		
mavenOptions	Optional (String). Specifies any additional Maven options required for your project.	(none)
Gradle parameters		
gradleTasks	Required (String). Specifies the Gradle tasks required for your project.	
useWrapper	Optional (boolean). Specifies whether or not to use a Wrapper.	false
gradleOptions	Optional (String). Specifies any additional Gradle options required for your project.	(none)
Other parameters		
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)

Parameter	Description	Default Value
Advanced parameters		
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(), fortifyOther().

The following is a basic example to translate a Java project and exclude 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')
  }
}
```

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 or not 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 `sca.log` and the default location is in 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 the Fortify Jenkins Plugin" on page 12). After an upload you can view results the results in Jenkins (see "Viewing Analysis Results" on page 29).

Parameter	Description	Default Value
appName	Required (String). Specifies the application name to store the results in Fortify Software Security Center.	
appVersion	Required (String). Specifies the application version 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

Parameter	Description	Default Value
		"./**/*.fpr" in the workspace for the FPR 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 failure. For example, `[fortify priority order]:critical`.	(none)
pollingInterval	Optional (int). Specifies the interval (in minutes) that the Fortify Jenkins Plugin polls Fortify Software Security Center to determine if the FPR processing is complete. The valid values are 0-60. **Important!** If the FPR processing requires approval, then this step will not complete until the approval is performed in Fortify Software Security Center.	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

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.

Security Vulnerability Graph for Your Project

The project page displays a Normalized Vulnerability Score (NVS) graph. 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:

```
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 very useful. For example, if Application A has 0 critical issues and 10 low issues, the total issue count is 10. If Application B has five critical issues and no low issues, the total issue count is 5. These values might mislead you to think that Application B is better than Application A, when it is not.

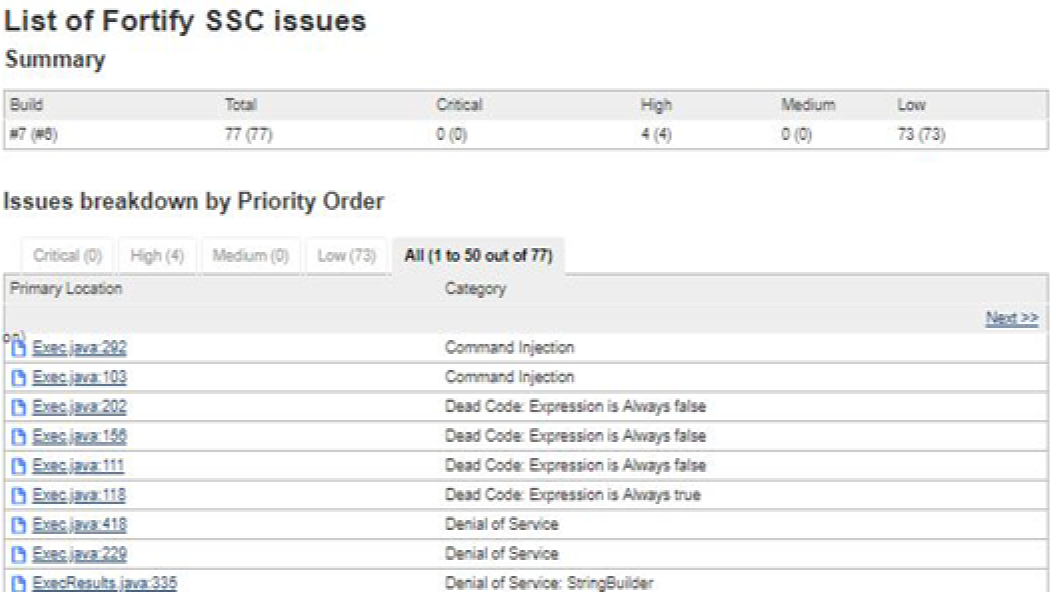
The NVS calculated for the two example applications provides a different picture (simplified equation):

- 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 Micro Focus Fortify Software Security Center, open your project 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 (#6)	77 (77)	0 (0)	4 (4)	0 (0)	73 (73)

Issues breakdown by Priority Order

Primary Location	Category
Exec.java:292	Command Injection
Exec.java:103	Command Injection
Exec.java:202	Dead Code: Expression is Always false
Exec.java:156	Dead Code: Expression is Always false
Exec.java:111	Dead Code: Expression is Always false
Exec.java:118	Dead Code: Expression is Always true
Exec.java:418	Denial of Service
Exec.java:229	Denial of Service
ExecResults.java:335	Denial of Service: StringBuilder

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 too long, 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 do 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, click **Advanced Settings**, and then change the value in the **Issue breakdown page size** box.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on User Guide (Fortify Jenkins Plugin 19.1.0)

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.

We appreciate your feedback!