Micro Focus
Fortify ScanCentral SAST

Software Version: 23.1.0

Installation, Configuration, and Usage Guide

Document Release Date: May 2023
Software Release Date: May 2023

Legal Notices

Open Text Corporation

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

Copyright Notice

Trademark Notices

Documentation Updates

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

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

This document was produced on April 12, 2023. To check for recent updates or to verify that you are using the most recent edition of a document, go to:

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

Contents

Preface

Contacting 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/cyberres/application-security

About the Documentation Set

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

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

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

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

Fortify Product Feature Videos

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

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

Change Log

The following table lists changes made to this document.

Software Release / Document Version	Changes

23.1.0	This document no longer contains a What's New in ScanCentral SAST topic.
	Added
	"Securing Tomcat Server" on page 18
	"Configuring Where job Files and the worker-persistence.properties File are Generated" on page 47
	Updated
	• Two new properties (`client_zip_location`, `ssc_restapi_connect_timeout` and `ssc_restapi_read_timeout`) are described in "Configuring the ScanCentral SAST Controller" on page 22. The `lim_proxy_server`, `remote_ip_proxy_header`, and `ssc_trusted_proxies_remote_ip` properties were removed.
	• "Avoiding Timeout Errors" on page 48 was rewritten to describe the new procedures for configuring timeout between the Controller and sensors, between the Controller and clients, and between the Controller and Fortify Software Security Center.
	• Information about auto-detection of the build tool was added to "Offloading Both Translation and Scanning" on page 72.
	• In "Fortify ScanCentral SAST Command-Line Options" on page 88:
	  • The `-block-until` start option was added to the `status` command options.
	  • The `-bto, --block-timeout` and `-pi, --poll-interval` options were added to the `retrieve` and `status` command options.
	  • The "Options Accepted for -targs" and "Options Accepted for -sargs" sections were added.

Software Release / Document Version	Changes
	Removed What's New in Micro Focus ScanCentral SAST 22.2.0 Configuring the Logging Level for Sensors
22.2.0	**Added** • What's New in Micro Focus ScanCentral SAST 22.2.0 • "Configuring Proxies for Fortify ScanCentral SAST Clients" on page 61 • "Submitting Scan Requests and Uploading Results to Fortify Software Security Center" on page 77 contains the new section "Specifying the Name of FPR Files Uploaded to Fortify Software Security Center" on page 78. **Updated** • The `allow_insecure_clients_with_empty_token` property was removed from the list of Controller properties in "Configuring the ScanCentral SAST Controller" on page 22. • The sections "Securing the Controller for Authorized Client Use Only" and "Allowing CloudScan Clients that do not Support Client Authentication to Connect to the Controller" were removed. Now, the Controller accepts only authorized clients. • The new section "Configuring the Java Memory for the Service" on page 20 was added to "About the Controller" on page 19. • Strings in the `log4j2.xml` file used to configure the logging level were modified in "Configuring the Logging Level on the Controller" on page 39. • A note regarding working with Java 8 projects was added to "Creating a Standalone Client" on page 58. • Changed the topic heading from "Viewing Client and Sensor Logs" to "Viewing ScanCentral Logs" on page 63. • The cautionary note in "About Upgrading ScanCentral SAST Components" on page 64 was revised.

Software Release / Document Version	Changes

	• The procedure used to upgrade the Controller was modified in "Upgrading the ScanCentral SAST Controller" on page 65.
	• "Enabling and Disabling Auto-Updates of Clients and Sensors" on page 68 contains revised version numbers.
	• "Fortify Static Code Analyzer Mobile Build Session Version Compatibility" on page 70 contains revised version numbers.
	• "Submitting Scan Requests" on page 71 includes the new section "Working with Java 8 Projects" on page 77.
	• "Fortify ScanCentral SAST Command-Line Options" on page 88 contains the following changes:
	  • Links to each options section were added.
	  • The `-application-version<id>` option was removed from "Start Command" on page 90.
	  • The new `-fprssc`, `--fpr-filename-on-ssc <file>` and `-versionid`, `--application-version-id <id>` options were added to "Start Command" on page 90.
	  • A cautionary note related to file paths that Include an umlaut was added to "Package Command" on page 101.
	  • (Fortify on Demand only) The `-oss` packaging option was added to "Package Command" on page 101.
	  • The description of the "Update Command" on page 106 was revised.
	Removed
	• What's New in Micro Focus ScanCentral SAST 22.1.0

22.1.0 / Revision 2 - August 25, 2022	• Fixed incorrect headings in the Contents table. • Changes were made to the `Start` command options in "Fortify ScanCentral SAST Command-Line Options" on page 88.
22.1.0	**Added** What's New in Micro Focus ScanCentral SAST 22.1.0 **Updated**

Software Release / Document Version	Changes
	• Information about creating a standalone client and instructions on how to create multiple standalone clients of different supported versions in the Controller was added to "Creating ScanCentral SAST Sensors" on page 42. • The new `update` command was added to "Fortify ScanCentral SAST Command-Line Options" on page 88. **Removed** • What's New in Micro Focus ScanCentral SAST 21.2.0

Chapter 1: Introduction

With Fortify ScanCentral SAST (ScanCentral), Fortify Static Code Analyzer users can better manage their resources by offloading code analysis tasks from their build machines to a cloud of machines (sensors) provided for this purpose. Its simple-to-use interface enables integration of static analysis with the build process and provides the ability to dynamically scale the sensors needed to perform the work required of the CI/CD pipeline with respect to running scans.

You can start a Fortify Static Code Analyzer analysis of your code from a ScanCentral client in one of two ways:

- You can perform the translation phase on a local or build machine to generate a mobile build session (MBS). The ScanCentral client then hands off the MBS to the ScanCentral Controller, which distributes the MBS to the sensors. The sensors then perform the scanning phase of the analysis.
- If your application version is written in a language supported for remote translation, you can also offload the translation phase of the analysis to your sensors. For information about the languages supported for offloading translation, see "Installing ScanCentral SAST Clients" on page 57. For information about the specific language versions supported, see the *Fortify Software System Requirements* document.

If your code is written using a language other than one supported for offloading project translation, the translation phase (less processor- and time-intensive than the scanning phase) is completed on the build machine. After translation is completed, ScanCentral generates a project package, which it then moves to a distributed cloud of machines (sensors) for scanning. In addition to freeing up build machines, this process makes it easy to add more resources to the cloud and grow the system as needed, without having to interrupt your build process. And, the ScanCentral Controller can direct the output FPR to Fortify Software Security Center.

This content provides information on how to install, configure, and use ScanCentral to streamline your static code analysis process.

Related Documents

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

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

All Products

The following documents provide general information for all products. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website.

Document / File Name	Description
About Fortify Product Software Documentation About_Fortify_Docs_<*version*>.pdf	This paper provides information about how to access Fortify product documentation. **Note:** This document is included only with the product download.
Fortify License and Infrastructure Manager Installation and Usage Guide LIM_Guide_<*version*>.pdf	This document describes how to install, configure, and use the Fortify License and Infrastructure Manager (LIM), which is available for installation on a local Windows server and as a container image on the Docker platform.
Fortify Software System Requirements Fortify_Sys_Reqs_<*version*>.pdf	This document provides the details about the environments and products supported for this version of Fortify Software.
Fortify Software Release Notes FortifySW_RN_<*version*>.pdf	This document provides an overview of the changes made to Fortify Software for this release and important information not included elsewhere in the product documentation.
What's New in Fortify Software <version> Fortify_Whats_New_<*version*>.pdf	This document describes the new features in Fortify Software products.

Fortify Software Security Center

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

Document / File Name	Description
Fortify Software Security Center User Guide	This document provides Fortify Software Security Center users with detailed information about how to deploy and use

Document / File Name	Description
SSC_Guide_<*version*>.pdf	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.

Fortify Static Code Analyzer

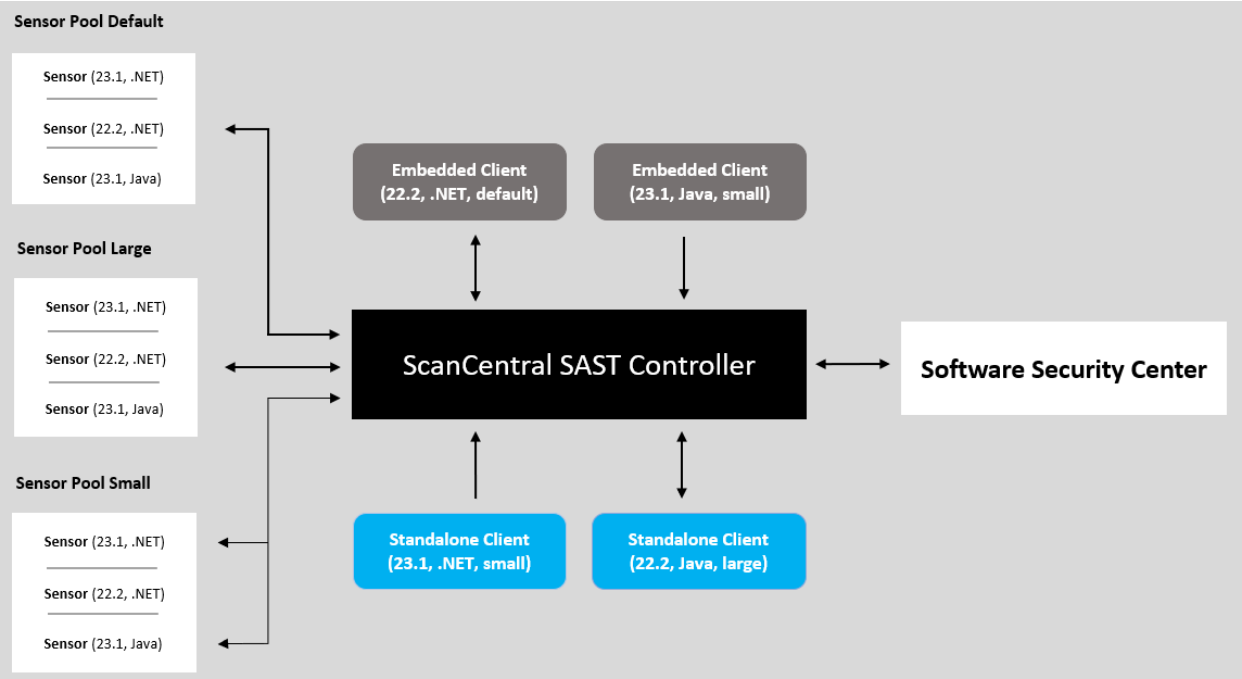
The following documents provide information about Fortify Static Code Analyzer. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-static-code.

Document / File Name	Description
Fortify Static Code Analyzer User Guide SCA_Guide_<*version*>.pdf	This document describes how to install and use Fortify Static Code Analyzer to scan code on many of the major programming platforms. It is intended for people responsible for security audits and secure coding.
Fortify Static Code Analyzer Applications and Tools Guide SCA_Apps_Tools_<*version*>.pdf	This document describes how to install Fortify Static Code Analyzer applications and tools. It provides an overview of the applications and command-line tools that enable you to scan your code with Fortify Static Code Analyzer, review analysis results, work with analysis results files, and more.
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.
Fortify Audit Workbench User Guide AWB_Guide_<*version*>.pdf	This document describes how to use Fortify Audit Workbench to scan software projects and audit analysis

Document / File Name	Description
	results. This guide also includes how to integrate with bug trackers, produce reports, and perform collaborative auditing.
Fortify Plugin for Eclipse User Guide Eclipse_Plugin_Guide_<*version*>.pdf	This document provides information about how to install and use the Fortify Complete Plugin for Eclipse.
Fortify Analysis Plugin for IntelliJ IDEA and Android Studio User Guide IntelliJ_AnalysisPlugin_Guide_<*version*>.pdf	This document describes how to install and use Fortify Analysis Plugin for IntelliJ IDEA and Android Studio.
Fortify Extension for Visual Studio User Guide VS_Ext_Guide_<*version*>.pdf	This document provides information about how to install and use the Fortify extension for Visual Studio to analyze, audit, and remediate your code to resolve security-related issues in solutions and projects.
Fortify Static Code Analyzer Applications and Tools Properties Reference Guide SCA_Tools_Props_Ref_<*version*>.pdf	This document describes the properties used by Fortify Static Code Analyzer applications and command-line tools.

Fortify ScanCentral SAST Components

The following diagram illustrates a Fortify ScanCentral SAST environment.



A Fortify ScanCentral SAST deployment includes the following three components:

Note: The minimum deployment requires three physical or virtual machines: a Fortify ScanCentral SAST client, a sensor, and a Controller. A Fortify Software Security Center server is optional.

- **ScanCentral Controller**: A standalone web application that receives the Fortify Static Code Analyzer mobile build sessions (MBS) and scan instructions from ScanCentral SAST clients (or project packages with translation and scan instructions), routes the information to sensors, and (optionally) uploads scan results (FPR files) to Fortify Software Security Center. For more detail, see "About the Controller" on page 19.
- **ScanCentral SAST client**: A build machine on which Fortify Static Code Analyzer translates code and generates Fortify Static Code Analyzer mobile build sessions (MBS). The translated source code, along with optional and required data, such as custom rules and Fortify Static Code Analyzer command-line arguments, are uploaded to the ScanCentral Controller. Clients can also generate packages for remote translation, independent of Fortify Static Code Analyzer. For more detail, see "About Clients" on page 57.
- **ScanCentral sensors:** Distributed network of computers set up to receive scan requests and analyze code using Fortify Static Code Analyzer. A sensor accepts either a mobile build session (MBS) file and performs a scan, or it accepts a prepared package that contains sources and dependencies, which it translates and scans. For more detail, see "About Sensors" on page 42.

- **Sensor pools:** To scan code, sensors must belong to a sensor pool. A sensor pool consists of one or more sensors, grouped based on any criteria, which you can then target for scan requests. Example: You can create a sensor pool that consists of machines with a lot of physical memory to use for scan requests that require a lot of memory. If you do not specifically add a sensor to a sensor pool, it is automatically assigned to the default sensor pool.

To successfully deploy Fortify ScanCentral SAST, in addition to installing Fortify Static Code Analyzer, complete the following tasks in the order listed here:

- (Recommended, but not required) Deploy a (or connect to an existing) Fortify Software Security Center instance
- Install the Fortify ScanCentral SAST Controller
- Create ScanCentral SAST sensors

Instructions for completing these tasks are provided in the following sections. For information about hardware and software requirements for these components, see the *Fortify Software System Requirements* document.

Securing ScanCentral SAST Deployment

The Micro Focus Fortify family of products collects and displays information about an enterprise's applications. That information includes summaries of the potential security vulnerabilities uncovered in the source code.

Just as you apply security precautions to your applications, you must also secure access to the ScanCentral SAST components. The security vulnerability summaries that Fortify products provide may mandate an even higher level of secure deployment.

ScanCentral SAST works with your code base. Because this information offers various opportunities for mishandling or abuse, Fortify recommends that you deploy ScanCentral SAST in a secure operations facility and secure access to ScanCentral SAST installation directories.

Securing Tomcat Server

You must ensure the operational security of Tomcat Server. At a minimum, configure Tomcat Server to use HTTPS in conjunction with an SSL certificate issued by a trusted certificate authority. Fortify also recommends that you use only strong cipher suites with Tomcat. Finally, take any additional steps necessary to secure Tomcat Server in your operating environment.

Using More Secure Cipher Suites

Fortify recommends that you disable weak SSL/TLS cipher suites in Tomcat in favor of more secure suites.

APR-based SSL Connections

If you use an APR-based SSL connection, use the SSLCipherSuite directive. For detailed information, see https://httpd.apache.org/docs/2.4/mod/mod_ssl.html#sslciphersuite and Cipher Suites and Enforcing Strong Security (https://httpd.apache.org/docs/current/ssl/ssl_howto.html).

JSSE-based SSL Connections

If you use a JSSE-based SSL connection, use the `ciphers` and the `honorCipherOrder` attributes. For details, see Apache Tomcat 9 Configuration Reference - The HTTP Connector.

Because of trade-offs between improved security and improved interoperability, better performance, and so on, there is no correct cipher suite choice. However, Apache provides information that can help you make your choice (see https://cwiki.apache.org/confluence/display/TOMCAT/Ciphers).

Chapter 2: About the Controller

The ScanCentral SAST Controller (Controller) is a standalone server that sits between the ScanCentral SAST clients, sensors, and optionally, Fortify Software Security Center. The Controller accepts scan requests issued by the clients and passes them on to an available sensor. A sensor returns scan results to the Controller, which stores them temporarily.

Installing the Controller

Important! Before you install the Controller, you must first download and configure a Java Runtime Environment (JRE). For information about supported JRE versions, see the *Fortify Software System Requirements* guide. For information about how to download and configure a JRE, see the documentation for the supported JRE version.

Caution! The name of the directory into which you install the Controller must not include spaces.

To install the Controller (on a Linux or Windows system):

- Extract the contents of the `Fortify_ScanCentral_Controller_<version>_x64.zip` file to a directory of your choosing.

Note: In this document, *<controller_dir>* refers to the ScanCentral Controller installation directory, *<sca_install_dir>* refers to the Fortify Static Code Analyzer installation directory, and *<ssc_install_dir>* refers to the Fortify Software Security Center server installation directory.

After you install the Controller, *<controller_dir>* resembles the following:

```
bin/

db-migrate/

tomcat/

readme.txt
```

Installing the Controller as a Service

To install the Controller as a service on a machine without other Tomcat instances running:

1. Log on to Windows as a local user with administrator privileges.
2. Check to make sure that the `JRE_HOME` and `JAVA_HOME` environment variables are correctly configured.
3. Check to make sure that the `CATALINA_HOME` environment variable is either empty or set up to point to the ScanCentral SAST `Tomcat` directory.
4. Navigate to the `<controller_dir>/tomcat/bin` directory, and then run the following:

```
service.bat install
```

This creates a service with the name "Tomcat9."

To install the Controller as a service with a different name:

1. Check to make sure that the `JRE_HOME` and `JAVA_HOME` environment variables are correctly configured.
2. Check to make sure that the `CATALINA_HOME` environment variable is either empty or set up to point to the ScanCentral SAST `Tomcat` directory.
3. Navigate to the `<controller_dir>/tomcat/bin` directory, and then run the following:

```
service.bat install <service_name>
```

Important! The service name must not contain any spaces.

Configuring the Java Memory for the Service

To configure the Java memory for the Controller service:

1. Run `tomcat9w.exe`.
2. In the Apache Tomcat Properties window, select the **Java** tab, and then set the **Maximum memory pool** value.
3. Restart the service.

Uninstalling the Controller Service

To uninstall the Apache Tomcat 9.0 service:

1. Stop the service.
2. Navigate to the <*controller_dir*>/tomcat/bin directory, and then run the following:

```
service.bat remove
```

To uninstall the controller as a service with a name other than Apache Tomcat 9.0:

1. Stop the service.
2. Navigate to the <*controller_dir*>/tomcat/bin directory, and then run the following:

```
service.bat remove <service_name>
```

See Next

"Configuring the ScanCentral SAST Controller" on the next page

See Also

"Starting the ScanCentral SAST Controller" on page 37

For information about how to update your Controller, see "About Upgrading ScanCentral SAST Components" on page 64 and "Upgrading the ScanCentral SAST Controller" on page 65.

Configuring the ScanCentral SAST Controller

After you install the Controller, edit global properties such as the email address to be used, the shared secret for the Controller (password that Fortify Software Security Center uses when it requests data from the ScanCentral Controller), the shared secret for the clients, and the Fortify Software Security Center URL (if you plan to upload your FPRs to Fortify Software Security Center).

Caution! To avoid potential conflicts, Fortify recommends that you run the Controller on a Tomcat Server instance other than the instance that Fortify Software Security Center uses.

To configure the Controller:

1. Navigate to `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes`.
2. Open the `config.properties` file in a text editor, and then configure the properties listed in the following table.

Property	Description
`accept_job_when_no_sensor_available`	Determines whether scan requests submitted by clients are accepted if no compatible sensors (or compatible versions) are available. The default value is `true`. In the following examples, the option is set to `false`: • If version 20.2 clients submit a scan request, and only version 21.2 sensors are available, the scan request is rejected. • If a client submits a request for a scan of code written in .NET, and no .NET sensors are available, the scan is rejected.
`cleanup_period`	Frequency (in minutes) with which expired jobs and sensors are cleaned up. (The default is 60.)
`client_auth_token`	Specify a client authentication token string that contains no spaces or backslashes to secure the Controller for use by authorized clients only. If you prefer not to use plain text, you can use an encrypted shared secret as the value for this property. For instructions on how to encrypt a shared secret, see "Encrypting the Shared Secret" on page 30.
`client_auto_update`	If set to `true`, enables the Controller to automatically update all outdated sensors and clients. For details, see "Enabling and Disabling Auto-Updates of Clients and Sensors" on page 68.
`db_dir`	ScanCentral SAST database home directory

Property	Description
	`db_dir=${catalina.base}/cloudCtrlDb`
`client_zip_location`	Location of the directory that contains ScanCentral SAST client ZIP files. To enable remote upgrades of one or multiple client versions, place them in this folder. (You can use any ZIP file names.) `client_zip_location=${catalina.base}/client`

`email_allow_list`	Use this property to specify the list of email domains that the Controller can use to send notifications.
	Examples of valid values:
	`*@yourcompanyname.com`
	`*@*yourcompanyname.com`
	`a*@yourcompanyname.com`
	`name@yourcompanyname.com`
	To specify multiple values, you can use commas (s), colons (:), or semicolons (;) as delimiters.

`email_deny_list`	Use this property to specify the list of email domains that the Controller cannot use to send notifications.
	Examples of valid values:
	`*@yourcompanyname.com`
	`*@*yourcompanyname.com`
	`a*@yourcompanyname.com`
	`name@yourcompanyname.com`
	To specify multiple values, you can use commas (s), colons (:), or semicolons (;) as delimiters.

`fail_job_if_ssc_upload_data_invalid`	If ScanCentral SAST is configured to upload scan results to an application version in Fortify Software Security Center, and either the `ScanCentralCtrlToken` token has expired or the specified application version does not exist, scan jobs are run, but the upload to Fortify Software Security Center fails. (The default behavior.) If you set this option to `true`, before the Controller creates a job and assigns it to a sensor, it checks to make sure that the `ScanCentralCtrlToken` token has not expired, and that the application version exists in Fortify Software Security Center. The default value is

Property	Description
	`false`. If set to `true` and the `ScanCentralCtrlToken` token expires before a scan job is assigned to sensor, the scan does not run and the job fails.
`from_email`	Email address of the sender.
`job_expiry_delay`	Number of hours after a job finishes that the job becomes a candidate for cleanup. Cleanup removes the job directory, removes jobs from the database, and removes information about expired sensors from the database so that they are no longer displayed in Fortify Software Security Center. By default, jobs are deleted from the Controller 168 hours, or 7 days.
`job_file_dir`	Job storage directory.
`lim_license_pool`	Name of the LIM license pool.
`lim_license_pool_password`	Password for the LIM license pool. **Note:** You can either use a plain text password, or use the `pwtool_keys_file` option to encrypt this password. For information about how to encrypt your passwords, see "Encrypting the Shared Secret" on page 30.
`lim_proxy_url`	To access the LIM server when the sensor is behind a proxy, configure the proxy server.
`lim_proxy_user`	LIM proxy username If authentication is required for the LIM proxy server.
`lim_proxy_password`	Password for the LIM proxy user. **Note:** You can either use a plain text password, or use the `pwtool_keys_file` option to encrypt this password. For information about how to encrypt your passwords, see "Encrypting the Shared Secret" on page 30.
`lim_server_url`	URL for the License and Infrastructure Manager (LIM) server Root Web Site.
`max_upload_`	Maximum size (MB) of files that can be uploaded to the Controller from

Property	Description
size	clients or sensors (for example, log files, result files, job files).
pool_mapping_mode	Used to configure different modes for mapping scan requests to sensor pools. For information about the valid values for pool_mapping_mode, see "About the pool_mapping_mode Property" on page 28.
pwtool_keys_file	Path to a file with pwtool keys. If encoded passwords are used, this must point to a file with the pwtool keys used to encode the passwords. Otherwise you can comment it out. pwtool_keys_file=${catalina.base}/pwtool.keys
scan_timeout	Maximum amount of time (in minutes) sensors can process a scan job and be prevented from doing other jobs. After the specified number of minutes is reached, a scan job is cancelled. This setting is applied to all sensors associated with the Controller but can be overridden if the -sto command-line option is specified for a given job. For information about the -sto option, see "Setting the Maximum Run Time for Scans" on page 45 and "Global Options" on page 88
smtp_host	SMTP server host name.
smtp_port	SMTP server port number.
smtp_ssl	If set to true, the Controller uses SSL for connections to the SMTP server. Otherwise, it does not use SSL (default).
smtp_ssl_check_trust	If set to false, the SMTP server certificate is always trusted. Otherwise, the certificate trust is based on the certification path (the default).
smtp_ssl_check_server_identity	If set to false, STMP server identity is not checked. Otherwise, the Controller checks server identity, as specified by RFC 2595 (the default).
smtp_auth_user/ smtp_auth_pass	If your SMTP server requires authentication, uncomment both the smtp_auth_user and smtp_auth_pass properties and set their values. Otherwise, leave both properties commented. You can use either a plain text password or a password encoded using pwtool for smtp_auth_pass.
ssc_lockdown_mode	If set to true, ScanCentral SAST clients are forced to work with the

Property	Description
	ScanCentral Controller through Fortify Software Security Center. Jobs must be uploaded to a Fortify Software Security Center application version (a job cannot be started without the upload). In SSC lockdown mode, users cannot assign scans to specific sensor pools manually. Instead , the mapping configured on Fortify Software Security Center for the selected application version is applied. In SSC lockdown mode, you: • Cannot use the client command `-url` option, but must use the `-ssc_url` option with the `-ssc_token` option instead • Must specify the application name and version, or the application version id, and the `-upload` option when starting the scan • Cannot specify the `-pool` option, because the job is assigned to the pool configured for the specified application version
`ssc_scancentral_ctrl_secret`	Password that Fortify Software Security Center uses to request data from the Controller. Specify a string that contains no spaces or backslashes. (Optional) Use an encrypted shared secret. For instructions on how to encrypt a shared secret, see "Encrypting the Shared Secret" on page 30. **Note:** The `ssc_cloudctrl_secret` option is supported for backward compatibility with Fortify CloudScan.
`ssc_remote_ip`	Remote IP address You can configure an allowed remote IP address for Fortify Software Security Center. Only requests with a matching remote IP address are allowed.
`ssc_remote_ip_header`	Remote IP HTTP header, where the Fortify Software Security Center remote IP is found if `ssc_remote_ip_trusted_proxies_range` is set. The default value is `X-FORWARDED-FOR`.
`ssc_remote_ip_trusted_proxies_range`	Remote IP range (in CIDR format) Set this if Fortify Software Security Center accesses the Controller via (reverse) proxy server. You can specify comma-separated IP addresses or CIDR network ranges. This is disabled by default, which means that `ssc_remote_ip_header` is

Property	Description
	never used to retrieve the remote IP address for Fortify Software Security Center.
`ssc_restapi_connect_timeout`	Used to specify the connection timeout between the Controller and Fortify Software Security Center (in milliseconds). The default is 10000. You can use this, and the `ssc_restapi_read_timeout` property, to solve timeout errors. (See "Avoiding Timeout Errors" on page 48.)
`ssc_restapi_read_timeout`	Used to specify the read timeout between the Controller and Fortify Software Security Center (in milliseconds). The default is 30000. You can use this, and the `ssc_restapi_read_timeout` property, to solve timeout errors. (See "Avoiding Timeout Errors" on page 48.)

`ssc_url`	URL for the Fortify Software Security Center server; all uploads are sent to this address.
	Examples:
	`https://<ssc_host>:<port>/ssc`
	`https://<ssc_host>:<port>/<context_path>`

`this_url`	URL for the Controller; used in emails to refer to this server for manual job result downloads. Example: `https://<controller_host>:8443/scancentral-ctrl`
`use_starttls`	Used to support STARTTLS. If set to `true`, use the STARTTLS protocol command (Opportunistic SSL/TLS) to inform the SMTP server that the email client wants to upgrade from an insecure connection to a secure connection using SSL/TLS. The default is `false`.
`worker_auth_token`	A string that contains no spaces or backslashes used to secure the Controller for use by authorized sensors only. If you prefer not to use plain text, you can use an encrypted shared secret as the value for this property. For instructions on how to encrypt a shared secret, see "Encrypting the Shared Secret on the Controller" on page 30.
`worker_expiry_delay`	Number of hours after a sensor stops communicating that it becomes a candidate for cleanup. (The default is 168 hours, or 7 days.)

Property	Description
worker_inactive_delay	Number of minutes after a sensor becomes inactive that all of its unfinished jobs are marked as faulted. Assign a value that is much larger than worker_stale_delay. Note that this option uses different time units than does worker_stale_delay.
worker_stale_delay	Number of seconds after a sensor stops communicating that it becomes stale. Assign a value that is larger than the worker_sleep_interval and worker_jobwatcher_interval defined for any sensor.

3. Save and close your config.properties file.
4. Start the Controller. (For instructions, see "Starting the ScanCentral SAST Sensors" on page 49.)

See Also

"Installing the Controller" on page 19

"Stopping the Controller" on page 38

"Placing the ScanCentral SAST Controller in Maintenance Mode" on page 40

"Configuring Job Cleanup Timing on Sensors" on page 85

About the pool_mapping_mode Property

The pool_mapping_mode property in the config.properties file determines how the Controller maps scan requests to sensor pools. Valid values for the pool_mapping_mode property are as follows:

- DISABLED— In this mode, a ScanCentral SAST client requests a specific sensor pool when it submits a scan request. Otherwise, the default pool is used. For details, see the following table.
- ENABLED— In this mode, if a scan request is associated with an application version in Fortify Software Security Center, the Controller queries Fortify Software Security Center to determine the sensor pool assigned to the application version. Or, a ScanCentral SAST client can request a specific sensor pool when it submits a scan request. (A client request for a specific sensor pool takes precedence over a query from the Controller.)

  Note: Sensors in the default sensor pool run scan requests that are not associated with an application version (and no specific pool is requested on the ScanCentral SAST client command line).

- ENFORCED—As with the ENABLED mode, if a scan request is associated with an application version in Fortify Software Security Center, the Controller queries Fortify Software Security Center for the sensor pool to use for the application version. Otherwise, the default sensor pool is targeted for scan requests. A client cannot request a specific sensor pool in the ENFORCED mode.

If `ssc_lockdown_mode` is enabled, then the value set for `pool_mapping_mode` in the `config.properties` file is ignored and `pool_mapping_mode` is automatically set to ENFORCED.

The following table shows how the Fortify Software Security Center integration with Fortify ScanCentral SAST responds to different input when `pool_mapping_mode` is set to `DISABLED`, `ENABLED`, or `ENFORCED`.

Note: By default, in enabled and enforced modes, all application versions are assigned to the Default pool.

INPUT	DISABLED	ENABLED	ENFORCED
No pool or version specified	Default sensor pool	Default sensor pool	Default sensor pool
Specific sensor pool (only) specified	Requested sensor pool	Requested sensor pool	Denied
Application version (only) specified	Default sensor pool	SSC-assigned pool	SSC-assigned pool
Invalid sensor pool (only) specified	Denied	Denied	Denied
Invalid application version (only) specified	Denied	Denied	Denied
Valid sensor pool and application version specified	Requested sensor pool	Requested sensor pool	Denied
Invalid sensor pool and valid application version specified	Denied	Denied	Denied
Valid sensor pool but invalid application version specified	Denied	Denied	Denied

See Also

"Configuring the ScanCentral SAST Controller" on page 22

Encrypting the Shared Secret

Passwords exist in the ScanCentral Controller and sensor configuration files as plain text. If you prefer to encrypt your passwords, you can.

You can use encrypted keys as values for:

- `worker_auth_token`, `smtp_auth_pass`, `ssc_scancentral_ctrl_secret`, `lim_license_pool_password`, `lim_proxy_password`, and `lim_proxy_user` properties in the `config.properties` file on the Controller
- `worker_auth_token` property in the `worker.properties` file on a sensor
- `client_auth_token` property in the `client.properties` file on a client

Encrypting the Shared Secret on the Controller

To encrypt a shared secret on the Controller:

1. Run one of the following:
   - On a Windows system, `<controller_dir>/bin/pwtool.bat <pwtool_keys_file>`
   - On a Linux system, `<controller_dir>/bin/pwtool <pwtool_keys_file>`
2. When prompted, type the password to encode, and then press **Enter**.

   Note: For the sake of security, make sure that the pwtool key file you use to encrypt secrets for sensors is different from the pwtool key file you use to encrypt secrets on the Controller.

   The pwtool generates a new key stored in the file on the path specified in step 1, or reuses an existing file on specified path.
3. Copy the new encrypted secret, and paste it as the value for one of the following properties in the `config.properties` file:
   - `worker_auth_token`
   - `smtp_auth_pass`
   - `ssc_scancentral_ctrl_secret`
   - `client_auth_token`

   Tip: Fortify recommends that you assign separate, unique shared secrets for the `worker_auth_token`, `smtp_auth_pass`, and `ssc_scancentral_ctrl_secret` properties.
4. Create two additional encrypted shared secrets (steps 1 and 2) and, in the `config.properties` file, paste these as values for the two properties to which you did not already assign an encrypted secret in step 3.
5. Uncomment the following line (property) in the `config.properties` file:

   `pwtool_keys_file=<pwtool_keys_file>`
6. Save the `config.properties` file.

Encrypting the Shared Secret on a Sensor

To encrypt a shared secret on a sensor:

1. Run one of the following:
   - On a Windows system, `<sca_install_dir>\bin\pwtool.bat <pwtool_keys_file>`
   - On a Linux system, `<sca_install_dir>/bin/pwtool <pwtool_keys_file>`
2. When prompted, type the password to encode, and then press **Enter**.

   The pwtool generates a new `pwtool.keys` file to `<pwtool_keys_file>` and prints a new encrypted secret to the console.
3. Copy the encrypted secret, and paste it as the value for `worker_auth_token` property in the `worker.properties` file.
4. Add the following line (property) to the `worker.properties` file:

   `pwtool_keys_file=<pwtool_keys_file>`

Encrypting the Shared Secret on a Client

To encrypt a shared secret on a client:

1. Run one of the following commands.
   - On a Windows system:
     - For a client used as part of Fortify Static Code Analyzer, run `<sca_install_dir>\bin\pwtool.bat <pwtool_keys_file>`
     - For a standalone client, run `<client_install_dir>\bin\pwtool.bat <pwtool_keys_file>`
   - On a Linux system:
     - For a client used as part of Fortify Static Code Analyzer, run `<sca_install_dir>/bin/pwtool <pwtool_keys_file>`
     - For a standalone client, run `<client_install_dir>/bin/pwtool <pwtool_keys_file>`
2. When prompted, type the password to encode, and then press **Enter**.

   The pwtool generates a new key in the file on the specified path, or reuses an existing file and prints the encrypted password.
3. Copy the new encrypted secret, and paste it as the value for the `client_auth_token` property in the `client.properties` file.
4. Add the following to the `client.properties` file:

   `pwtool_keys_file=<pwtool_keys_file>`

See Also

"Configuring the ScanCentral SAST Controller" on page 22

"Creating ScanCentral SAST Sensors" on page 42

Securing the Controller

The following procedure describes how to create a secure connection (HTTPS) between the ScanCentral SAST Controller/Tomcat server and ScanCentral SAST CLI. This procedure requires either a self-signed certificate or a certificate signed by a certificate authority such as VeriSign.

To create a secure connection (HTTPS) between the Controller/Tomcat server and ScanCentral CLI, use one of the following procedures.

Note: The following sections show *examples* of how to create a connection. For the most current information, see your Apache Tomcat documentation.

Creating a Secure Connection Using Self-Signed Certificates

To enable SSL on Tomcat using a self-signed certificate:

1. To generate a keystore that contains a self-signed certificate, open a command prompt and run the following `Java keytool` command:

   $JAVA_HOME/bin/keytool -genkey -alias *<alias_name>* -keyalg RSA -keystore *<mykeystore>*

2. Provide values for the prompts listed in the following table.

Prompt	Value
Enter keystore password:	Type a secure password.
Re-enter new password:	Re-type your secure password.
What is your first and last name?	Type your hostname. You can use your fully-qualified domain name here. **Note:** If you plan to provide an IP address as the hostname, then you must also provide the `-ext san=ip:<ip_address>` parameter to keytool. Without the `-ext san=ip:<ip_address>` parameter, the SSL handshake fails.
What is the name of your organizational unit?	Name to identify the group that is to use the cert.

Prompt	Value
What is the name of your organization?	Name of your organization.
What is the name of your City or Locality?	City or locality in which your organization is located.
What is the name of your State or Province?	State or province in which your organization is located.
What is the two-letter country code for this unit?	If your server is located in the United States, type **US**.
Confirm your entries:	Type yes to confirm your entries.
Enter key password for <tomcat><Return if same as keystore password>:	Password for your Tomcat server key. Press **Return / Enter** to use the same password you established for your keystore. (Fortify recommends that you create a new key password.)
Re-enter new password:	Re-type your key password.

3. To export the certificate from the Tomcat keystore, open a command prompt and type one of the following:

   On a Windows system:

   ```
   %JAVA_HOME/bin/keytool -export -alias <alias_name> -keystore
   <mykeystore> -file YourCertFile.cer
   ```

   On a Linux system:

   ```
   $JAVA_HOME/bin/keytool -export -alias <alias_name> -keystore
   <mykeystore> -file YourCertFile.cer
   ```

4. Add the following connector to the `server.xml` file in the `tomcat/conf` directory:

   ```
   <Connector port="8443" maxThreads="200"
   scheme="https" secure="true" SSLEnabled="true"
   keystoreFile="<mykeystore>" "keystorePass="<mypassword>"
   clientAuth="false" sslProtocol="TLS"/>
   ```

   Note: The default `server.xml` file installed with Tomcat includes an example `<connector>` element for an SSL connector.

5. Navigate to the following directory, and then open the `config.properties` file in a text editor:
   <controller_dir>`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes`

6. Update the `this_url` property, with your https address and port.
   Example: `this_url=https://<controller_host>:8443/scancentral-ctrl`
7. Restart your Tomcat server.
8. Set up your clients and sensors. For information about how to set up the ScanCentral SAST clients and sensors, see "Installing ScanCentral SAST Clients" on page 57 and, "Creating ScanCentral SAST Sensors" on page 42, respectively.
9. Add your self-signed certificate to the java keystore on all entities that communicate with the Controller (includes all clients, sensors, and Fortify Software Security Center installations) as follows:
   a. For ScanCentral SASTclients and sensors, open a command prompt and type the following:

      ```
      cd <sca_install_dir>\jre\bin
      ```

      Where *<sca_install_dir>* is the directory where the sensor or client is installed.
      For a installation or for standalone ScanCentral SAST clients, open a command prompt and type one of the following:
      - On Windows:

        ```
        cd %JAVA_HOME%\jre\bin
        ```

      - On Linux:

        ```
        cd $JAVA_HOME/jre/bin
        ```

   b. Run the following command:

      ```
      keytool -importcert -alias <aliasName> -keystore
      ..\lib\security\cacerts -file YourCertFile.cacerts -file
      YourCertFile.cer –trustcacerts
      ```

      Where `YourCertFile.cer` is the same certificate file that you exported in step 1.

Creating a Secure Connection Using a Certificate Signed by a Certificate Signing Authority

To enable SSL on Tomcat using a certificate signed by a certificate signing authority:

1. Use the Java keytool to generate a new keystore containing a self-signed certificate, as follows:
   - On a Windows system:

```
%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA -keystore
"<mykeystore>"
```

   - On a Linux system:

```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore
"<mykeystore>"
```

2. The keytool prompts you for the information described in the following table.

Prompt	Data
Enter keystore password:	Type a secure password.
Re-enter new password:	Re-enter your secure password.
What is your first and last name?	Type your hostname. You can use your fully qualified domain name here. **Note:** If you plan to enter an IP address as the hostname, then you will also need to pass an additional parameter to keytool, `-ext san=ip:<ipaddress>`. Without this additional parameter, the SSL handshake fails.
What is the name of your organizational unit?	Type the name of the group that is to use the certificate. (This can be anything you want.)
What is the name of your organization?	Type the name of your organization (This can be anything you want.)
What is the name of your City or Locality?	Type the city or locality. (This can be anything you want.)
What is the name of your	Type the state or province. (This can be anything you want.)

Prompt	Data
State or Province?	
What is the two-letter country code for this unit?	If your server is located in the United States, type **US**.
Confirm your entries:	Type **yes** to confirm your entries.
Enter key password for `<tomcat><Return if same as keystore password>`:	Type a password for your Tomcat server key, or press **Return** to use the same password you established for your keystore. Fortify recommends that you create a new password.
Re-enter new password:	Re-type your key password.

3. Generate a Certificate Signing Request (CSR).

   To obtain a certificate from a certificate signing authority, you must generate a Certificate Signing Request (CSR). The certificate authority uses the CSR to create the certificate. Create the CSR as follows:

   On a Windows system:

```
 %JAVA_HOME%\bin\keytool -certreq -alias <alias_name> -keyalg RSA -file
"yourCSRname.csr" -keystore "<mykeystore>"
```

   - On a Linux system:

```
 $JAVA_HOME/bin/keytool -certreq -alias <alias_name> -keyalg RSA -file
"yourCSRname.csr" -keystore "<mykeystore>"
```

4. Send the CSR file to the certificate signing authority you have chosen.
5. Once you receive your certificate from the certificate signing authority, import it into the keystore that you created, as follows:
   - On a Windows system:

```
 %JAVA_HOME%\bin\keytool -importcert -alias  <alias_name> -trustcacerts -file
"YourVerisignCert.crt"-keystore "<mykeystore>"
```

   - On a Linux system:

```
$JAVA_HOME/bin/keytool -importcert -alias  <alias_name> -trustcacerts -file
"YourVerisignCert.crt" -keystore "<mykeystore>"
```

   The root CA already exists in the `cacerts` file of your JDK, so you are just installing the intermediate CA for your certificate signing authority.

Note: If you purchased your certificate from VeriSign, you must first import the chain certificate. You can find the specific chain certificate on the VeriSign website or click the link for the chain certificate in the email you received from VeriSign with your certificate.

- On a Windows system:

```
%JAVA_HOME%\bin\keytool -importcert -alias IntermediateCA -
trustcacerts-file "chainCert.crt" -keystore "<mykeystore>"
```

- On a Linux system:

```
$JAVA_HOME/bin/keytool -importcert -alias IntermediateCA -
trustcacerts -file "chainCert.crt" -keystore "<mykeystore>"
```

6. Add the following connector to the `server.xml` file in the `tomcat\config` directory:

```
<Connector port="8443" maxThreads="200"

scheme="https" secure="true" SSLEnabled="true"

keystoreFile="<mykeystore>" keystorePass="<mypassword>"

clientAuth="false" sslProtocol="TLS"/>
```

Note: An example `<Connector>` element for an SSL connector is included in the default `server.xml` file installed with Tomcat.

7. Restart Tomcat Server.
8. Navigate to the following directory, and then open the `config.properties` in a text editor:

```
<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes
```

9. Update the `this_url` property with your https address and port.

Example: `this_url=https://<controller_host>:8443/scancentral-ctrl`

Starting the ScanCentral SAST Controller

You can start the Controller manually or set it to start automatically, as a service.

Starting the Controller Manually

To start the Controller:

1. If you plan to upload your scan results to Fortify Software Security Center, check to make sure that the Fortify Software Security Center instance is running.

2. On the machine that hosts the Controller, navigate to the Tomcat <bin> directory:
   - On a Windows system: `cd <controller_dir>\tomcat\bin`
   - On a Linux system: `cd <controller_dir>/tomcat/bin`
3. Run one of the following commands:
   - On a Windows system, run `startup.bat`.

     Note: If Tomcat is running as a service, rather than running `start.bat`, you can just start the service.
   - On a Linux system, run `./startup.sh`.

For Information about how to start the Controller automatically, see "Installing the Controller as a Service" on page 20.

See Also

"Placing the ScanCentral SAST Controller in Maintenance Mode" on page 40

Stopping the Controller

You can stop the Controller immediately using the following procedure. However, Fortify strongly recommends that you first place the Controller in maintenance mode to preserve any scans that are running. (See "Placing the ScanCentral SAST Controller in Maintenance Mode" on page 40.)

To stop the ScanCentral SAST Controller:

1. On the machine where the Controller is installed, navigate to the Tomcat `bin` directory:
   - On a Windows system: `cd <controller_dir>\tomcat\bin`
   - On a Linux system: `cd <controller_dir>/tomcat/bin`
2. Type one of the following commands:
   On a Windows system:

   ```
   shutdown.bat
   ```

   On a Linux system:

   ```
   ./shutdown.sh
   ```

See Also

"Placing the ScanCentral SAST Controller in Maintenance Mode" on page 40

"Removing the ScanCentral SAST Controller from Maintenance Mode" on page 41

"Safely Shutting Down Sensors" on page 55

Troubleshooting the Controller

Following are some of the issues you might encounter working with the Controller, and suggestions on how to address them.

After upgrading the binaries on the local server for the Controller, you can access the Controller using the address `https://host:port/scancentral-ctrl/`, but you cannot access it from the workstation. Also, while trying to integrate Fortify Software Security Center with the Controller, the Controller status is not visible, even though the `config.properties` file was updated with the required details.

On your client machines, go to the `Core/config` directory and check the `client.properties` file to make sure that the value set for the `client_auth_token` parameter matches the value for the same parameter in the `config.properties` file found in your Controller installation folder at `tomcat/webapps/scancentral-ctrl/WEB-INF/classes`.

Configuring the Logging Level on the Controller

ScanCentral SAST logging typically provides enough information to follow the flow of operations under normal conditions. If things are not working as expected, the logging may not provide enough information to determine the actual root cause of the issue.

In the event that ScanCentral SAST does not provide enough information to diagnose a situation, you can increase the amount of information that is logged. The following steps describe how to configure the logging level on the Controller.

To configure the logging level on the Controller:

1. Navigate to `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes`, and open the `log4j2.xml` file in a text editor.
2. Locate one of the following strings:
   - `<Logger name="com.fortify.cloudscan" level="info" additivity="false">`
   - `<Logger name="com.fortify.cloudscan.ctrl.service" level="info" additivity="false">`
3. For a more detailed level of logging, change the level, as follows:

   `<Logger name="com.fortify.cloudscan" level="debug" additivity="false">`

   Standard log levels supported by `log4j2.xml` are as follows.

Standard Level	intLevel
OFF	0

Standard Level	intLevel
FATAL	100
ERROR	200
WARN	300
INFO	400
DEBUG	500
TRACE	600
ALL	Integer.MAX_VALUE

4. To apply the change, restart the Controller.

For more information about defining custom log levels, see the Apache Logging Services website (https://logging.apache.org/log4j/2.x/manual/customloglevels.html).

See Also

"Enabling Debugging on Clients and Sensors" on page 63

Placing the ScanCentral SAST Controller in Maintenance Mode

An abrupt shutdown of the ScanCentral SAST Controller can result in the loss of scans already started on sensors. To prevent this from happening, place your Controller in maintenance mode. After you do, the Controller accepts no new job requests from clients and assigns no queued jobs to sensors.

After the Controller is placed in maintenance mode, sensors complete the scans they are currently running, but accept no new scans. After the Controller is back up and running, the sensors again become available.

Tip: If the Controller is in maintenance mode, you can manually shut down any sensor that is not running a scan.

The following procedure describes how to place the Controller in maintenance mode.

Important! To place the Controller in maintenance mode, the Controller must be version 21.2.0 or later.

To place the Controller in maintenance mode:

1. Log on to Fortify Software Security Center as an administrator, and then, on the Fortify header, click **SCANCENTRAL**.
2. In the left pane of the SAST page, select **Controller**.
3. Click **START MAINTENANCE MODE**.

The Controller receives the maintenance request from Fortify Software Security Center and, if any sensors are running scans, the Controller mode changes from ACTIVE to WAITING_FOR_JOB_COMPLETED. If no job is being processed, the mode changes directly from ACTIVE to MAINTENANCE. At this point, you can safely shut down the Controller.

See Also

"Starting the ScanCentral SAST Controller" on page 37

"Safely Shutting Down Sensors" on page 55

"Stopping the Controller" on page 38

Removing the ScanCentral SAST Controller from Maintenance Mode

To remove the Controller from maintenance mode:

1. Log on to Fortify Software Security Center as an administrator, and then, on the Fortify header, click **SCANCENTRAL**.
2. In the left pane of the SAST page, select **CONTROLLER**.
3. Click **END MAINTENANCE MODE**.

See Also

"Placing the ScanCentral SAST Controller in Maintenance Mode" on the previous page

"Stopping the Controller" on page 38

Chapter 3: About Sensors

ScanCentral SAST sensors are computers set up to receive scan requests and analyze code using Fortify Static Code Analyzer. A sensor accepts either a mobile build session (MBS) file and performs a scan, or it accepts a prepared package that contains sources and dependencies, which it translates and scans.

For MBS scans, ScanCentral SAST supports all languages that Fortify Static Code Analyzer supports. For remote translation and scans of the prepared packages, ScanCentral SAST supports only the languages that can be used to perform remote translation. For information about the languages supported for performing remote translation, see "Installing ScanCentral SAST Clients" on page 57.

Tip: As you set up your ScanCentral SAST environment, you can use subnets to segment your build machines from the sensors. The build machines need only communicate with the Controller, which in turn communicates with the sensors.

Creating ScanCentral SAST Sensors

To make it convenient for network administrators to isolate traffic to ScanCentral SAST sensors, Fortify recommends that you install sensors in a separate subnet. Use the sensors only as scan boxes. ScanCentral SAST supports only one sensor per machine.

Creating a Sensor Using Static Code Analyzer

The following procedure describes how to create a new sensor. For information about how to upgrade an existing sensor, see "Upgrading ScanCentral SAST Sensors" on page 66.

Note: If you use Windows, you can install the sensor as a Windows service. For instructions, see "Creating a ScanCentral SAST Sensor as a Service" on the next page.

To create a sensor:

1. Install Fortify Static Code Analyzer. (For instructions, see the *Micro Focus Fortify Static Code Analyzer User Guide*.)
2. Navigate to the `<sca_install_dir>/Core/config` directory, and open the `worker.properties` file in a text editor.
3. Add the following property to the `worker.properties` file:

   ```
   worker_auth_token=<value_set_in_controller_configuration>
   ```
4. Specify either a clear text password, or an encrypted shared secret (password the Controller uses to communicate with the sensor) as the `worker_auth_token` value. For information about how to generate an encrypted shared secret, see "Encrypting the Shared Secret on a Sensor" on

page 31.

5. Save and close your `worker.properties` file.

Creating a ScanCentral SAST Sensor as a Service

If you use Windows services, you can install the sensor as a Windows service.

To install the sensor as a Windows service:

1. Navigate to the `<sca_install_dir>\bin\scancentral-worker-service` directory, and then do one of the following:
   - To use a clear text password, run `setupworkerservice.bat <sca_version> <full_controller_url> <shared_secret>`
   - To use an encrypted password, run `setupworkerservice.bat <sca_version> <full_controller_url> "<encrypted_shared_secret>" <path_to_pwtool.keys_file>`

     Important! Make sure that you enclose `<encrypted_shared_secret>` in quotation marks. This ensures that the encrypted shared secret does not get corrupted when the services installer creates the `worker.properties` file.

     Caution! The `setupworkerservice` command does not correctly handle `worker_auth_token` tokens that contain the caret character (^). If you must use the caret character as a part of a `worker_auth_token`, use the following formula:
     `saved_caret_count = carets_used_on_command_line / 8`

     Examples:
     For a worker_auth_token that contains a single caret, such as this^that, run the following command:
     ```
     setupworkerservice.bat 23.2 http://url.com this^^^^^^^^that
     ```
     For a worker_auth_token that contains two caret characters, such as this^^that, run the following command:
     ```
     setupworkerservice.bat 23.2 http://url.com this^^^^^^^^^^^^^^^^that
     ```
     For information about how to encrypt a shared secret, see "Encrypting the Shared Secret on a Sensor" on page 31.

2. Start the service, as follows:

   ```
   net start FortifyScanCentralWorkerService
   ```

The services installer creates the `<sca_install_dir>\Core\config\worker.properties` file for you.

See Next

"Enabling Sensor Auto-Start on Windows as a Service" on page 50

See Also

"Fortify ScanCentral SAST Components" on page 16

"Creating ScanCentral SAST Sensors" on page 42

(Windows only) Configuring Sensors to Offload Translation for .NET Languages

If you plan to use your ScanCentral SAST sensors for remote translation of code written in a .NET language, make sure that the following requirements are met.

ScanCentral SAST client machine requirements:

- MSBuild (See supported versions of MSBuild in the *Micro Focus Fortify Software System Requirements* document.)
- NuGet (optional)
- .NET Framework, .NET Core, or .NET Standard, depending on project configuration
- Windows operating system

ScanCentral SAST sensor machine requirements:

- .NET Framework supported for Fortify Static Code Analyzer

  A sensor machine must have .NET Framework 4.7.2 or later version, and .NET 6.0 Runtime or later installed to translate .NET projects.

  Note: Sensors accept .NET jobs if .NET Framework 4.7.2 or later is installed without .NET 6.0. However, translations will fail because Fortify Static Code Analyzer requires .NET 6.0 to successfully translate .NET projects. The sensor machine must have .NET 6.0 installed for successful remote .NET translation.
- Windows operating system

Tip: For information about specific version requirements, see the *Micro Focus Fortify Software System Requirements* document.

Beginning with (CloudScan) version 19.2.0, remote translation and scanning for .NET projects were supported. ScanCentral SAST supports the same MSBuild versions as Fortify Static Code Analyzer. (.NET packaging and scanning work only on Windows systems.)

The requirements for using this feature are as follows:

- Configure at least one sensor with the software required to support .NET capability.
- Clients must have the software required to build and pack .NET projects installed.

Enabling .NET Translation Capability on Sensors

To enable remote translation of .NET, do the following:

- Install .NET Framework version 4.7.2.
- Install .NET 5.0 runtime

After you start a ScanCentral SAST sensor, it automatically detects the .NET Framework version installed and displays a message that .NET capability is enabled for the detected .NET Framework version. The rule is not applied to .NET Core or .NET Standard because any .NET Framework version can scan this kind of project.

Remote translation of .NET is disabled if:

- .NET Framework is not installed on the sensor.
- A .NET Framework version earlier than 4.7.2 is installed on the sensor.

Important! To avoid Windows errors caused by too long a path during .NET translation, Fortify strongly recommends that you start ScanCentral SAST sensors from a folder with a short name and path. For more information, see https://docs.microsoft.com/en-us/windows/win32/fileio/naming-a-file.

Excluding .NET Projects from Analysis

To exclude a .NET project from ScanCentral SAST analysis, you must create a build configuration to exclude the project, and then specify the build configuration in the `--build-command` option.

Example: The *<solution_name.sln>* MSBuild solution includes two projects: ProjectA and ProjectB. The *<build_config>* file, created in Visual Studio, was created to exclude ProjectB from builds.

To exclude ProjectB from ScanCentral SAST translation and scanning run the folllowing:

```
cd <solution_dir>
scancentral package -bt msbuild -bf <solution_name.sln> -bc
"/t:Rebuild/p:Configuration=<build_config>" -o <package_name>.zip
```

Setting the Maximum Run Time for Scans

By default, a sensor can run a scan for an indefinite period of time, thus preventing it from running other scans. You can limit the amount of time scans can run on sensors by setting the `scan_timeout` option (in minutes) for a given job, for a given sensor, or globally for all sensors.

Precedence in Timeout Settings

The following rules of precedence apply to timeout settings:

- Job `timeout` settings override any sensor-specific or global timeout settings.
- Sensor timeout configured on the command line overrides a global timeout setting.

Configuring Maximum Run Time for a Specific Job

To configure the maximum run time of one minute for a given job, run the following:

```
scancentral -url <controller_url> start -package <path> --scan-time-out 1
```

To configure the maximum run time of two minutes for a given sensor, run the following:

```
scancentral -url <controller_url> worker --scan-time-out 2
```

Configuring Maximum Run Time for All Sensors

To configure the maximum run time for all sensors:

1. Navigate to the `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory, and open the `config.properties` file in a text editor.
2. Set the `scan_timeout` property to the maximum number of minutes for scans to run on sensors.
3. Save and close the `config.properties` file.

See Also

"Fortify ScanCentral SAST Command-Line Options" on page 88

Configuring Sensors to Use the Progress Command when Starting on Java

If you want to use the `progress` command to check the progress of your Fortify Static Code Analyzer scans, the following sensor configuration is required:

1. Create a JMX access file, and add the following text to it:

   `<user_role> readonly`

   where `<user_role>` is text that represents something like a username.
2. Create a JMX password file, and add the following text to it:

   `<user_role> <password> readonly`

   where `<user_role>` is the value you specified in the JMX access file.
3. Run one of the following commands:
   - On Windows systems, run `cacls jmxremote.password /P <username>:R`
   - On Linux systems, run `chmod 600 jmxremote.password`
4. Open the `worker.properties` file in a text editor, and then add the following properties to it:

```
sca_jmx_port=<port>
sca_jmx_access_file=<path_to_access_file>
sca_jmx_password_file=<path_to_password_file>
sca_jmx_password=<password>
sca_jmx_user=<user_role>
sca_jmx_auth=true
```

5. Save and close the `worker.properties` file.

After you complete this configuration, ScanCentral SAST clients start on the specified port using JMX password authentication. Make sure that the port is not already bound.

Important! If you use `sca_jmx_auth`, you can start only one sensor. Any attempt to open a new Fortify Static Code Analyzer instance results in a bind port error. To have multiple sensors on a machine, you must have several ScanCentral SAST instances, each with its own `worker.properties` file.

Changing Sensor Expiration Time

By default, sensors expire 168 hours after they become inactive. To reset this default value:

1. Navigate to the `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory, and open the `config.properties` file in a text editor.
2. Locate the `worker_expiry_delay` setting, and then change the number of hours to elapse after inactivity before sensors expire.

See Also

"Creating ScanCentral SAST Sensors" on page 42

Configuring Where job Files and the worker-persistence.properties File are Generated

For containerized deployments, it is useful to determine where certain files are generated so that you can customize persistence. This enables you to persist the `worker-persistence.properties` file, which you need to maintain sensor pool assignments, without having to keep all of the old job files.

Note: If you choose not to configure these locations, the default locations are used. The default location for the `worker-persist.properties` file is `<working_dir>/props`. The default location for the jobs files is `<working_dir>/jobs`.

To configure where job files and the `worker-persistence.properties` file get generated:

1. On a sensor machine, navigate to the <*sca_install_dir*>/Core/config directory, and then open the worker.properties file in a text editor.
2. Add the following properties to the file, and specify the directories for each:
   - The props_dir property determines where the worker_persist.properties file is to be saved.
   - The jobs_dir property determines the directory in which the jobs folders are to be created.
3. Save and close your worker.properties file.
4. Restart the sensor.

Avoiding Timeout Errors

To avoid timeout errors that can occur during attempts to upload very large log files, you can configure the connection and read timeouts between the Controller and sensors, between the Controller and clients, and between the Controller and Fortify Software Security Center.

Configuring the timeouts between the Controller and a sensor

To configure the connection and read timeouts between the Controller and a sensor:

1. On the sensor machine, navigate to the <*sca_install_dir*>/Core/config directory and open the worker.properties file in a text editor.
2. Raise the values of the restapi_connect_timeout and restapi_read_timeout properties to acceptable thresholds (in milliseconds).

   Note: The default value for restapi_connect_timeout is 10000 ms, and the default value for restapi_read_timeout is 30000 ms.
3. Save your changes.

Configuring the timeouts between the Controller and a client

To configure the connection and read timeouts between the Controller and a client:

1. On the client machine, navigate to the <*sca_install_dir*>/Core/config directory and open the client.properties file in a text editor.
2. Raise the values of the restapi_connect_timeout and restapi_read_timeout properties to acceptable thresholds (in milliseconds).

   Note: The default value for restapi_connect_timeout is 10000 ms, and the default value for restapi_read_timeout is 30000 ms.
3. Save your changes.

Configuring the timeouts between the Controller and Fortify Software Security Center

To configure the connection and read timeouts between the Controller and Fortify Software Security Center:

1. On the Controller, navigate to the `<controller_dir>/tomcat/webapps/scancentralctrl/WEB-INF/classes` directory and open the `config.properties` file in a text editor.
2. Raise the values of the `ssc_restapi_connect_timeout` and `ssc_restapi_read_timeout` properties to acceptable thresholds (in milliseconds).
3. **Note:** The default value for `ssc_restapi_connect_timeout` is 10000 ms, and the default value for `ssc_restapi_read_timeout` is 30000 ms.
4. Save your changes.

Starting the ScanCentral SAST Sensors

To start the sensors:

1. Start the Controller if it is not already running.
2. On each sensor, navigate to one of the following:
   - On a Windows system, `cd <sca_install_dir>\bin`
   - On a Linux system, `cd <sca_install_dir>/bin`
3. Run one of the following commands:

   On a Windows system:

   ```
   scancentral.bat -url <controller_url> worker
   ```

   On a Linux system:

   ```
   ./scancentral -url <controller_url> worker
   ```

If the sensor starts successfully, it prints messages that signal its waiting status to the console. After you verify that the sensor is working, you can create a Startup Task in Windows Task Scheduler or add it to your startup scripts. For more information, see "Configuring Sensor Auto-Start" on the next page.

Note: Make sure that you run a given sensor consistently from the same directory. Otherwise, its UUID changes and, if ScanCentral SAST is connected to Fortify Software Security Center, Fortify Software Security Center identifies it as different sensor.

See Also

"Placing the ScanCentral SAST Controller in Maintenance Mode" on page 40

Configuring Sensor Auto-Start

The following procedures are designed to provide general guidance to enable sensor auto-start and may not be appropriate in all environments. Fortify strongly recommends that you review the instructions with your system administrator and make any changes required for your environment.

Enabling Sensor Auto-Start on Windows as a Service

Check to make sure the Controller is running before you perform the following procedure.

To enable sensor auto-start on Windows as a service:

1. Log in to the sensor machine as a local admin user.

   Note: Sensors are dedicated machines that are meant only to run Fortify Static Code Analyzer on behalf of ScanCentral SAST; they are not shared with any other service. To avoid issues associated with insufficient privileges, use a fully-privileged administrative account for the auto-start setup.
2. Open a command prompt and navigate to the `<sca_install_dir>\bin\scancentral-worker-service` directory.
3. Run the `setupworkerservice.bat` script with no arguments to see the usage help.
4. Re-run the batch script with the required arguments included.
5. Open Windows Services and check to make sure that the sensor service is present.
6. Right-click the listed sensor service, and then select **Start**.
7. Fortify recommends that you change the startup type setting to **Manual** until you verify that the sensor runs successfully. After verification, change the startup type setting to **Automatic (Delayed Start)** in Windows Services.
8. Check to make sure that the sensor communicates with the Controller.

See Also

"Creating a ScanCentral SAST Sensor as a Service" on page 43

Troubleshooting

Review the following logs to troubleshoot issues encountered during the configuration of sensor auto-start as a Windows service:

- Main ScanCentral SAST sensor log on Windows:

  ```
  C:/Windows/System32/config/systemprofile/AppData/Local/Fortify/
  scancentral-<version>/log/scancentral.log
  ```
- Sensor temporary folders that contain MBS files, Fortify Static Code Analyzer log files, and generated FPR files: `C:/Users/Public/Fortify/SC/<job_token>`

- Sensor `stdOut` and `stdErr` logs: `C:/Users/Public/Fortify/SC/workerout.log` and `C:/Users/Public/Fortify/SC/workererr.log`

  Note: Before you start a sensor, check to make sure that the log files are not open in an application. Open log files prevent procrun from writing to the file.

- Commons-daemon log: `C:/Users/Public/Fortify/SC/<year_month_day>.log`

Enabling ScanCentral Sensor Auto-Start on Windows as a Scheduled Task

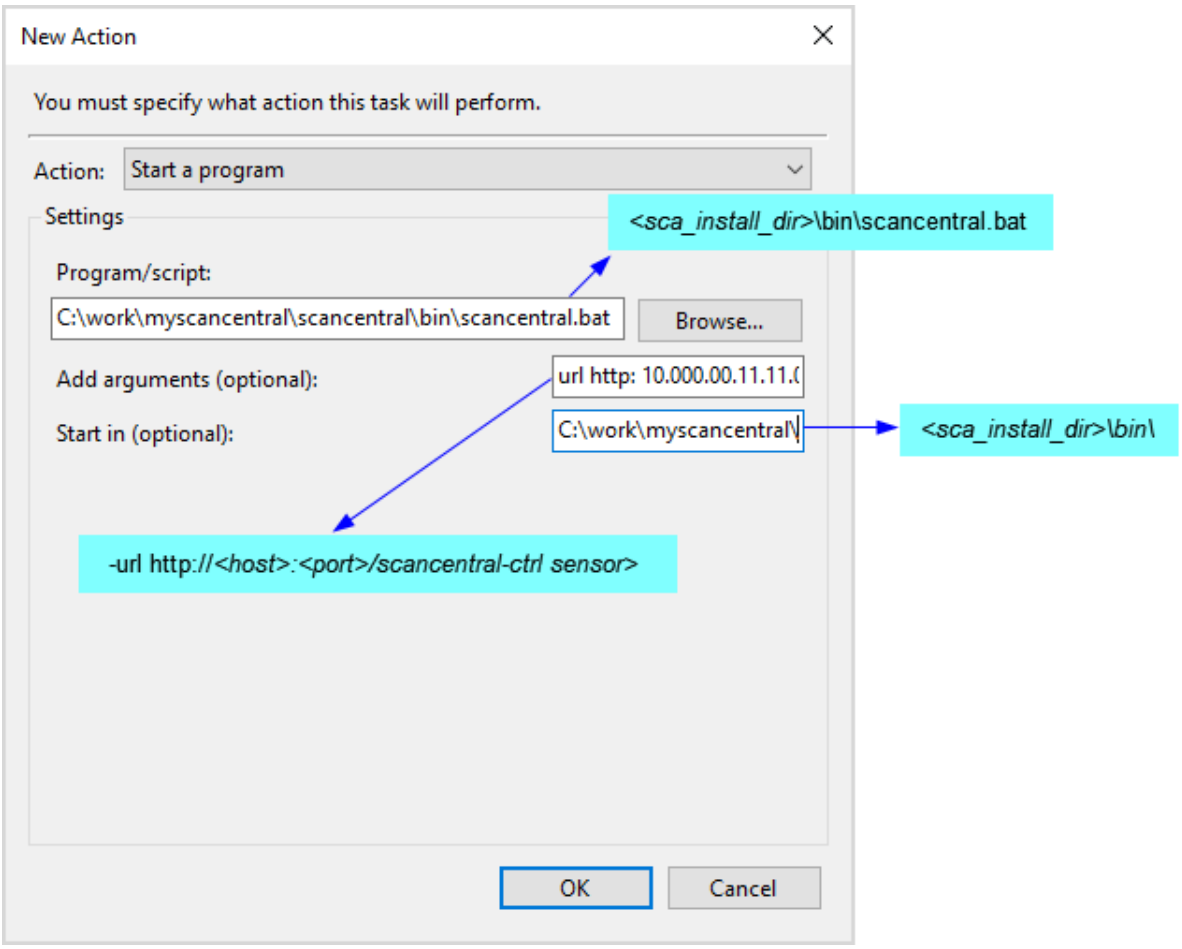
1. Log on to the sensor machine as the local admin user.

   Note: Sensors are dedicated machines that are meant only to run Fortify Static Code Analyzer on behalf of Fortify ScanCentral SAST; they are not shared with any other service. To avoid issues related to insufficient privileges, use a fully-privileged administrator account for the auto-start setup.

2. Start the Task Scheduler.
3. In the **Actions** panel, select **Create Task**.

   The Create Task window opens.
4. On the **General** tab, provide the following information:
   a. In the **Name** box, type a name for the task.
   b. Select the **Run whether user is logged on or not** option.
5. Select the **Actions** tab, and then click **New**.

   The New Action dialog box opens.

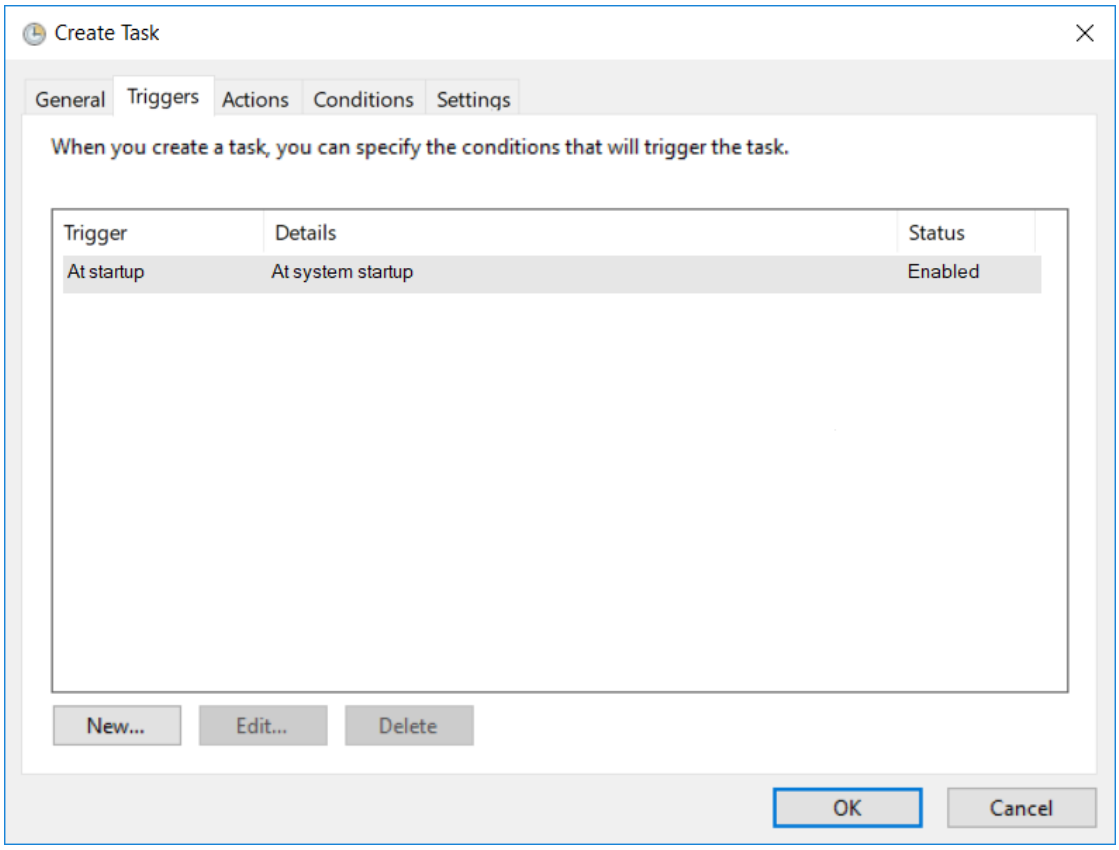


a. From the **Action** list, select a program to start.
b. In the **Program/script** box, type the directory path to your `scancentral.bat` file.
   Example: *<sca_install_dir>*`\bin\scancentral.bat`
c. In the **Add arguments (optional)** box, type the following:

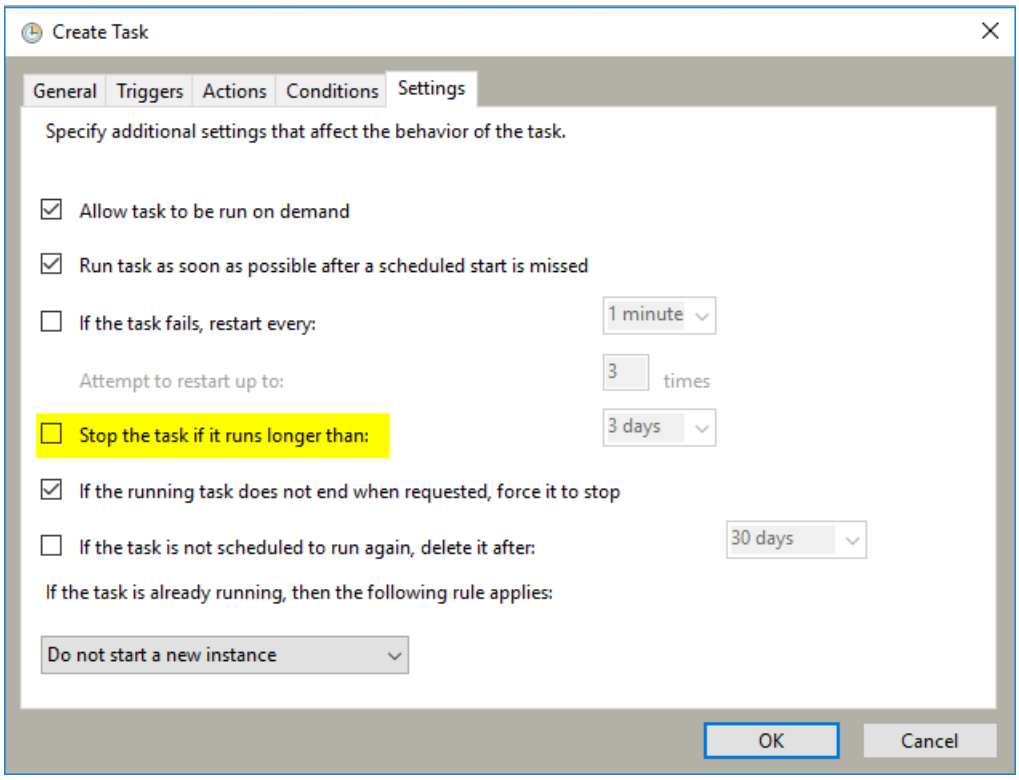
   `-url http://`*<host>*`:`*<port>*`/scancentral-ctrl worker >taskout.txt 2>&1`

d. In the **Start in (optional)** box, type the path to the ScanCentral sensor `bin` directory.
   Example: *<sca_install_dir>*`\bin\`
e. Click **OK**.

6. Return to the Task Scheduler and select the **Triggers** tab.



7. Check to make sure that the **At startup trigger** is enabled, and then click **OK**.
8. Select the **Settings** tab.



9. Make sure the **Stop the task if it runs longer than** check box is cleared, and then click **OK**.
10. Click **Save**.
11. Restart the machine.

The script output in the `taskout.txt` file indicates whether the sensor started successfully.

You can also start and stop the scheduled task manually from the Task Scheduler interface when logged into the machine.

Enabling ScanCentral Sensor Auto-Start on a Linux System

Note: The following procedure has been tested with Red Hat; there may be some variation for other Linux varieties. Please review these steps with your system administrator before you make any changes.

1. Log in to the machine as "root."
2. Run the `visudo` command to edit the `sudoers` file and disable requiretty.

```
Defaults  !requiretty
```

Note: You can also disable requiretty per user.

3. Set auto-start, as follows:
   a. Verify the command invocation from the console (modify according to your install directory).

      sudo -u <*username*> -- <*sca_install_dir*>/bin/ScanCentral -url <*controller_url*> worker > <*sca_install_dir*>/bin/workerout.txt 2>&1 &

      - Add the `sudo` command to the end of the file (add it before the line `exit 0` if it exists).
      - The ampersand (`&`) at the end enables the machine to boot up even if sensor startup fails or hangs.
      - The double-dash (`--`) is important to separate the options for sudo from the options for your service.

   b. Make the change to the startup file.

      Caution! Make sure that you do not change anything else in your bootup script.

      ```
      vi /etc/rc.d/rc.local
      ```

4. Check the setup:
   a. Reboot and log in to the machine as "root."
   b. To verify the processes under root, type:

      ```
      ps -x | grep java
      ```

   c. Verify that the output shows that the sensor is not started under root.
   d. To verify the processes under the user, type:

      sudo -u <*username*> ps x | grep java

   e. Verify that the output displays the sensor process.
   f. To verify the existence and contents of the script output file, type:

      tail -f/opt/<*sca_install_dir*>/bin/workerout.txt

      Example: `tail -f/Fortify/Fortify_SCA_<version>/bin/workerout.txt`

Safely Shutting Down Sensors

This section describes how to move ScanCentral SAST sensors to shutdown, or shutdown scheduled mode from Fortify ScanCentral SAST.

Important! If the Controller is in maintenance mode (see "Placing the ScanCentral SAST Controller in Maintenance Mode" on page 40), you cannot shut down sensors from Fortify Software Security Center. Also, in order to shut down sensors from Fortify Software Security Center, the sensors must be version 21.2.0 or later.

Shutting Down Sensors

To shut down active sensors:

1. Log on to Fortify ScanCentral SAST as an administrator, and then, on the Fortify header, click **SCANCENTRAL**.
2. In the left pane of the **SAST** tab, select **Sensors**.
3. In the sensors table, do one of the following:
   - Expand the row for a sensor you want to shut down, and then click **SHUT DOWN**.
   - Select the check boxes for one or more sensors you want to shut down, and then click **SHUT DOWN**.

   Note: If the **SHUT DOWN** button is not enabled, it can mean that:
   - The sensor version is earlier than 21.2.0
   - The sensor was already shut down
   - The Controller is in maintenance mode
   - The sensor is inactive or disabled

If a sensor you shut down is running a scan, the **State** value for the sensor changes from **Active** to **Shutdown scheduled**. After the scan is completed, the state then changes to **Inactive**.

Chapter 4: About Clients

A client is a build machine on which Fortify Static Code Analyzer translates code and generates Fortify Static Code Analyzer mobile build sessions (MBS). The translated source code, along with optional and required data, such as custom rules and Fortify Static Code Analyzer command-line arguments, are uploaded to the Controller.

Clients not only translate code and generate MBSs, but can also generate packages with sources and dependencies for remote translation on sensors. (You can use this functionality independent of Fortify Static Code Analyzer.)

Embedded Clients and Standalone Clients

A client can be either an *embedded* client, which is part of the Fortify Static Code Analyzer distribution/installation, or a *standalone* client, which is independent of Fortify Static Code Analyzer.

Within a Fortify Static Code Analyzer installation, the files used to create ScanCentral SAST sensors and embedded clients are the same. The only difference is how you invoke their functionality from the command line. To use ScanCentral SAST as a sensor, you run ScanCentral SAST using the `worker` command. To use ScanCentral SAST as a client to initiate a scan, you invoke it using the `start` command. Sensor functionality depends on Fortify Static Code Analyzer. So, you can have a standalone client, but not a standalone sensor.

The interface for issuing Fortify ScanCentral SAST commands is installed on your clients. You can use this interface to create or identify a Fortify Static Code Analyzer mobile build session, set the parameters for the scan, and communicate your intentions to the ScanCentral Controller.

Note: A standalone client, which does not require that Fortify Static Code Analyzer be installed, may pack the code with dependencies into a package to send to the Controller for translation and scanning.

See Also

"Avoiding Timeout Errors" on page 48

Installing ScanCentral SAST Clients

Unless you use a language that supports offloading the translation phase of analysis to your sensors, you must have a licensed copy of Fortify Static Code Analyzer on each of the machines you plan to use as ScanCentral SAST clients. If you use a language that supports offloading the translation phase of analysis to your sensors, you can install standalone clients, independent of Fortify Static Code Analyzer.

The languages and container configurations that are supported for offloading the translation phase of analysis are:

- Python
- Go
- Ruby
- JavaScript
- PHP
- Java
- ABAP (Advanced Business Application Programming)
- Apex (Salesforce)
- Classic ASP (ASP Classic)
- Adobe ColdFusion
- PL/SQL / T-SQL
- Microsoft TypeScript
- Visual Basic 6.0
- .NET applications (C#, VB.NET, .NET Core, ASP.NET, and .NET Standard)
- Dockerfiles

Caution! As you specify an installation path, make sure that the path name contains no spaces.

Creating a Standalone Client

If you plan to offload both the translation and scanning phases of analysis to your ScanCentral SAST sensors, you can use standalone clients.

Important! Before you install a standalone client, you must first download and configure a Java Runtime Environment (JRE) on the machine on which you plan to install it. For information about supported JRE versions, see the *Fortify Software System Requirements* guide. For information about how to download and configure a JRE, see the documentation for the supported JRE version.

To install a standalone client (independent of Fortify Static Code Analyzer):

1. Extract the contents of the `Fortify_ScanCentral_Client_<version>_x64.zip` file to any directory on your machine.
2. On the machine to which you extracted the `Fortify_ScanCentral_Client_<version>_x64.zip` file, install JRE 11.
3. Set the `JAVA_HOME` environment variable to point to JRE 11, and make sure that you add the java executable to the `PATH` environment variable.

Important! If you have a Java 8 project that fails to build because ScanCentral SAST requires Java 11 to run, set the `SCANCENTRAL_JAVA_HOME` environment variable to point Java 11. After you do, ScanCentral SAST runs correctly, and the build runs with the `JAVA_HOME` set to Java 8.

4. Navigate to the `Core/config` directory, and open the `client.properties` in a text editor.
5. Add the `client_auth_token` property to the `client.properties` file, and set the same value for it that you set for the `client_auth_token` property on the Controller (in the *<controller_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file).

Caution! If you want the `client_auth_token` property value encoded in the `config.properties` file on the Controller, take the decoded value, encode it on the client machine, and then save the encoded value to the `client.properties` file.

Placing Multiple Standalone Clients Under the Controller

You can place multiple standalone clients of different supported versions in the Controller. To do this, just place any number of client ZIP files for any and all supported versions into the *<controller_dir>*`/tomcat/client` directory. The ZIP file names themselves are unimportant. On startup, the Controller parses the available clients.

To install a patch for a given client or sensor version installed on the Controller, place the patch ZIP file into the *<controller_dir>*`/tomcat/client` directory. If auto-upgrade is enabled, the clients of that version are automatically upgraded with the patch. For information about how to enable or disable automatic updates of your clients and sensors, see "Enabling and Disabling Auto-Updates of Clients and Sensors" on page 68.

Installing an Embedded Client Using Fortify Static Code Analyzer

Use the following procedure to install an embedded client (client included with the Fortify Static Code Analyzer installation) if you do *not* plan to offload project translation to your sensors.

To install an embedded client:

1. Log on to a build machine using credentials for an account that is *not* an administrator or root account.
2. Use the instructions provided in the *Fortify Static Code Analyzer User Guide* to install Fortify Static Code Analyzer on your build machine.
3. Make sure that you add the `client_auth_token` property to the `client.properties` file and set the same value for it that you set for the `client_auth_token` property on the Controller (in the `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file).

Upgrading a Client

Important! Fortify recommends that your standalone ScanCentral SAST clients and your Fortify Static Code Analyzer installation be the same version.

To upgrade a standalone client (independent of Fortify Static Code Analyzer):

- Delete the client, and then extract the `Fortify_ScanCentral_Client_<version>_x64.zip` file to any directory on the machine.

  Or,

- Extract the contents of the `Fortify_ScanCentral_Client_<version>_x64.zip` file on top of the existing client.

To upgrade an embedded client that resides on the same machine as Fortify Static Code Analyzer:

1. Log on to the build machine using credentials for an account that is *not* an administrator account or root.
2. Back up the following directories:
   - `<sca_install_dir>/bin`
   - `<sca_install_dir>/Core/lib`
   - `<sca_install_dir>/Core/config`
3. Upgrade Fortify Static Code Analyzer. For instructions on how to install and upgrade Fortify Static Code Analyzer, see the *Fortify Static Code Analyzer User Guide*.

4. Accept all overwrite requests.

Note: On a Linux system, you may also need to run `chmod +x ScanCentral` (in the `<sca_install_dir>/bin/ScanCentral` directory).

Tip: After you configure a client, you can copy the configuration files and use them to create other clients.

See Also

"(Windows only) Configuring Sensors to Offload Translation for .NET Languages" on page 44

"Configuring Sensors to Use the Progress Command when Starting on Java" on page 46

"Viewing ScanCentral Logs" on page 63

Configuring Proxies for Fortify ScanCentral SAST Clients

If all of your outbound traffic must go through a proxy, you can configure one for your clients.

1. Go to the `<sca_install_dir>/Core/config` directory, and, in both the `client.properties` and `worker.properties` files, uncomment, and then set values for the properties listed in the following table.

Property	Description
ctrl_proxy_host	Type the name of the Controller proxy host.
ctrl_proxy_port	Type the Controller proxy port number.
ctrl_proxy_user	If authentication is required, type a user name.
ctrl_proxy_password	If authentication is required, type the password for the user.
ssc_proxy_host	Type the name of the Fortify Software Security Center proxy host.
ssc_proxy_port	Type the number of the Fortify Software Security Center proxy port.
ssc_proxy_user	If authentication is required, type a Fortify Software Security Center user name.

Property	Description
ssc_proxy_password	If authentication is required, type the password for the Fortify Software Security Center user.

2. To enable proxy authentication when the Controller is running under HTTPS, go to the *<sca_install_dir>*/bin directory, and then add the following property to the scancentral.bat file:

   ```
   -Djdk.http.auth.tunneling.disabledSchemes
   ```

   Example:

   ```
   $JAVA_CMD -Djdk.http.auth.tunneling.disabledSchemes= -
   Dscancentral.installRoot="${FORTIFY_HOME}" -Dlog4j.dir="${SCANCENTRAL_
   LOG}" $SCANCENTRAL_JAVA_PROPS -jar "${FORTIFY_HOME}/Core/lib/scancentral-
   launcher-22.2.0.0.jar" "$@"
   ```

Using the MSBuild ScanCentral SAST Integration

To use MSBuild ScanCentral SAST integration, the required MSBuild version must be on the PATH. To make sure the project is built correctly, Fortify recommends that you start ScanCentral SAST from the Developer Command Prompt for Visual Studio, which sets the required .NET environment variables automatically.

Some projects also require that you start NuGet to restore some dependencies. If any dependencies are unresolved, the MSBuild would fail and the scan results might be incomplete. For these kinds of projects, you need to install NuGet manually on the machine and make sure it is available on the PATH. If NuGet is found, ScanCentral SAST runs it automatically.

To translate and scan a .NET project on ScanCentral SAST, run the following:

```
scancentral -url <controller_url> start --build-tool msbuild --build-file
<solution file name or path to solution file> [--save-package]
```

Alternatively, you can save the project package locally, as follows:

```
scancentral package -o <path to package> --build-tool msbuild --build-file
<solution file>
```

To send the package to ScanCentral SAST, run:

```
scancentral -url <controller_url> start -package <package path>
```

ScanCentral SAST returns a job ID that you can use to track the scan.

Chapter 5: Viewing ScanCentral Logs

To retrieve the ScanCentral Controller log, navigate to `<controller_dir>\tomcat\logs\scancentralCtrl.log`.

To view the ScanCentral client and sensor logs on a Windows system:

- Navigate to `%FORTIFY_HOME%\scancentral-<version>\log`, where `%FORTIFY_HOME% is %LOCALAPPDATA%\Fortify`.
  On Windows 10, for example, the location is

  `C:\Users\<user>\AppData\Local\Fortify\scancentral-<version>\log`
- To retrieve the ScanCentral SAST log on a Linux system, navigate to `~/.fortify/scancentral-<version>/log/scancentral.log`.

Enabling Debugging on Clients and Sensors

ScanCentral SAST logging typically provides enough information to follow the flow of operations under normal conditions. If things are not working as expected, the logging may not provide enough information to determine the actual root cause of the issue.

If ScanCentral SAST does not provide enough information to diagnose a situation, you can configure the logging level for the clients and for sensors. To increase the log level for clients and sensors, use the use the `-debug` command-line option. (See "Global Options" on page 88.) Make sure that you specify the -debug option *before* the action (start, retrieve, and so on).

Examples:

```
scancentral -debug -url <url> worker
scancentral -debug -url <url> start
```

The next time the sensor is called, the log contains debug-level information.

For information about how to configure the logging level for the Controller, see "Configuring the Logging Level on the Controller" on page 39.

Chapter 6: About Upgrading ScanCentral SAST Components

ScanCentral SAST-related functionality in Fortify Software Security Center requires updated ScanCentral SAST components.

Important! You must upgrade the Controller before you upgrade the ScanCentral SAST sensors and clients. Also, make sure that your Controller version is the same as your Fortify Software Security Center server version.

Caution! A sensor of a given version does not support packages that clients of an earlier version have generated. For example, if you want to offload translation for scan projects uploaded by a version 22.2.0 client, do not upgrade your sensors to version 23.1.0 or 23.2.0.

This section contains the following topics:

Support for Multiple Fortify Static Code Analyzer Versions

To support heterogeneous environments and facilitate phased Fortify Static Code Analyzer upgrades, the ScanCentral Controller supports scan request routing based on the Fortify Static Code Analyzer version. For example, you can configure two different client machines, each with a different Fortify Static Code Analyzer version, and configure the sensors with compatible Fortify Static Code Analyzer versions. Jobs from each client are then routed to the sensor that has the same Fortify Static Code Analyzer version installed.

If you have an existing Fortify Static Code Analyzer installation (that includes `scancentral.bat`) in your path and a mixed version environment, make sure that you are running the latest ScanCentral SAST executable when you run the client and sensor commands. (Use explicit paths.) Adding capacity (new clients or sensors) is simple—just clone the VMs you have already configured, or use sensor hosts with the same specifications and installation folder structure.

Important! If you clone VMs, then after cloning, you *must* remove the `worker_persist.properties` file from the directory that was specified for the `props_dir` property (see "Configuring Where job Files and the worker-persistence.properties File are Generated" on

page 47).

Note: Use sensor machines dedicated to ScanCentral SAST and run sensors under a dedicated username. Run only one sensor instance per machine.

If the Controller and Fortify Software Security Center run on different machines, you must check to make sure that `scancentral-ctrl\WEB-INF\classes\config.properties (ssc_url, this_url)` and the ScanCentral Controller URL set on Fortify Software Security Center (select **Administration** > **Configuration** > **ScanCentral SAST**) resolve to the correct IP addresses.

Check to make sure that the following channels of communication are not blocked by a firewall or other tool:

- Controller to Fortify Software Security Center port (for scan uploads)
- Fortify Software Security Center to the ScanCentral Controller port (for Fortify ScanCentral SAST administration console functionality)
- Clients to the ScanCentral Controller port
- Sensors to the ScanCentral Controller port
- Clients to the Fortify Software Security Center port (required only if Fortify Software Security Center is in lock down mode, or if the `-ssc_url` option is used)

Upgrading the ScanCentral SAST Controller

The following procedure describes how to upgrade the Controller.

Caution! Before you upgrade the Controller, you must first download and configure a Java Runtime Environment (JRE). For information about supported JRE versions, see the *Fortify Software System Requirements* guide. For information about how to download and configure JRE, see the Oracle documentation for the supported JRE version.

To upgrade your ScanCentral SAST Controller:

1. Go to the Software Licenses and Downloads (SLD) portal (`https://sld.microfocus.com`) and download the `Fortify_ScanCentral_Controller_<version>_x64.zip` file.

   Note: For detailed instructions on how to download Fortify Software, see https://www.brainshark.com/mfLD/vu?pi=zFszsRA7ezW1H3z0&nodesktopflash=1.

2. (Recommended) Allow all jobs to finish.

   Note: If you do not allow all jobs to finish before you shut down the Controller, some jobs fail after the upgrade, and the failure may not be evident for some time. (See the `worker_inactive_delay` configuration parameter in the `<new_controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file.)

3. Shut down the Controller.

4. Install the new Controller. (For information, see "Installing the Controller" on page 19.)
5. If your existing `config.properties` file has been modified, you must merge it with the new `config.properties` file. (You cannot simply copy the existing `config.properties` file.)
6. If (and only if) you are upgrading the Controller from a 22.2.x or earlier version to version 23.1.0, run the migration script as follows. Please revisit.

   Note: If you are upgrading the Controller from a 22.2.0 or later version, you need not run the script. For example, if you are currently on 22.1.0 and are upgrading to 23.1.0, you must run the migration script. But, if you are on 22.2.0 and upgrading to 23.1.0, you do not need to run the migration script.

   a. Extract the contents of the `Fortify_ScanCentral_Controller_<version>_x64.zip` file.
   b. Open a command prompt, and go to the `db-migrate` directory.
   c. Identify the `cloudCtrlDb` and Controller directories for the older (existing) Fortify ScanCentral SAST version.

      Example:

      ```
      C:/scancentral<version>/tomcat/cloudCtrlDb
      C:/scancentral<version>/tomcat/webapps/scancentral-ctrl
      ```

7. If you are upgrading a 22.1.x or earlier version Controller to version 22.2.x, run the following command. (This command includes the example directories shown in the preceding step.)

   ```
   migrate C:/scancentral22.1/tomcat/cloudCtrlDb
   C:/scancentral22.1/tomcat/webapps/scancentral-ctrl
   ```

   The `cloudCtrlDb` directory is generated in the current working directory.
8. Navigate to the `jobFiles` and `cloudCtrlDb` directories of the existing Controller, and then copy these to the new Controller. (If you used the preceding step to migrate the database, make sure that you use that migrated database rather than the unmigrated database.)
9. Start the new Controller. (The database is automatically migrated.)

See Also

"About Upgrading ScanCentral SAST Components" on page 64

"Upgrading ScanCentral SAST Sensors" below

"Enabling and Disabling Auto-Updates of Clients and Sensors" on page 68

Upgrading ScanCentral SAST Sensors

Important! If Fortify Static Code Analyzer is installed in a location that requires that you have administrator privileges to modify it (for example, program files), in order to update a sensor, you must start it with administrator privileges. Otherwise, the sensor cannot write files to disk. If auto-update is enabled, major updates on standalone clients must finish successfully before the sensor can start. With auto-update enabled, patch updates allow sensors and clients to start unless the

upgrade fails.

To upgrade your ScanCentral SAST sensors (on Windows or Linux), you can either install the latest version of Fortify Static Code Analyzer, or unzip the `Fortify_ScanCentral_Client_<version>_x64.zip` file. You can use the client-only approach if you will plan only to use remote translation and analysis workflows. Local translation requires a local Fortify Static Code Analyzer installation. You can also find the ScanCentral SAST client inside the `Fortify_ScanCentral_Controller_<version>_x64.zip` file in the `tomcat/client/scancentral.zip` directory.

Tip: You can configure automatic upgrades of both ScanCentral SAST sensors and clients. For details, see "Enabling and Disabling Auto-Updates of Clients and Sensors" on the next page.

To upgrade sensors by installing or upgrading Fortify Static Code Analyzer:

1. Stop all sensors from running.
2. Go to the Software Licenses and Downloads (SLD) portal (`https://sld.microfocus.com`) and download the installer file for your operating system:

   Windows: `Fortify_SCA_<version>_windows_x64.exe`

   Linux: `Fortify_SCA_<version>_linux_x64.run`

   Note: For detailed instructions on how to download Fortify Software, see https://www.brainshark.com/mfLD/vu?pi=zFszsRA7ezW1H3z0&nodesktopflash=1.
3. Install or upgrade Fortify Static Code Analyzer based on the instructions provided in the *Fortify Static Code Analyzer User Guide*.
4. Check the `<sca_install_dir>/Core/config` directory to make sure that the `worker.properties` file resides there.
5. Add the following property to the `worker.properties` file:

   ```
   worker_auth_token=<value_set_in_controller_configuration>
   ```
6. Specify either a clear text password, or an encrypted shared secret (password the Controller uses to communicate with the sensor) as the `worker.properties` value. For information about how to generate an encrypted shared secret, see "Encrypting the Shared Secret on a Sensor" on page 31.
7. Save the `worker.properties` file.
8. Start the sensors.

See Also

"Enabling and Disabling Auto-Updates of Clients and Sensors" on the next page

"Creating ScanCentral SAST Sensors" on page 42

"Installing ScanCentral SAST Clients" on page 57

"About Upgrading ScanCentral SAST Components" on page 64

"Upgrading the ScanCentral SAST Controller" on page 65

Enabling and Disabling Auto-Updates of Clients and Sensors

You can have all ScanCentral SAST clients and sensors check with the Controller after a manual update and following each startup to determine whether updates are available (the client or sensor version is earlier than the Controller version). Then, if an update is available, the Controller updates all sensors and clients.

The upgrade paths for clients and sensors are as follows:

- Standalone clients can be upgraded to a patch or major version (for example from 22.2.0 to 23.1.0, or from 23.1.0 to 23.2.0).
- If auto-upgrade is enabled and a major upgrade of standalone clients fails, the clients do not start any jobs until they are upgraded.
- If auto-upgrade is enabled and a patch upgrade of standalone clients fails, the clients continue to work, but a warning is displayed.
- You can upgrade embedded clients and sensors to a patch version only (for example, from 23.1.0 to 23.1.1 or 23.1.2, but not to 23.2.0). Auto-upgrade for major versions is not available for embedded clients and sensors.
- If auto-upgrade is enabled and a patch upgrade of an embedded client fails, the clients and sensors continue to work but a warning is displayed.

To upgrade sensors and embedded clients to the next version, you must install the latest Fortify Static Code Analyzer version.

About Scan Assignment

Clients can assign scans to Fortify Static Code Analyzer instances that have the same major version and any patch of that version. For example, a 23.1.0 client can send scans to Fortify Static Code Analyzer versions 23.1.0, 23.1.1, 23.1.2, and so on. However, a client cannot assign scans to Fortify Static Code Analyzer of a different major version. For example, 23.1.0 clients cannot send scans to Fortify Static Code Analyzer version 23.2.0.

Important! ScanCentral SAST clients and sensors check for updates only if you use the `-url` or `-sscurl` options. The `package` command will not start the update process.

To enable or disable automatic updates of your clients and sensors:

1. Navigate to the `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory and open the `config.properties` file in a text editor.
2. Locate the `client_auto_update` property.
3. To enable auto-updates, set `client_auto_update` to `true`. To disable auto-updates, set the value to `false` (the default).
4. Save and close the file.

The update process (and its resulting success or failure status) is printed to the console.

Important! If Fortify Static Code Analyzer is installed in a location that requires that you have administrator privileges to modify it (for example, program files), in order to update the sensor, you must start it with administrator privileges. Otherwise, the sensor cannot write files to disk. If auto-update is enabled, major updates on standalone clients must finish successfully before the sensor can start. With auto-update enabled, patch updates allow sensors and clients to start unless the upgrade fails.

See Also

"About Upgrading ScanCentral SAST Components" on page 64

"Upgrading the ScanCentral SAST Controller" on page 65

Chapter 7: Fortify Static Code Analyzer Mobile Build Session Version Compatibility

The Fortify Static Code Analyzer version on a ScanCentral client must be compatible with the Fortify Static Code Analyzer version installed on the sensors. The version number format is `major.minor.patch.buildnumber` (for example 22.2.0.0080). The major and minor portions of the Fortify Static Code Analyzer version numbers on both the ScanCentral client and sensor must match. For example, 22.2.0 works with 22.2.x.

To check the Fortify Static Code Analyzer version used, run the command `sourceanalyzer.exe -version`.

Chapter 8: Submitting Scan Requests

Depending on the language used to develop your source code, you can request a scan that offloads only the scanning phase of code analysis, or a scan that offloads both project translation and scanning to your ScanCentral SAST sensors.

Offloading Scanning Only

To submit a scan request that offloads only the scanning phase of code analysis, run the following command:

```
scancentral.bat –url <controller_url> start -b <my_build_id> -scan
```

You can pass any relevant Fortify Static Code Analyzer scan tuning option on the command line after the `–scan` keyword. If you use options such as `–build-label`, `-build-application`, or `–build-version`, make sure that you escape any quotes around the parameter. For example:

```
-scan –build-label \"Application 5.4 – November 20, 2022\"
```

If the submission succeeds, you receive a token. The Fortify ScanCentral SAST sensor pulls the scan request from the Controller, processes it, and publishes the results to the Controller.

For information about the options to use for larger scans, see the *Micro Focus Fortify Static Code Analyzer User Guide*.

Note: Jobs submitted (and FPRs) can be no larger than 1 GB. Before you start large scans, review "Optimizing Scan Performance" on page 79.

Targeting a Specific Sensor Pool for a Scan Request

To target a specific sensor pool for a scan request, you must have:

- UUID for the sensor pool
- `pool_mapping_mode` property set to enabled or disabled

To get the UUID for the sensor pool:

1. Log on to Fortify Software Security Center.
2. On the Fortify header, select **SCANCENTRAL**.
3. In the left panel, select **Sensor Pools**.
   The **Sensor Pools** table lists the existing sensor pools.

4. In the **Sensor Pools** table, copy the value shown in the **Pool UUID** column for the sensor pool you want to target for a scan request.

   Note: All sensors that are unassigned and enabled are used, even they are not assigned to sensor pools.

To specify a sensor pool to use for a scan request:

- From the command line on the client host, run the following:

```
scancentral.bat –url <controller_url> start -b <mybuildid> -pool <uuid> -
scan
```

Offloading Both Translation and Scanning

If you use a supported language, you can offload both translation and scanning phases of code analysis to your ScanCentral SAST sensors.

ScanCentral SAST detects the build tool you are using automatically based on the project files being scanned. For example, if Fortify ScanCentral SAST detects a `pom.xml` file, it automatically sets `-bt` to `mvn`. If it detects a `build.gradle` file, it sets `-bt` to `gradle`. If Fortify ScanCentral SAST detects a `*.sln` file, it sets `-bt` to `msbuild` and sets `-bf` to the `xxx.sln` file.

If ScanCentral detects multiple file types (for example, `pom.xml` and `build.gradle`), it prioritizes the build tool selection as follows: Maven > Gradle > MSBuild and prints a message to indicate which build tool type was selected based on the multiple file types found.

In the examples shown in the following table, ScanCentral SAST is integrated with Fortify Software Security Center, email is configured for ScanCentral SAST, and Fortify Software Security Center, the Controller, and sensors are up and running. (Note that the `-bt` option in these commands is not required.)

Objective	Command
Start a job to scan a MSBuild project	`scancentral.bat -url <controller_url> start -bt msbuild -bf mySolution.sln`

Start a job to scan a Maven project that includes the test scope	`scancentral.bat -url <controller_url> start -bt mvn -t`
	or
	`scancentral.bat -url <controller_url> start -t`

Start a job to scan a Maven project with a non-default build file	`scancentral.bat -url <controller_url> start -bt mvn -bf c:\myproj\myproj-pom.xml`

Objective	Command
Start a job to scan a JavaScript/TypeScript project	`scancentral.bat -url <controller_url> start -bt none`
Start a job to scan a PHP 7.1 project	`scancentral.bat -url <controller_url> start -bt none -hv 7.1`
Start a job to scan an ABAP project	`scancentral.bat -url <controller_url> start -bt none`
Start a job to scan a Ruby project	`scancentral.bat -url <controller_url> start -bt none`
Start a job to scan a Gradle project	`scancentral.bat -url <controller_url> start -bt gradle`
Start a job to scan a Gradle project, get email notifications from the Controller, and upload the results to Fortify Software Security Center	`scancentral.bat -url <controller_url> start -bt gradle -email username@domain.com -upload -uptoken <ssc_upload_token> -application "MyProject" -version "1.0"`

Working with Go Projects

ScanCentral SAST clients can package Go projects for remote translation and scanning. To enable this, the following requirements must be met:

- The Go compiler must be installed on clients to resolve project dependencies.
- The Go compiler executable location must be available in the PATH variable.
- Because ScanCentral SAST relies on Go environment variables, you must configure things accordingly. For example, to use a specific Go proxy, configure it as follows:

  `set GOPROXY=....` (Windows)

  `export GOPROXY=...` (Linux)

  Note: Sensors do not require a connection to a Go proxy website to resolve dependencies because they run Go translation with `GOPROXY=off` configured. Also, the vendor folder under the project root has all of the required dependencies. It rewrites the `GOFLAGS` system variable with `GOFLAGS=-mod=vendor` when running a Fortify Static Code Analyzer translation.

- The Go project must include a `go.mod` file.

To start a job to scan a Go project, run the following:

```
scancentral.bat -url <controller_url> start -bt none
```

Working with Python Projects

Objective	Command
Start a job to scan a Python 3 project	`scancentral.bat -url <controller_url> start -bt none --python-version 3 --python-requirements <path_to_requirements_file>`
Start a job to scan a Python project under an active virtual environment with dependencies already installed	`scancentral.bat -url <controller_url> start -bt none`
Start a job to scan a Python project under an active virtual environment without project dependencies installed	`scancentral.bat -url <controller_url> start -bt none --python-requirements <path_to_requirements_file>`
Start a job to scan a Python project using an existing Python virtual environment and install project dependencies	`scancentral.bat -url <controller_url> start -bt none --python-virtual-env <virtual_environment_location> --python-requirements <path_to_requirements_file>`

You can use ScanCentral SAST to work with Python in any of three ways. You can start ScanCentral SAST in a prepared virtual environment (see "Starting ScanCentral SAST in a Virtual Environment" below). You can use an existing virtual environment, without activating that virtual environment (see "Starting ScanCentral SAST in an Unactivated Virtual Environment" on the next page). In this case, ScanCentral SAST activates the virtual environment itself. Finally, you can start the job outside of a virtual environment (see "Starting ScanCentral SAST Outside of a Virtual Environment" on the next page).

Starting ScanCentral SAST in a Virtual Environment

If you work in a virtual environment, all of your project dependencies are already installed. You do not need to invoke the pip package manager before you start ScanCentral SAST, or to specify the Python version (this is detected automatically).

To start ScanCentral SAST in a virtual environment:

1. Open a command line.
2. Activate the virtual environment.
3. Start ScanCentral SAST.

Example: `scancentral.bat -url <controller_url> start -bt none`

If pip dependencies are not yet installed in the virtual environment used, ScanCentral SAST installs them automatically using the requirements file:

```
scancentral.bat -url <controller_url> start -bt none --python-requirements
<path_to_requirements_file>
```

Starting ScanCentral SAST in an Unactivated Virtual Environment

To start ScanCentral SAST in a virtual environment (with all dependencies installed) without activating that virtual environment:

1. Open a command line.
2. Start the Python project scan:

   ```
   scancentral -url <controller_url> start -bt none --python-virtual-env
   <venv_location>
   ```

   or

   ```
   scancentral -url <controller_url> start -bt none --python-virtual-env
   <venv_location> --python-requirements <path_to_requirements_file>
   ```

ScanCentral SAST goes to the virtual environment, determines the Python version used, packages all required libraries, and then creates the package.

Starting ScanCentral SAST Outside of a Virtual Environment

If you plan to start ScanCentral SAST and there is no virtual environment on the client, you must have Python installed on the client, specify the Python version, and specify the Python requirements file. ScanCentral SAST locates the Python installation. In this case, ScanCentral SAST creates a temporary virtual environment, installs all dependencies from the requirements file, and then generates the package.

To start ScanCentral SAST outside of a virtual environment:

1. Open a command line.
2. Start ScanCentral SAST.
3. Run the following:

   ```
   scancentral –url <controller_url> start -bt none --python-requirements
   <path> --python-version <version>
   ```

Working with Apex Projects

To perform remote translation of an Apex project, you must specify an additional translation argument for the project so that Fortify Static Code Analyzer "knows" that the CLS files are related to Apex, and not to Visual Basic 6.

To scan an Apex project using ScanCentral SAST, run the following:

```
scancentral -url <controller_url> start -bt none -targs "-apex"
```

Alternatively, you can save the project package locally, as follows:

```
scancentral package -o <path to package> -bt none -targs "-apex"
```

To send an existing package to ScanCentral SAST, run the following:

```
scancentral -url <controller_url> start -package <package path>
```

ScanCentral SAST returns a job ID that you can use to track the scan.

Working with SQL Projects

To perform remote translation of a SQL project, you must specify an additional translation argument for the project so that Fortify Static Code Analyzer "knows" what type of SQL (T-SQL or PL/SQL) is required. (By default, on Windows, Fortify Static Code Analyzer uses T-SQL, but on Linux, it uses PL/SQL.)

Note: For information about using the `-sargs` and `-targs` options, see "Fortify ScanCentral SAST Command-Line Options" on page 88.

To scan the project, run the following command:

```
scancentral -url <controller_url> start -bt none -targs "-sql-language
<PL/SQL OR TSQL>"
```

Alternatively, to save the package locally, run:

```
scancentral package -o <path to package> -bt none -targs "-sql-language
<PL/SQL OR TSQL>"
```

To send existing package to ScanCentral SAST, run:

```
scancentral -url <controller_url> start -package <package path>
```

ScanCentral SAST returns a job ID that you can use to track the scan.

Working with Java 8 Projects

If you have a Java 8 project that fails to build because ScanCentral SAST requires Java 11 to run, set the `SCANCENTRAL_JAVA_HOME` environment variable to point Java 11. After you do, ScanCentral SAST runs successfully, and the build runs with the `JAVA_HOME` set to Java 8.

See Also

"Fortify ScanCentral SAST Command-Line Options" on page 88

"Submitting Scan Requests and Uploading Results to Fortify Software Security Center" below

Submitting Scan Requests and Uploading Results to Fortify Software Security Center

To submit a scan request, the results of which you want to upload to an application version in Fortify Software Security Center, use the fortifyclient tool to obtain the application version ID, and access tokens from Fortify Software Security Center. You can reuse the token for future requests. For information about how to use the fortifyclient tool, see the *Micro Focus Fortify Software Security Center User Guide*.

Note: The Fortify Software Security Center user account must have permission to upload scan results for the application version, and must have access to the application version on Fortify Software Security Center. A user who submits a ScanCentral SAST job for upload to a Fortify Software Security Center application version must use a token that was obtained using an account that has permission to upload scan results. If a Fortify Software Security Center user is assigned to a target application version with a view-only role, and that user requests a token and uses it to submit the job, the upload fails.

To submit a job to be uploaded to an application version in Fortify Software Security Center:

1. Open a command prompt, and then type the following command:

   ```
   fortifyclient.bat listApplicationVersions -url <ssc_url> -user <user> -password
   <pwd>
   ```

   Sample Output

   ```
   ID      Name                    Version

   10     ScanCentral Test         1.0

   12     ScanCentral Test         2.0
   ```

```
4      Bill Payment Processor   1.1

3      Logistics                2.5

2      Logistics                1.3

8      RWI                      2.0

5      RWI                      1.0
```

2. To generate a Controller token, run the following command:

```
fortifyclient.bat token -gettoken ScanCentralCtrlToken -url <ssc_url> -user
<user> -password <pwd>

Authorization Token: <..scancentralCtrlToken...>
```

3. To submit your job and upload your scan results to a Fortify Software Security Center application version, run the following command:

```
scancentral.bat -sscurl <ssc_url> -ssctoken <ScanCentralCtrlToken> start
-upload -versionid <app_version_id> -b <mybuildId> -uptoken
<ScanCentralCtrlToken> -scan
```

Note: Instead of –versionid *<app_version_id>*, you can pass –application *<application_name>* -version *<version_name>*. The *<application>* and *<version>* must match the values in Fortify Software Security Center. These values are case sensitive.

Typically, the steps above are combined into a scripted flow from a build server.

Specifying the Name of FPR Files Uploaded to Fortify Software Security Center

You can use the `-fprssc` (`--fpr-filename-on-ssc`) `start` command option to specify the name of the FPR files you upload to Fortify Software Security Center.

Example 1 (Local translation and remote scan):

```
scancentral.bat -sscurl <ssc_url> -ssctoken <ScanCentralCtrlToken> start
-upload -uptoken <ScanCentralCtrlToken> -versionid <app_version_id> -fprssc
<my_frp_name>.fpr -b <build_id> -scan
```

Note: The `-uptoken` option is required if you use the `-upload` flag.

Example 2: (Remote translation and remote scan):

```
scancentral.bat -sscurl <ssc_url> -ssctoken <ScanCentralCtrlToken> start
-upload -versionid 10 -bt <build tool: msbuild, gradle, mvn, or none>
-uptoken <ScanCentralCtrlToken> -fprssc <my_frp_name>.fpr
```

Note: The `-uptoken` option is required if you use the `-upload` flag.

The file name you specify must not contain more than 128 characters and *must not* contain the following invalid characters:

- colon (:)
- backslash (\)
- forward slash (/)
- asterisk (*)
- question mark (?)
- vertical bar or pipe (|)
- less than (<)
- greater than (>)
- double quote (")

Optimizing Scan Performance

If you plan to regularly scan large applications, Fortify recommends that you run a manual test scan on hardware that is equivalent to the hardware on which your sensor is installed.

To optimize your scan:

1. To set the Fortify Static Code Analyzer scan parameters for optimal performance, adjust the memory settings to align with your hardware.

   For information about how to tune Fortify Static Code Analyzer, see the *Micro Focus Fortify Static Code Analyzer User Guide*.
2. Run the scan.
3. Note the size of the resulting FPR file and scan log. To ensure that the ScanCentral Controller and Fortify Software Security Center can accept FPR or log files larger than 1 GB, increase the following file size threshold:
   - Navigate to the `<controller_dir>/tomcat/webapps/scancentral-ctrl` directory, open the `config.properties` file, and then set the Controller threshold as follows:

     ```
     max_upload_size=<max_fpr_or_logfile_size_in_MB>
     ```

     The default value is 1024.

4. Check to make sure that your Fortify Software Security Center hardware and application startup parameters are set to process very large FPR files. For more information, see the *Micro Focus Fortify Static Code Analyzer User Guide*.

Generating a ScanCentral SAST Package

The examples listed in the following table illustrate various ways to generate a ScanCentral SAST package. Note that the `-bt` and `-bf` options in the commands shown in the following table are not required since ScanCentral SAST detects your build tools and solutions automatically based on the project files being scanned.

Objective	Command

Create a package from a Gradle project	`scancentral package -bt gradle -o myPackage.zip`
	or
	`scancentral package -o myPackage.zip`
	In the second example command, `-bt` and `-bf` are detected automatically, based on the underlying files.

Create a package from a Maven project with a `custom pom.xml`	`scancentral package -bt mvn -bf myCustomPom.xml -o myPackage.zip`
	or
	`scancentral package -bf pom.xml -o myPackage.zip`
	In the second example command, because the `pom.xml` file is specified, ScanCentral sets `-bt` to mvn automatically.

Create a package from an MSBuild project	`scancentral package -bt msbuild -bf mySolution.sln -o myPackage.zip`
Create a package from a JavaScript/TypeScript project	`scancentral package -bt none -o myPackage.zip`
Create a package from a JavaScript/TypeScript project and include the node_modules **Caution!** This may greatly increase the package size as well as the scan time.	`scancentral package -bt none --scan-node-modules -o myPackage.zip`
Create a package from a PHP 7.1 project	`scancentral package -bt none -hv 7.1 -o`

Objective	Command
	`myPackage.zip`
Create a package from an ABAP project	`scancentral package -bt none -o myPackage.zip`
Create a package from a Ruby project	`scancentral package -bt none -o myPackage.zip`
Create a package from a Python 2 project	`scancentral package -bt none -yv 2 -pyr <path_to_requirements_file> -o myPackage.zip`
Create a package from a Python project under an active virtual environment with dependencies already installed	`scancentral package -bt none -o myPackage.zip`
Create a package from a Python project under an active virtual environment without project dependencies installed	`scancentral package -bt none -pyr <path_to_requirements_file> -o myPackage.zip`
Create a package from a Python project using an existing Python virtual environment and install project dependencies	`scancentral package -bt none -pyv <virtual_environment_location> -pyr <path_to_requirements_file> -o myPackage.zip`

Viewing Scan Request Status

To view the status of a scan request, run the following command:

```
scancentral.bat –url http://<Controller_Host>:8080/scancentral-ctrl status -token <job_token>
```

You can also view scan request status from the Fortify Software Security Center user interface. For instructions, see the *Micro Focus Fortify Software Security Center User Guide*.

Using the PackageScanner Tool

If you have Fortify Static Code Analyzer installed locally, you can run an analysis of a package locally, without first sending it to the Controller. The packagescanner tool (`packagescanner.bat` on Windows and `packagescanner` on Linux) takes a package generated using the ScanCentral SAST package command, generates Fortify Static Code Analyzer commands, and then scans it using a locally-installed Fortify Static Code Analyzer instance. The packagescanner tool is located in the `<sca_install_dir>/bin<sca_install_dir>/bin` directory.

The command-line options used with the packagescanner tool are described in the following table.

Option	Description
`-b, --build-id <id>`	(Optional) Specifies the build ID. Fortify Static Code Analyzer uses the build ID to track which files are compiled and combined as part of a build, and later, to scan those files. If you do not specify a build ID, ScanCentral SAST generates one based on language, number of projects, and so on.
`-debug`	Enables debug logging on ScanCentral SAST clients and sensors.
`-fpr`	(Required) Path of saved FPR files.
`-package`	(Required) Path to the package file generated by the ScanCentral SAST command-line interface.
`-sargs, --scan-arguments`	(Optional) Additional Fortify Static Code Analyzer scan options. Enclose multiple options in quotes separated by spaces, or repeat this option for each Fortify Static Code Analyzer option and parameter.
`-sca-path`	(Optional if started from Fortify Static Code Analyzer) Path to the Fortify Static Code Analyzer executable. If ScanCentral SAST is part of SCA and Apps, the path is determined automatically.
`--sca-scan-log`	(Optional) Log for a scan command. By default, the log file is created in a temp folder, which is removed after program execution.
`--sca-translation-log`	(Optional) Log for all translation commands. By default, the log file is created in a temp folder, which is removed after program execution.
`-targs, --translation-arguments`	(Optional) Fortify Static Code Analyzer translation options. Enclose multiple options in quotes separated by spaces, or repeat this option for each Fortify Static Code Analyzer option and

Option	Description
	parameter.
`-version`	(Optional) PackageScanner version.

Retrieving Scan Results from the Controller

To retrieve scan results, run the following command:

```
scancentral.bat -url <controller_url> retrieve -token <job_token> -f
worker.fpr -log sca.log
```

Configuring Job Cleanup Timing on Sensors

To prevent the progressive loss of disc space as job files accumulate, Fortify ScanCentral SAST sensors automatically clean up internal job files (packages received from the Controller, FPRs, logs, and so on), and Fortify Static Code Analyzer build files related to cleaned ScanCentral jobs. Although you cannot disable this feature, you can configure its timing.

To configure the timing of job file cleanup on a sensor:

1. Navigate to the `<sca_install_dir>/Core/config` directory, and then open the `worker.properties` file in a text editor.
2. Configure the following properties based on your scheduling needs.

Property Name	Description	Default Value (hours)
`worker_cleanup_age`	Age (in hours) job files must be before they are removed from the sensor working directory	168 (or, one week)
`worker_cleanup_interval`	Frequency with which the cleanup process runs	1

3. Save and close your `worker.properties` file.
4. Restart the sensor.

Cancelling Scan Requests

To cancel a scan request, run the following command:

```
scancentral.bat –url <controller_url> cancel -token <tokenid>
```

You can also cancel scan requests from the Scans view in Fortify Software Security Center. For instructions, see the *Fortify Software Security Center User Guide*.

Chapter 9: Working with ScanCentral SAST from Fortify Software Security Center

While you can deploy the Controller in standalone mode, communication with Fortify Software Security Center provides additional benefits. If Fortify Software Security Center is integrated with ScanCentral SAST, then the Fortify Software Security Center Scans view includes the ScanCentral SAST pages, which are described in the following table.

Scans View Page	Functionality
Scan Requests	View and export ScanCentral SAST scan request details Cancel prepared scan requests
Controller	View Controller information
Sensors	View sensor information
Sensor Pools	Create and manage groups of sensors to which you can target scan requests.

For detailed information, see the *Fortify Software Security Center User Guide*.

See Also

"Configuring the Connection to Fortify Software Security Center" below

Configuring the Connection to Fortify Software Security Center

While the Controller can be deployed in standalone mode, communication with Fortify Software Security Center provides additional benefits:

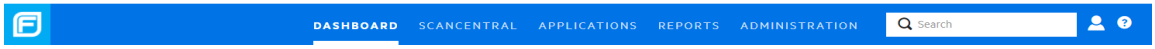
- The Fortify Software Security Center user interface includes a Scans view that makes it easy to view the status of recent scan requests.
- The Controller can upload scan results directly to Fortify Software Security Center application versions.
- You can create and manage ScanCentral SAST sensor pools from Fortify Software Security Center. (For information about sensor pools, see the *Fortify Software Security Center User Guide*.)

To integrate Fortify Software Security Center with ScanCentral SAST:

1. Log in to Fortify Software Security Center as an administrator, and then, on the Fortify header, click **ADMINISTRATION**.
2. In the left panel, select **Configuration**, and then select **ScanCentral SAST**.

   The ScanCentral SAST page opens.
3. To enable the polling of Controller to retrieve scan request status, select the **Enable ScanCentral SAST** check box.
4. In the **ScanCentral Controller URL** box, type the URL for the Controller.
5. In the **ScanCentral poll period (seconds)** box, either select or type the number of seconds to elapse between ScanCentral SAST polls.
6. In the **SSC and ScanCentral Controller shared secret** box, type the password for Fortify Software Security Center to use when it requests data from the Controller. (If you use clear text, this string must match the value stored in the Controller `config.properties` file for the `ssc_scancentral_ctrl_secret` key.

   Note: The `ssc_cloudctrl_secret` key is supported for backward compatibility with Fortify CloudScan.
7. Click **SAVE**.
8. Restart the Fortify Software Security Center server.



When you next log in to Fortify Software Security Center, notice that the Fortify header includes the **SCANCENTRAL** link.

Important! You must use the same or a later version of ScanCentral SAST as the Fortify Static Code Analyzer version installed on your clients.

See Also

"Working with ScanCentral SAST from Fortify Software Security Center" on the previous page

"Starting the ScanCentral SAST Sensors" on page 49

Appendix A: Fortify ScanCentral SAST Command-Line Options

This appendix provides information about the command-line options that you can use with Fortify ScanCentral SAST. The Fortify ScanCentral SAST options are:

Global Options

This section provides information about the command-line options that you can use with Fortify ScanCentral SAST.

Global Options	Use to:
`-debug`	Enables debug logging on ScanCentral SAST clients and sensors. For information on how to configure the logging level on the Controller, see "Configuring the Logging Level on the Controller" on page 39.

`-h <command>`
or
`--help <command>`
	Get help for the selected command. To see all command help, type `-h all`.

`-ssctoken <ScanCentralCtrlToken>`	Specify the Fortify Software Security Center authorization token.
`-sscurl <url>`	Specify the Fortify Software Security Center server URL.

Global Options	Use to:
`-url <url>`	Specify the ScanCentral SAST Controller URL.
`-version`	Get the product version.

Status Command

Use the `status` command to check the status of the Controller or a job.

Status Options	Description
`-bl, --block-until <action>`	Use this option to have the process (scanning or merging) wait until Fortify Software Security Center FPR upload and processing are complete, and then download the merged FPR from Fortify Software Security Center. Valid values are `scan` and `sscproc`. If you specify `scan`, the `status` command directs the scan process to continue to run until the scan is complete and available on the Controller. If you specify `sscproc`, the `status` command waits for Fortify Software Security Center processing to complete. If the scan result is not uploaded to Fortify Software Security Center, an error occurs.

Example commands using `--block-until`:
To wait until a scan is completed, run:
`scancentral -url <ctrl_url> status -token <job_token> --block-until scan`
To wait until a scan is done, and the FPR file is uploaded to Fortify Software Security Center and processed, run:
`scancentral -url <ctrl_url> status -token <job_token> block-until sscproc`
To wait until a scan is completed, but stop and exit if the scan is not finished withing two minutes:
`scancentral -url <ctrl_url> status -token <job_token> --block-until scan --block-timeout 2`
To wait until a scan is completed, and have the client ask the Controller for the current job status every 5 minutes (300 seconds), run:
`scancentral -url <ctrl_url> status -token <job_token> --block-until scan --poll-interval 300`	

Status Options	Description
`-bto, --block-timeout`	Specify how long (in minutes) to block processing. Valid range is from 0 to 10080. If 0 is specified, no timeout is set.
`-ctrl`	Verify that the Controller is running.
`-pi, --poll-interval`	Specify how frequently (in seconds) to poll the processing status. Valid range is from 10 to 60.
`-token, --job-token <token>`	Specify the job token to query.

Start Command

You can use the options listed in the following tables with the `start` command to perform a remote scan, or to perform a remote translation *and* scan.

Use the options listed in the following table with the `start` command to perform a remote scan.

Start Options	Description
`-application, --application <name>`	Specifies the Fortify Software Security Center application name.
`-bc, --build-command <commands>`	For use with Maven, Gradle and MSBuild. Specifies custom build parameters for preparing and building a project. For example, to invoke a Gradle build before packaging: `-Prelease=true clean customTask build` If you use the `-bc` option, and the build fails, ScanCentral stops working on the build. (Gradle only) If you *do not* use `-bc`, the default command, default tasks and target are invoked. If the build fails, ScanCentral displays a warning, but continues to work and then displays a message to indicate that the build procedure failed and your results may be incomplete.
`-b, --build-id <id>`	Specifies the build ID of the session to export.
`-bf, --build-file <file>`	Specifies the build file, unless it has a default name such as

Start Options	Description
	`build.gradle` or `pom.xml`. You cannot use this option with the `-scan` option.
`-bt, --build-tool <name>`	(Optional) Specifies the build tool name used for the project. Example: `-bt mvn -bc "package --setting custom.xml"` You cannot use this option with the `-scan` option. The `-bt` option is *not* required. Fortify ScanCentral SAST can detect the build tool automatically based on the project files being scanned.
`-email <address>`	Specifies the email address for job status notifications.
`-exclude`	Specifies the files or directories (with absolute or relative path, or Ant-style path pattern) to exclude from a package (repeatable).
`-f, --output-file <file>`	Specifies the name for the local FPR file output. Use with the `-block` option to specify the name for the local FPR file output after a scan is completed.
`-filter <file>`	Specifies the filter file to use during a scan (repeatable).
`-fprssc, --fpr-filename-on-ssc <file>`	Specifies the name to use for the FPR files uploaded to Fortify Software Security Center. The file name must not exceed 128 characters in length and *must not* contain the following invalid characters: • colon (:) • backslash (\) • forward slash (/) • asterisk (*) • question mark (?) • vertical bar or pipe (|) • less than (<) • greater than (>)

Start Options	Description
	• double quote (")
`-hv, --php-version <version>`	Specifies the PHP version.
`-log, --log-file <file>`	Use with the `-block` option to specify the name for the Fortify Static Code Analyzer log file output after a scan is completed.
`-mbs <file>`	Specifies the mobile build session to upload.
`-o, --overwrite`	Overwrites the existing FPR or log with new data.
`-p, --package <file>`	Specifies the project package file to upload.
`-pool, --submit-to-pool <uuid>`	Specifies the sensor pool into which a sensor is to be placed at startup.
`-projroot, --project-root <dir>`	Specifies the project directory for the mobile build session export.
`-projtl, --project-template <file>`	Specifies the issue template file to include.
`-pyr, --python-requirements <file>`	Specifies the Python project requirements file to install and collect dependencies.
`-pyv, --python-virtual-env <directory>`	Specifies the Python virtual environment location.
`-q, --quiet`	Prevents the printing of stdout from the build execution.
`-rules <file/dir>`	Specifies custom rules file or directory to use during the scan (repeatable).
`-sargs, --scan-args`	Fortify Static Code Analyzer scan arguments (repeatable) Takes a single string argument. For multiple scan arguments, use multiple `-sargs` options. If the scan option has a path parameter that includes a space, enclose the path with single quotes. **Note:** You cannot use the `-sargs` option with the `-scan` option. It is for use in remote translation and scan only.

Start Options	Description
`-scan`	Sets the point beyond which all arguments are for sourceanalyzer. You cannot use this option with the `--build-tool` or `--package` option.
`-snm, --scan-node-modules`	Specifies node_modules dependencies in the package. If you set `--scan-node-modules`, all third-party library scan results are added to the resulting FPR. **Tip:** Because including node_modules dependencies in a package does not greatly improve type resolution or dataflow, and can result in an excessive number of false positives, Fortify recommends that you exclude them from scans. By default, node_modules are not applied to a package unless you apply the `--scan-node-modules` option from the command line.
`-skipBuild`	Disables the project preparation build step before packaging. If you use `-skipBuild` option, the `-bc` option (if used) is ignored. **Caution!** You can apply this option to Gradle and Maven build tools, but not to MSBuild.
`-sp, --save-package <file>`	Specifies the package file to save after uploading. The file extension must be `*.zip`.
`-sto, --scan-timeout`	Specifies the maximum amount of time (in minutes) a scan job can be processed (and prevent a sensor from doing other work). **Note:** Use of this worker option has a higher priority than the `scan_timeout` property setting in the `config.properties` file.
`-t, --include-test`	Includes test source set (Gradle) or test scope (Maven) to scan (for Java projects only).
`-targs, --translation-args`	Fortify Static Code Analyzer translation arguments (repeatable)

Start Options	Description
	Takes a single string argument. For multiple translation arguments, use multiple `-targs` options. If the translation option has a path parameter that includes a space, enclose the path with single quotes. If you use the `-targs` option with the start command `+p` option, ScanCentral SAST ignores it and displays an error message. **Note:** You cannot use the `-targs` option with the `-scan` option. It is for use in remote translation and scan only. For a list of the Fortify Static Code Analyzer options you can use with the `-targs` option, see "Options Accepted for -targs (--translation-args)" on page 106.
`-upload, --upload-to-ssc`	Uploads the FPR to Fortify Software Security Center upon completion.
`-uptoken, --ssc-upload-token <token>`	Specifies the Fortify Software Security Center file upload token. **Note:** If the `pool_mapping_mode` property is set to `DISABLED` on the Controller, you can use a Fortify Software Security Center `AnalysisUploadToken` instead. However, if `pool_mapping_mode` is ENABLED, an `AnalysisUploadToken` does not work, and a `ScanCentralCtrlToken` is required instead. For information about how to acquire `AnalysisUploadToken` and `ScanCentralCtrlToken` tokens, see the *Fortify Software Security Center User Guide*.
`-version, --application-version <name>`	Specifies the Fortify Software Security Center application version name.
`-versionid, --application-version-id <id>`	Specifies the Fortify Software Security Center application version ID.
`-yv, --python-version <version>`	Specifies the Python version to automatically find the installed Python. Allowed values: 2 or 3. This flag is ignored if the ScanCentral SAST client is started under a Python virtual environment or if `-python-virtual-env` is specified.

Use the options listed in the following table with the `start` command to perform a remote translation and scan.

Start Options	Description
`-application, --application <name>`	Specifies the Fortify Software Security Center application name.
`-bc, --build-command <commands>`	For use with Maven, Gradle and MSBuild. Specifies custom build parameters for preparing and building a project. For example, to invoke a Gradle build before packaging: `-Prelease=true clean customTask build` If you use the `-bc` option, and the build fails, ScanCentral stops working on the build. (Gradle only)If you *do not* use `-bc`, the default command, default tasks and target are invoked. If the build fails, ScanCentral displays a warning, but continues to work and then displays a message to indicate that the build procedure failed and your results may be incomplete.
`-b, --build-id <id>`	Specifies the build ID of the session to export.
`-bf, --build-file <file>`	Specifies the build file, unless it has a default name such as `build.gradle` or `pom.xml`. You cannot use this option with the `-scan` option.
`-bl, --block-for`	If you use this option with the `-upload` option, the process waits until the Fortify Software Security Center upload and processing are complete, and then downloads the merged FPR from Fortify Software Security Center. Valid values are `scan` and `ssc`. If you specify scan, the start command waits for the scan to complete. If you specify `ssc`, the start command waits for Fortify Software Security Center processing to complete. If the scan result is not uploaded to Fortify Software Security Center, an error occurs.
`-bt, --build-tool <name>`	(Optional) Specifies the build tool name used for the project. Example: `-bt mvn -bc "package --setting custom.xml"`

Start Options	Description
	You cannot use this option with the `-scan` option. The `-bt` option is *not* required. Fortify ScanCentral SAST can detect the build tool automatically based on the project files being scanned.
`-email <address>`	Specifies the email address for job status notifications.
`-exclude`	Specifies the files or directories (with absolute or relative path, or Ant-style path pattern) to exclude from a package (repeatable).
`-f, --output-file <file>`	Specifies the name for the local FPR file output. Use with the `-block` option to specify the name for the local FPR file output after a scan is completed.
`-filter <file>`	Specifies the filter file to use during a scan (repeatable).
`-fprssc, --fpr-filename-on-ssc <file>`	Specifies the name to use for the FPR files uploaded to Fortify Software Security Center. The file name must not exceed 128 characters in length and *must not* contain the following invalid characters: • colon (:) • backslash (\) • forward slash (/) • asterisk (*) • question mark (?) • vertical bar or pipe (|) • less than (<) • greater than (>) • double quote (")
`-hv, --php-version <version>`	Specifies the PHP version.
`-log, --log-file <file>`	Use with the `-block` option to specify the name for the local log file output after a scan is completed.

Start Options	Description
`-mbs <file>`	Specifies the mobile build session to upload.
`-o, --overwrite`	Overwrites the existing FPR or log with new data.
`-p, --package <file>`	Specifies the project package file to upload.
`-pool, --submit-to-pool <uuid>`	Specifies the sensor pool into which a sensor is to be placed at startup.
`-projroot, --project-root <dir>`	Specifies the project directory for the mobile build session export.
`-projtl, --project-template <file>`	Specifies the issue template file to include.
`-pyr, --python-requirements <file>`	Specifies the Python project requirements file to install and collect dependencies.
`-pyv, --python-virtual-env <directory>`	Specifies the Python virtual environment location.
`-q, --quiet`	Prevents the printing of stdout from the build execution.
`-rules <file/dir>`	Specifies custom rules file or directory to use during the scan (repeatable).
`-sargs, --scan-args`	Fortify Static Code Analyzer scan arguments (repeatable) Takes a single string argument. For multiple scan arguments, use multiple `-sargs` options. If the scan option has a path parameter that includes a space, enclose the path with single quotes. **Note:** You cannot use the `-sargs` option with the `-scan` option. It is for use in remote translation and scan only.
`-scan`	Sets the point beyond which all arguments are for sourceanalyzer. You cannot use this option with the `--build-tool` or `--package` option.
`-snm, --scan-node-modules`	Specifies node_modules dependencies in the package. If you set `--scan-node-modules`, all third-party library scan results are added to the resulting FPR.

Start Options	Description
	Tip: Because including node_modules dependencies in a package does not greatly improve type resolution or dataflow, and can result in an excessive number of false positives, Fortify recommends that you exclude them from scans. By default, node_modules are not applied to a package unless you apply the `--scan-node-modules` option from the command line.
`-skipBuild`	Disables the project preparation build step before packaging. If you use `-skipBuild` option, the `-bc` option (if used) is ignored. **Caution!** You can apply this option to Gradle and Maven build tools, but not to MSBuild.
`-sp, --save-package <file>`	Specifies the package file to save after uploading. The file extension must be `*.zip`.
`-sto, --scan-timeout`	Specifies the maximum amount of time (in minutes) a scan job can be processed (and prevent a sensor from doing other work). **Note:** Use of this worker option has a higher priority than the `scan_timeout` property setting in the `config.properties` file.
`-t, --include-test`	Includes test source set (Gradle) or test scope (Maven) to scan (for Java projects only).
`-targs, --translation-args`	Fortify Static Code Analyzer translation arguments (repeatable) Takes a single string argument. For multiple translation arguments, use multiple `-targs` options. If the translation option has a path parameter that includes a space, enclose the path with single quotes. If you use the `-targs` option with the start command `+p` option, ScanCentral SAST ignores it and displays an error message.

Start Options	Description
	Note: You cannot use the `-targs` option with the `-scan` option. It is for use in remote translation and scan only. For a list of the Fortify Static Code Analyzer options you can use with the `-targs` option, see "Options Accepted for -targs (--translation-args)" on page 106.
`-upload, --upload-to-ssc`	Uploads the FPR to Fortify Software Security Center upon completion.
`-uptoken, --ssc-upload-token <token>`	Specifies the Fortify Software Security Center file upload token. **Note:** If the `pool_mapping_mode` property is set to `DISABLED` on the Controller, you can use a Fortify Software Security Center `AnalysisUploadToken` instead. However, if `pool_mapping_mode` is ENABLED, an `AnalysisUploadToken` does not work, and a `ScanCentralCtrlToken` is required instead. For information about how to acquire `AnalysisUploadToken` and `ScanCentralCtrlToken` tokens, see the *Fortify Software Security Center User Guide*.
`-version, --application-version <name>`	Specifies the Fortify Software Security Center application version name.
`-versionid, --application-version-id <id>`	Specifies the Fortify Software Security Center application version ID.
`-yv, --python-version <version>`	Specifies the Python version to automatically find the installed Python. Allowed values: 2 or 3. This flag is ignored if the ScanCentral SAST client is started under a Python virtual environment or if `-python-virtual-env` is specified.

Retrieve Command

Use the `retrieve` command to download the result of a remote scan job.

Retrieve Options	Description
`-block`	Wait for the job to complete and download the result.
`-bto, --block-timeout`	Specify how long (in minutes) to block processing. Valid range is from 0 to 10080. If 0 is specified, no timeout is set.
`-f, --output-file <file>`	Specify the file name for local FPR output. Use with the `-block` option to specify the name for the local FPR file output after a scan is completed.
`-log, --log-file <file>`	Use with the `-block` option to specify the name for the Fortify Static Code Analyzer log file output after a scan is completed.
`-o, --overwrite`	Overwrite the existing FPR or log with new data.
`-pi, --poll-interval`	Specify how frequently (in seconds) to poll the processing status. Valid range is from 10 to 60.
`-token, --job-token <token>`	Specify the job token to query.

Cancel Command

Use the `cancel` command to cancel a remote scan job.

Cancel Options	Description
`-token, --job-token <token>`	Specify the job token to query.

Worker Command

Caution! To avoid packaging failure for projects with file paths that contain an umlaut, you must first add the `com.fortify.sca.CmdlineOptionsFileEncoding` property to the

`fortify-sca.properties` file (located in the `<sca_install_dir>/Core/config` directory) and give it a value that is not encoded in ASCII.

Use the `worker` command to start or test a sensor.

Worker Options	Description
`-hello`	Sensor reporting for duty.
`-pool, --assign-to-pool`	Specifies the sensor pool to which the sensor is to be assigned after It connects to the Controller. If the sensor is already assigned to a pool, this option overrides that assignment. (If an error occurs In sensor pool assignment, the sensor shuts down.)
`-sto, --scan-timeout`	Specifies the maximum amount of time (in minutes) a scan job can be processed (and prevent a sensor from doing other work). **Note:** Use of this worker option has a higher priority than the `scan_timeout` property setting in the `config.properties` file.

Package Command

Use the `package` command to create a zip package of the specified project.

Package Options	Description
`-bc, --build-command <commands>`	Specify custom build parameters for preparing and building a project. For example, to invoke a Gradle build before packaging: `-Prelease=true clean customTask build` If you use the `-bc` option, and the build fails, ScanCentral stops working on the build. (Gradle only) If you *do not* use `-bc`, the default tasks and targets are invoked. If the build fails, ScanCentral SAST displays a warning, but continues.

Package Options	Description
	You can use this option with Maven, Gradle and MSBuild.
`-bf, --build-file <file>`	Specify the build file if you are not using a default name such as `build.gradle` or `pom.xml`.
`-bt, --build-tool <name>`	Specify the build tool name used for the project. You cannot use this option with the project.
`-exclude`	Specify the files or directories (with absolute or relative path, or Ant-style path pattern) to exclude from a package (repeatable).
`-hv, --php-version <version>`	Specify the PHP version.
`-o, --output <file>`	Specify the output file name. The file extension must be `*.zip`.
`-oss, --open-source-scan`	(Applies only to Fortify on Demand) Used to generate and collect additional files for scanning. For details see Fortify on Demand documentation.
`-pyr, --python-requirements <file>`	Specify the Python project requirements file to install and collect dependencies.
`-pyv, --python-virtual-env <directory>`	Specify the Python virtual environment location.
`-q, --quiet`	Prevent the printing of stdout from the build execution.
`-snm, --scan-node-modules`	Specifies node_modules dependencies in the package. If you set `--scan-node-modules`, all third-party library scan results are added to the resulting FPR. **Tip:** Because including node_modules dependencies in a package does not improve type resolution or dataflow results, and because they degrade translation and scan speed, Fortify recommends that you exclude them from scans. By default, node_modules

Package Options	Description
	are not applied to a package unless you apply the `--scan-node-modules` option from the command line.
`-skipBuild`	Disables the project preparation build step before packaging.
`-t, --include-test`	Include the test source set (Gradle) or test scope (Maven) to scan (for Java projects only).
`-targs, --translation-args`	Fortify Static Code Analyzer translation arguments (repeatable) Takes a single string argument. For multiple translation arguments, use multiple `-targs` options. If the translation option has a path parameter that includes a space, enclose the path with single quotes. For a list of the Fortify Static Code Analyzer options you can use with the `-targs` option, see "Options Accepted for -targs (--translation-args)" on page 106.
`-yv, --python-version <version>`	Specify the Python version to automatically find the installed Python. Allowed values: 2 or 3. This flag is ignored if the ScanCentral SAST client is started under a Python virtual environment or if `-python-virtual-env` is specified.

Arguments Command

Use the arguments command to generate a settings file for additional Fortify Static Code Analyzer command-line options. The settings file must reside in the same directory you specify ScanCentral SAST commands for remote translation and scanning.

Deprecated: As of the 23.1.0 release, the `arguments` command is deprecated.

Arguments Options	Description
`-o, --overwrite`	Overwrite the existing arguments file.
`-p, --project-dir <directory>`	Specify the project directory in which to create the Fortify Static Code Analyzer translation and scan additional arguments file.
`-sargs, --scan-args`	Fortify Static Code Analyzer scan arguments (repeatable) **Note:** The arguments command is deprecated. Fortify recommends that you use the `-sargs` option directly with the start command. For a list of the Fortify Static Code Analyzer options you can use with `-sargs`, see "Options Accepted for -sargs (--scan-args)" on page 108.
`-targs, --translation-args`	Fortify Static Code Analyzer translation arguments (repeatable) **Note:** Fortify recommends that you use the `-targs` option directly with the start or package command. The arguments command is deprecated. For a list of the Fortify Static Code Analyzer options you can use with `-targs`, see "Options Accepted for -targs (--translation-args)" on page 106.

Important! The `-targs` and `-sargs` options take a single string argument. To specify multiple translation or scan arguments, use multiple `-targs` and (or) `-sargs` options. If the translation or scan option has a path parameter that includes a space, enclose the path in single quotes.

Example: The following generates a `fortify-sca.settings` file in the current directory.

```
scancentral.bat arguments -o -targs "-Xmx4G" -targs "-cp 'myProjectDir/path
to/lib/*.jar'" -targs "-exclude 'myProject Dir/path to/src/*.js' "-sargs
"-Xms256M" -sargs "-analyzers controlflow, dataflow"
```

The resulting `fortify-sca.settings` file looks similar to the following:

```
{
  "translationArgs": [
  "-Xmx4G",
  "-cp",
  "myProject Dir/path to/lib/*.jar",
  "-exclude",
  "myProject Dir/path to/src/*.jar"
 ],
  "scanArgs": [
  "-Xms256M",
  "-analyzers",
  "controlflow,dataflow"
 ]
}
```

Progress Command

Use the `progress` command to get the progress of a Fortify Static Code Analyzer scan.

Important! If your projects are based on Java 11, and you want to use the progress command to check the progress of your scans, some minor sensor configuration is required. For instructions, see "Configuring Sensors to Use the Progress Command when Starting on Java" on page 46.

Update Command

Use the `update` command to update a client or sensor to the latest version available on the Controller. This updates a standalone client to the latest available client version. It updates an embedded client or sensor to the latest available patch version, but does not update these to the next major version.

Options Accepted for -targs (--translation-args)

The following table lists the Fortify Static Code Analyzer options you can use with the Fortify Static Code Analyzer `-targs` option.

Note: Fortify recommends that you use the `-targs` option directly with the "Start Command" on page 90.

Accepted Options: -targs	
-64	-goproxy
-autoheap	-goroot
-abap-includes	-jdk
-apex	-jdk-bootclasspath
-apex-sobject-path	-jsp-as-top-level
-apex-version	-jvm-default
-appserver	-machine-output
-appserver-home	-noextension-type

Accepted Options: -targs	
-appserver-version	-php-source-root
-bootclasspath	-php-version
-build-label	-project-root
-build-project	-python-no-auto-root-calculation
-build-version	-python-no-file-function-optimization
-cp	-python-path
-debug	-python-version
-debug-mem	-python-warnings-suppression
-debug-verbose	-quiet
-disable-java-kotlin-interop	-rubygem-path
-disable-language	-ruby-on-rails
-django-disable-autodiscover	-ruby-path
-django-template-dirs	-show-python-resolution
-document-root	-show-unresolved-symbols
-enable-language	-source-base-dir
-encoding	-source-jars
-exclude	-sourcepath
-exit-code-level	-sql-language
-extdirs	-v
-gopath	-verbose

Options Accepted for -sargs (--scan-args)

The following table lists the Fortify Static Code Analyzer options you can use with the Fortify Static Code Analyzer `-sargs` option.

Note: Fortify recommends that you use the `-sargs` option directly with the "Start Command" on page 90.

Accepted Options: -sargs	
-64	-machine-output
-autoheap	-mt
-build-label	-no-default-issue-rules
-build-project	-no-default-rules
-build-version	-no-default-sink-rules
-debug	-no-default-source-rules
-debug-mem	-p
-debug-verbose	-project-root
-disable-analyzer	-project-template
-disable-default-rule-type	-quick
-disable-filtering	-quiet
-disable-funptr-analysis	-rules
-enable-analyzer	-v
-filter	-validate
-legacy-jsp-dataflow	-verbose

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 Installation, Configuration, and Usage Guide (Fortify ScanCentral SAST 23.1.0)

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!