opentext™

OpenText™ Fortify ScanCentral SAST

Software Version: 24.4.0

Installation, Configuration, and Usage Guide

Document Release Date: Revision 1: January 2025
Software Release Date: October 2024

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 January 06, 2025. 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 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 Product Documentation website:

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

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

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

Fortify Product Feature Videos

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

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

Change log

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

Software Release / Document Version	Changes
24.4.0 / Revision 1: January 2025	Updated: • The location of the Helm charts has changed (see "Optional Kubernetes and Docker deployment" on page 18)
24.4.0	Added: • "Configuring licensing with Fortify License and Infrastructure Manager" on page 41 • "Scanning JavaScript and TypeScript code" on page 75 • Instructions for scanning PHP projects that use Composer (see "Scanning PHP projects" on page 78) • You can add JVM system and ScanCentral SAST properties (for clients and sensors) to the client commands by adding the -D option to an environment variable (see "Configuring client properties" on page 63, and "Configuring sensor properties" on page 51) Updated: • You can retrieve job files using the retrieve command (see "Retrieve command" on page 109) • You can add JVM system properties to an environment variable for use with PackageScanner (see "Using the PackageScanner tool" on page 87) • The `start` command `-uptoken` option is no longer required to upload scan results to Fortify Software Security Center if you include the global `-ssctoken` (see "Uploading results to Fortify Software Security Center" on page 79) • Described the ScanCentral SAST Controller service account for uploading scan results to Fortify Software Security Center (see "Uploading results to Fortify Software Security Center" on page 79)

Software Release / Document Version	Changes
	• Added supported `-gotags` option (see "Options accepted for -targs (--translation-args)" on page 107) • Added supported `-bin` option (see "Options accepted for -sargs (--scan-args)" on page 108) Removed: • The `-hello` option for the `worker` command is ignored and was removed from this document. This option will be removed from the product in a future release. • The `--scan-node-modules` option for the `start` and `package` commands is ignored and was removed from this document. This option will be removed from the product in a future release.
24.2.0	Added: • Option to replace duplicate scan requests that are uploaded to the same application version in Fortify Software Security Center (see "Configuring the Controller" on page 29, and "Preventing replacement of duplicate scan requests" on page 82) • Option to configure the Controller to assign scan jobs to a specific version of Fortify Static Code Analyzer (see "Configuring the Controller" on page 29) • (For use with Fortify on Demand only) Ability to use the Debricked CLI for open source software composition analysis (see "Generating a ScanCentral SAST package" on page 84, "Open source software composition analysis (Fortify on Demand only)" on page 87, and "Package command" on page 104) Updated: • MSBuild and dotnet build logs are included in the debug archive (see "Creating a log archive for Customer Support" on page 97) • The options to display the version are `-v` and `--version`. The `-version` option is deprecated (see "Global options" on page 98) • The `--php-version` option for the `start` and `package` commands is no longer required because Fortify ScanCentral SAST automatically detects the installed PHP version (see "Start command" on page 99

Software Release / Document Version	Changes
	and "Package command" on page 104) • The option `--output` for the `package` command is no longer required (see "Package command" on page 104) Removed: • The "Working with Salesforce Apex Projects" topic was removed because the `-apex` Fortify Static Code Analyzer option is no longer required to analyze Apex projects.
23.2.0	Added: • "Optional Kubernetes and Docker deployment" on page 18 • "Installing the Controller as a service on Linux" on page 23 • "Managing the Controller service on Linux" on page 24 • "Fortify ScanCentral SAST API" on page 43 • "Scanning COBOL projects" on page 78 and added supported COBOL-related options (see "Options accepted for -targs (--translation-args)" on page 107) • "Retrying failed uploads to Fortify Software Security Center" on page 82 • "Preserving the Fortify Static Code Analyzer project root directory" on page 95 • "Creating a log archive for Customer Support" on page 97 Updated: • Changed the requirements for when to run the migration script to upgrade the ScanCentral SAST Controller (see "Upgrading the Controller" on page 65) • Updates for analyzing .NET projects (see "Configuring sensors for remote translation of .NET languages" on page 49 and "Scanning .NET projects" on page 73) • Added descriptions of the scan status values (see "Viewing the scan request status" on page 90) • Added supported `-scan-policy` option (see "Options accepted for -sargs (--scan-args)" on page 108)

Software Release / Document Version	Changes
	• Added supported COBOL options (see "Options accepted for -targs (--translation-args)" on page 107) Removed: • The `arguments` command is deprecated and removed from this document. Use the `-targs` or `-sargs` option with the `start` or `package` commands instead.
23.1.0	Added: • "Securing Tomcat server" on page 17 • "Configuring where to generate job files and the worker_persist.properties file" on page 50 Updated: • Instruct users to select the Fortify ScanCentral SAST Client check box during the Fortify Static Code Analyzer installation (see "Installing a sensor using Fortify Static Code Analyzer" on page 45) • Changed the `--block-for` option to `--block-until` (see "Starting the sensors" on page 52) • Added the `client_zip_location`, `ssc_restapi_connect_timeout`, and `ssc_restapi_read_timeout` properties for configuring the Controller (see "Configuring the Controller" on page 29) • Added content to describe the procedures to configure timeout between the Controller and sensors, between the Controller and clients, and between the Controller and Fortify Software Security Center (see "Avoiding read timeout errors" on page 40) • Added information about auto-detection of the build tool (see "Submitting remote translation and scan requests" on page 71) • Added the `--block-timeout`) and `--poll-interval` options (see "Status command" on page 108 and "Retrieve command" on page 109) • Added a list of accepted Fortify Static Code Analyzer options (see "Options accepted for -targs (--translation-args)" on page 107 and "Options accepted for -sargs (--scan-args)" on page 108)

Software Release / Document Version	Changes
	Removed: • Configuring the Logging Level for Sensors • Removed the `lim_proxy_server`, `remote_ip_proxy_header`, and `ssc_trusted_proxies_remote_ip` properties for configuring the Controller

Chapter 1: Introduction

With Fortify ScanCentral SAST, OpenText™ Fortify Static Code Analyzer users can better manage their resources by offloading code analysis tasks from their build machines to a distributed network of computers (sensors) configured for this purpose. In addition to freeing up build machines, this process enables you to add more resources to the scan machines as needed, without having to interrupt the build process. The command-line 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 pipelines with respect to running scans.

There are two ways to start a Fortify Static Code Analyzer analysis of your code from a ScanCentral SAST client:

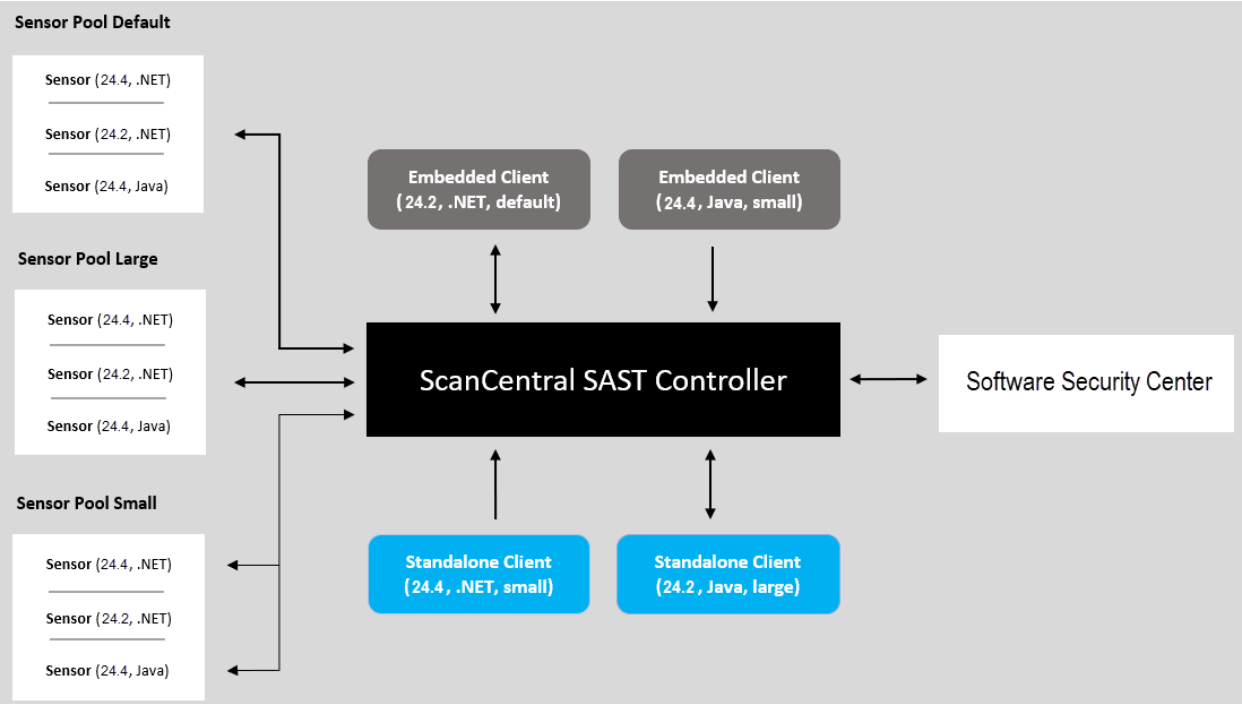
- Remote Translation and Scan—Offload the entire analysis to the sensors. Your application must be written in a language supported for remote translation. For a list of supported languages, see the *Fortify Software System Requirements* document. If your code is written in a language other than those supported in remote translation, then you must perform a local translation and remote scan.
- Local Translation and Remote Scan—Perform the translation phase (less processor- and time-intensive than the scan phase) on a local or build machine. After the translation is complete, use ScanCentral SAST client to move the Fortify Static Code Analyzer mobile build session (MBS) to sensors to scan.

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

This section contains the following topics:

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 Controller, a sensor, and a client. An OpenText™ Fortify Software Security Center server is optional.

- **ScanCentral SAST Controller**—A standalone web application that receives project packages with translation and scan instructions (or Fortify Static Code Analyzer mobile build sessions (MBS) and scan instructions from ScanCentral SAST clients), routes the information to sensors, and (optionally) uploads scan results (FPR files) to Fortify Software Security Center. For more detail, see "About the Fortify ScanCentral SAST Controller" on page 21.
- **ScanCentral SAST sensors**—A 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 project package that contains sources and dependencies, which it translates and scans. For more detail, see "About Fortify ScanCentral SAST sensors" on page 45.

  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. For 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.

- **ScanCentral SAST client**— On a build machine, clients can generate packages for remote translation and scan independent of Fortify Static Code Analyzer. Clients can also be run on 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 options, are uploaded to the ScanCentral SAST Controller for analysis. For more detail, see "About Fortify ScanCentral SAST clients" on page 57.

To successfully deploy Fortify ScanCentral SAST, 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

  For more information, see "Working with Fortify Software Security Center" below.
- Install the Fortify ScanCentral SAST Controller
- Install Fortify ScanCentral SAST sensors
- Install Fortify ScanCentral SAST clients

The following sections provide instructions for completing these tasks. For information about hardware and software requirements for these components, see the *Fortify Software System Requirements* document.

Working with Fortify Software Security Center

Although you can deploy a standalone ScanCentral SAST Controller, communication with Fortify Software Security Center provides the following additional benefits:

- The ScanCentral SAST Controller can upload scan results directly to Fortify Software Security Center application versions.
- The Fortify Software Security Center user interface includes a ScanCentral view where you can:
  - View Fortify ScanCentral SAST Controller and sensor information
  - View scan request details and export scan results and log files
  - Create and manage Fortify ScanCentral SAST sensor pools to which you can target scan requests
  - Prioritize and cancel scan requests
  - Place the Fortify ScanCentral SAST Controller in maintenance mode (see "Placing the Controller in maintenance mode" on page 42)
  - Shut down sensors (see "Safely shutting down sensors" on page 56)

For instructions on how to integrate Fortify ScanCentral SAST with Fortify Software Security Center, see the *OpenText™ Fortify Software Security Center User Guide*.

Securing Fortify ScanCentral SAST deployment

The Fortify Software products collect and display 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 Fortify ScanCentral SAST components. The security vulnerability summaries that OpenText products provide might mandate an even higher level of secure deployment.

Fortify ScanCentral SAST works with your codebase. Because this information allows for some opportunities of mishandling or abuse, OpenText recommends that you deploy Fortify ScanCentral SAST in a secure operations facility and secure access to the Fortify 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. OpenText 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 secure cipher suites

OpenText recommends that you make weak SSL/TLS cipher suites unavailable 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 the Apache server documentation for Apache Module mod_ssl and Cipher Suites and Enforcing Strong Security.

JSSE-based SSL connections

If you use a JSSE-based SSL connection, use the `ciphers` and the `honorCipherOrder` attributes. For details, see the Apache Tomcat 10 Configuration Reference. 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 choose one (see the Apache Tomcat wiki space).

Optional Kubernetes and Docker deployment

This guide describes how to install Fortify ScanCentral SAST without using a Kubernetes cluster or Docker. To use Kubernetes for ScanCentral SAST container orchestration, Helm charts are available on Docker Hub at https://hub.docker.com/r/fortifydocker/helm-scancentral-sast.

OpenText provides Fortify ScanCentral SAST Docker images that are available for download the Docker Hub. Access to the Fortify Docker repository requires credentials and is granted through your Docker ID. To access the Fortify Docker repository, email your Docker ID to mfi-fortifydocker@opentext.com.

Related Documents

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

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

All Products

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

Document / File Name	Description
About Fortify 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 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.

Document / File Name	Description
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. This document is available on the Product Documentation website at https://www.microfocus.com/documentation/fortify-software-security-center.

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

Fortify Static Code Analyzer

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

Document / File Name	Description
OpenText™ Fortify Static Code Analyzer User Guide SCA_Guide_<*version*>.pdf	This document describes how to install and use Fortify Static Code Analyzer to scan code on many of the major programming platforms. It is intended for people responsible for security audits and secure coding.

Document / File Name	Description
OpenText™ Fortify Static Code Analyzer Custom Rules Guide SCA_Cust_Rules_Guide_<*version*>.zip	This document provides the information that you need to create custom rules for Fortify Static Code Analyzer. This guide includes examples that apply rule-writing concepts to real-world security issues. **Note:** This document is included only with the product download.
OpenText™ Fortify License and Infrastructure Manager Installation and Usage Guide LIM_Guide_<*version*>.pdf	This document describes how to install, configure, and use the Fortify License and Infrastructure Manager (LIM), which is available for installation on a local Windows server and as a container image on the Docker platform.

Chapter 2: About the Fortify ScanCentral SAST Controller

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

This section contains the following topics:

Installing the Controller

For information about how to upgrade your Controller, see "Upgrading Fortify ScanCentral SAST components" on page 64 and "Upgrading the Controller" on page 65.

Important!

- Before you install the Controller, you must first download and configure a supported Java Runtime Environment (JRE). For information about supported JRE versions, see the *Fortify Software System Requirements* document. For information about how to download and configure a JRE, see the documentation for the supported JRE version.
- To install the Controller as a Windows or Linux service, make sure that you extract the contents in a directory where the local service (Windows) or the user or group using the service (Linux) has access.
- The name of the directory into which you install the Controller must not include spaces.

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

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

In this guide, *<controller_install_dir>* refers to the Controller installation directory and *<sca_install_dir>* refers to the Fortify Static Code Analyzer installation directory.

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

```
bin/
db-migrate/
tomcat/
readme.txt
```

Installing the Controller as a Windows service

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

1. Log on to Windows as a local user with administrator permissions.
2. Make sure that the `JRE_HOME` and `JAVA_HOME` environment variables are correctly configured.
3. Make sure that the `CATALINA_HOME` environment variable is either empty or set up to point to the *<controller_install_dir>*`\tomcat` directory.
4. Navigate to the *<controller_install_dir>*`\tomcat\bin` directory, and then run the following:

   ```
   service.bat install
   ```

This creates a service with the name `Tomcat10`.

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

1. Make sure that the `JRE_HOME` and `JAVA_HOME` environment variables are correctly configured.
2. Make sure that the `CATALINA_HOME` environment variable is either empty or set up to point to the *<controller_install_dir>*`\tomcat` directory.
3. Navigate to the *<controller_install_dir>*`\tomcat\bin` directory, and then run the following:

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

   Important! The service name must not contain any spaces.

Configuring Java memory for the service

To configure the Java memory for the Controller service:

1. Run `tomcat10w.exe`.
2. In the Apache Tomcat Properties window, click the **Java** tab, and then set the **Maximum memory pool** value.
3. Restart the service.

Uninstalling the Controller Windows service

To uninstall the Apache Tomcat 10 service for the Controller:

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

```
service.bat remove
```

To uninstall the Controller as a service with a name other than Tomcat10:

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

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

Installing the Controller as a service on Linux

You can install the Fortify ScanCentral SAST Controller as a service on Linux. The instructions in this topic provide an example of one method of installing the Controller as a service.

To install the Controller as a service on a Linux system:

1. Install the Controller in a location where the user and group using the service has access.

   For installation instructions, see "Installing the Controller" on page 21.
2. Configure the Controller service by creating a systemd unit file scancentral.service in the /etc/systemd/system directory with the following content.

   In the following content, replace *<controller_install_dir>* with the directory where you installed the Controller in step 1. Replace *<path_to_jre>* with the location of your JRE.

```
[Unit]
Description=ScanCentral SAST Controller Service
After=syslog.target network.target

[Service]
Type=forking
#User to run ScanCentral SAST Controller. If commented out, the root user is used.
#User=sc_user
#Group to run ScanCentral SAST Controller. If commented out, the root group is used.
#Group=sc_user

#Specify the location of JRE
```

```
Environment=JAVA_HOME=<path_to_jre>
Environment=CATALINA_PID=<controller_install_dir>/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=<controller_install_dir>/tomcat
Environment=CATALINA_BASE=<controller_install_dir>/tomcat
#Uncomment and specify CATALINA_OPTS if needed
#Environment=CATALINA_OPTS=
#Uncomment and specify JAVA_OPTS if needed
#Environment=JAVA_OPTS=

ExecStart=<controller_install_dir>/tomcat/bin/startup.sh
ExecStop=/bin/kill -15 $MAINPID

[Install]
WantedBy=multi-user.target
```

3. Reload the daemon to discover and load the new service file:

```
systemctl daemon-reload
```

4. Enable the service to start on startup by running the following command:

```
systemctl enable scancentral.service
```

See also

"Managing the Controller service on Linux" below

Managing the Controller service on Linux

To manage the Fortify ScanCentral SAST Controller service, run the following command:

```
service scancentral [start | stop | restart | status]
```

or you can use Systemd directly:

```
systemctl [start | stop | restart | status] scancentral
```

See also

"Installing the Controller as a service on Linux" on the previous page

Specifying the Controller web address

In this guide, *<controller_url>* refers to a correctly formatted Fortify ScanCentral SAST web address. The correct format for the Controller web address is as follows:

<protocol>://*<controller_host>*:*<port>*/scancentral-ctrl

Securing the Controller

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

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

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:

```
keytool -genkey -alias <alias_name> -keyalg RSA -keystore <mykeystore>
```

2. Provide values for the prompts as described in the following table.

Prompt	Description
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:** To provide an IP address as the hostname, you must also provide the `-ext san=ip:<ip_address>` option to keytool. Without this additional option, the SSL handshake fails.
What is the name of your organizational unit?	Name to identify the group that is to use the certificate.
What is the name of your organization?	Name of your organization.
What is the name of	City or locality in which your organization is located.

Prompt	Description
your City or Locality?	
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?	For example, if your server is 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 or press **Enter** to use the same password you established for your keystore. OpenText 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 the following:

```
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. Open the *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file in a text editor:
6. Update the `this_url` property with your HTTPS address and port as shown in the following 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 Fortify ScanCentral SAST clients and sensors, see "Installing clients" on page 58 and, "Installing sensors" on page 45, 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 Fortify ScanCentral SAST embedded clients and sensors, navigate to the *<sca_install_dir>*/jre/bin directory where *<sca_install_dir>* is the directory where the sensor or client is installed.
   b. For an installation of standalone Fortify ScanCentral SAST clients, type one of the following commands:
      - On a Windows system: `cd %JAVA_HOME%\jre\bin`
      - On a Linux system: `cd $JAVA_HOME/jre/bin`
   c. Run the following command:

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

      where `YourCertFile.cer` is the same certificate file that you exported in step 3.

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:

   ```
   keytool -genkey -alias <alias_name> -keyalg RSA -keystore <mykeystore>
   ```

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

Prompt	Description
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:** To enter an IP address as the hostname, you must also pass an additional option to keytool, `-ext san=ip:<ip_address>`. Without this additional option,

Prompt	Description
	the SSL handshake fails.
What is the name of your organizational unit?	Type the name of the group that is to use the certificate.
What is the name of your organization?	Type the name of your organization.
What is the name of your City or Locality?	Type the city or locality.
What is the name of your State or Province?	Type the state or province.
What is the two-letter country code for this unit?	If your server is 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. OpenText 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:

   ```
   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. After you receive your certificate from the certificate signing authority, import it into the keystore that you created, as follows:

   ```
   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.

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

6. Add the following Connector element 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: The default `server.xml` file installed with Tomcat includes an example `<connector>` element for an SSL connector.

7. Restart Tomcat server.
8. Open *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file in a text editor.
9. Update the `this_url` property with your HTTPS address and port as shown in the following example:

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

Configuring the Controller

After you install the Fortify ScanCentral SAST Controller, edit global properties such as the email address to use, the shared secret for the Controller (password that Fortify Software Security Center uses when it requests data from the Controller), the shared secret for clients, and the Fortify Software Security Center web address.

Caution! To avoid potential conflicts, OpenText 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. Open the *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file in a text editor.

2. Configure the properties described in the following table.

Property	Description
`client_auth_token`	Specifies 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 the Controller" on page 39.
`client_auto_update`	If set to `true`, the Controller automatically updates all outdated sensors and clients. For details, see "Enabling automatic updates of clients and sensors" on page 68.
`client_zip_location`	Specifies the location of the directory that contains Fortify ScanCentral SAST client ZIP files. To enable remote upgrades of one or more client versions, place them in this directory. The default value is `client_zip_location=${catalina.base}/client`.
`db_dir`	Specifies the Fortify ScanCentral SAST database home directory. The default value is `${catalina.base}/cloudCtrlDb`.
`job_file_dir`	Specifies the job storage directory. The default value is: `${catalina.base}/jobFiles`.
`fail_job_if_ssc_upload_data_invalid`	If set to `true`, then before the Controller creates a scan job and assigns it to a sensor, it verifies that the following requirements are true: • The token has not expired If the token expires before the Controller assigns the scan job to a sensor, the scan does not run and the job fails. • The application version exists in Fortify Software Security Center and is active The default value for this property is `true`.

Property	Description
`job_expiry_delay`	Specifies the 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, the jobs are deleted from the Controller after 168 hours (or 7 days).
`worker_expiry_delay`	Specifies amount of time (in hours) after a sensor stops communicating that it becomes a candidate for cleanup. The default is 168 hours (or 7 days).
`cleanup_period`	Specifies the frequency (in minutes) that expired jobs and sensors are cleaned up. The default is `60`.
`lim_server_url`	Specifies the web address for the OpenText™ Fortify License and Infrastructure Manager (LIM) server website. The web address format is `https://<location>:<port>`, where *<location>* is IP address, hostname, or domain name. **Note:** If the LIM does not use SSL certificates, the protocol is `http`. For more information about using the LIM for sensors, see "Configuring licensing with Fortify License and Infrastructure Manager" on page 41.
`lim_license_pool`	Specifies the name of the LIM license pool.
`lim_license_pool_password`	Specifies the password for the LIM license pool. You can either use a plain text password or use the `pwtool_keys_file` property to encrypt this password. For information about how to encrypt your passwords, see "Encrypting the shared secret on the Controller" on page 39.
`lim_proxy_url`	Specifies the proxy server to access the LIM server if the sensor is behind a proxy.
`lim_proxy_user`	Specifies the LIM proxy user name if authentication is required for the LIM proxy server. For information about how to encrypt user names and passwords, see "Encrypting the shared secret on the Controller" on

Property	Description
	page 39.
`lim_proxy_password`	Specifies the password for the LIM proxy user. You can either use a plain text password or use the `pwtool_keys_file` property to encrypt this password. For information about how to encrypt your passwords, see "Encrypting the shared secret on the Controller" on page 39.
`max_upload_size`	Specifies the maximum size (in megabytes) for files uploaded to the Controller from clients or sensors (for example, log files, result files, and job files).
`pool_mapping_mode`	Configures the mode for mapping scan requests to sensor pools. For information about the valid values for `pool_mapping_mode`, see "Specifying how the Controller maps scan requests to sensor pools" on page 38.
`pwtool_keys_file`	Specifies the path to a file with pwtool keys. If encrypted passwords are used, this must specify a file with the pwtool keys used to encrypt the passwords.
`scan_timeout`	Specifies the maximum amount of time (in minutes) that sensors can process a scan job and be prevented from doing other jobs. After the specified time has passed, a scan job is canceled. This setting applies to all sensors associated with the Controller but can be overridden with the `--scan-timeout` command-line option for a specific job or sensor (see "Setting the maximum run time for scans" on page 48 and "Start command" on page 99).
`accept_job_when_no_sensor_available`	Determines whether to accept scan requests if no compatible sensors (or compatible versions) are available. The default value is `true`. Also see "sensor_version_for_all_jobs" on the next page. In the following examples, the property is set to `false`: • If a version 24.2 client submits a scan request, and only version 24.4 sensors are available, the scan request is rejected. • If a client submits a request to scan a .NET application and no .NET sensors are available, the scan request is rejected.

Property	Description
`sensor_version_for_all_jobs`	Specifies the version (*<year>.<quarter>* portion only) of the sensor to which the Controller assigns scan jobs for remote translation and scan. For example, if this property is set to 24.4, then scan requests from 23.2, 24.2, or 24.4 version clients are assigned to a 24.4 version sensor. If the ScanCentral SAST client version is later than the sensor version specified in this property, then the Controller assigns jobs to the sensor version that matches the client version. For example, if this property is set to 24.2, a scan request from a 24.4 version client is assigned to a 24.4 sensor. If this property is not set (default), remote translation and scan jobs are assigned to a sensor with the same version as the ScanCentral SAST client.
`from_email`	Specifies the outgoing email address that the Controller uses to send notifications.

`email_allow_list`	Specifies a comma-, colon-, or semicolon-separated list of email domains to which the Controller can send notifications. Examples of valid values for this property:
	`*@yourcompanyname.com`
	`*@*yourcompanyname.com`
	`a*@yourcompanyname.com`
	`name1@yourcompanyname.com,name2@yourcompany.com`

`email_deny_list`	Specifies a comma-, colon, or semicolon-separated list of email domains to which the Controller cannot send notifications. Examples of valid values for this property:
	`*@yourcompanyname.com`
	`*@*yourcompanyname.com`
	`a*@yourcompanyname.com`
	`name1@yourcompanyname.com,name2@yourcompany.com`

`smtp_host`	Specifies the SMTP server host name.
`smtp_port`	Specifies the SMTP server port number.

Property	Description
`smtp_auth_user`	If your SMTP server requires authentication, uncomment both the `smtp_auth_user` and `smtp_auth_pass` properties and set their values.
`smtp_auth_pass`	You can either use a plain text password or use the `pwtool_keys_file` property to encrypt the password for `smtp_auth_pass`. For information about how to encrypt this password, see "Encrypting the shared secret on the Controller" on page 39.
`smtp_ssl`	If set to `true`, the Controller uses SSL for connections to the SMTP server. By default, the Controller does not use SSL.
`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`, the SMTP server identity is not checked. Otherwise, the Controller checks server identity as specified by RFC 2595 (the default).
`use_starttls`	If set to `true`, uses 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`.
`ssc_lockdown_mode`	If set to `true`, ScanCentral SAST clients must work with the Fortify ScanCentral SAST Controller through Fortify Software Security Center. Jobs must be uploaded to an application version and users cannot manually assign scans to specific sensor pools. In SSC lockdown mode, you: • Cannot use the `start` command `-url` option, but must use the `-sscurl` and `-ssctoken` options instead • Must specify the application name and version, or the application version ID, and the `-upload` option when starting the scan • Cannot use the `-pool` option, because the job is automatically assigned to the pool configured for the specified application version

Property	Description
`ssc_ctrl_account_username`	Specifies the user name of a ScanCentral SAST Controller service account created in Fortify Software Security Center with the **ScanCentral SAST Controller** role. For information about how the Controller uses this account, see "Uploading results to Fortify Software Security Center" on page 79. For information about how to encrypt this value, see "Encrypting the shared secret on the Controller" on page 39.
`ssc_ctrl_account_password`	Specifies the password for the ScanCentral SAST Controller service account. For information about how to encrypt this value, see "Encrypting the shared secret on the Controller" on page 39.
`ssc_remote_ip`	Specifies the 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`	Specifies the remote IP HTTP header, where the Fortify Software Security Center remote IP is found if the `ssc_remote_ip_trusted_proxies_range` property is set. The default value is `X-FORWARDED-FOR`.
`ssc_remote_ip_trusted_proxies_range`	Specifies the remote IP range (in CIDR format). Set this property if Fortify Software Security Center accesses the Controller using a (reverse) proxy server. You can specify comma-separated IP addresses or CIDR network ranges. This is unavailable by default, which means that `ssc_remote_ip_header` is never used to retrieve the remote IP address for Fortify Software Security Center.
`ssc_restapi_connect_timeout`	Specifies the Fortify Software Security Center connection timeout (in milliseconds). The default is 10000. You can use this, and the `ssc_restapi_read_timeout` property to resolve timeout errors between the Controller and Fortify Software Security Center.
`ssc_restapi_read_timeout`	Specifies the Fortify Software Security Center read timeout (in milliseconds). The default value is `130000`. You can use this property and the `ssc_restapi_connect_timeout` property to resolve timeout errors

Property	Description
	between the Controller and Fortify Software Security Center.
`ssc_scancentral_ctrl_secret`	Specifies the password that Fortify Software Security Center uses to request data from the Controller. Use a string that contains no spaces or backslashes. For instructions on how to encrypt this shared secret value, see "Encrypting the shared secret on the Controller" on page 39.

`ssc_url`	Specifies the web address 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>`

`replace_duplicate_scans`	If set to `true`, ScanCentral SAST replaces a pending scan request with a newer scan request if it is a duplicate. A duplicate scan request occurs if you have more than one scan request that uploads scan results to the same application version in Fortify Software Security Center. The Controller places the new scan request in the same queue position as the one it replaced. Any existing duplicate scan requests in the pending state are automatically canceled. The scan requests are run sequentially to maintain the submission order. This is typically useful if you submit ScanCentral SAST scans with upload as part of your build process, which might cause a large queue of unnecessary scan requests that can cause delays for the sensors to process. The default value for this property is `false`. You can override the replacement of duplicate scan requests for specific scans. For more information, see "Preventing replacement of duplicate scan requests" on page 82.
`ssc_upload_retry_count`	Specifies the maximum number of times the Controller can retry to upload scan results after an upload fails. The default value is `5`. For more information, see "Retrying failed uploads to Fortify Software Security Center" on page 82.
`ssc_upload_retry_interval`	Specifies the amount of time (in seconds) the Controller waits after a failed upload before it tries again. The default is 120 seconds (or 2 minutes). For more information, see "Retrying failed uploads to Fortify Software Security Center" on page 82.

Property	Description
`swagger_username`	Specifies the user name for access to the Fortify ScanCentral SAST API documentation. For information about how to encrypt this value, see "Encrypting the shared secret on the Controller" on page 39.
`swagger_password`	Specifies the password for access to the Fortify ScanCentral SAST API documentation. You can either use a plain text password or use the `pwtool_keys_file` property to encrypt this password. For information about how to encrypt this password, see "Encrypting the shared secret on the Controller" on page 39.
`this_url`	Specifies the web address for the Controller; used in emails to refer to this server for manual job result downloads. Example: `https://<controller_host>:8443/scancentral-ctrl`
`worker_auth_token`	Specifies a string that contains no spaces or backslashes 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 this value, see "Encrypting the shared secret on the Controller" on page 39.
`worker_inactive_delay`	Specifies the amount of time (in minutes) after which a non-communicating sensor is considered inactive and all jobs are marked as faulted. Assign a value that is much larger than `worker_stale_delay`. Note that this property uses different time units than `worker_stale_delay`.
`worker_stale_delay`	Specifies the amount of time (in seconds) after which a non-communicating sensor is considered inactive. 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 Controller" on page 41.

See also

"Installing the Controller" on page 21

"Stopping the Controller" on page 43

"Placing the Controller in maintenance mode" on page 42

"Configuring job cleanup timing on sensors" on page 51

How the Controller assigns scan requests to sensors

The Fortify ScanCentral SAST Controller accepts scan requests and assigns them a sensor of the same version. For example, if a 24.4.0 client submits a scan request, the Controller can assign the job to a version 24.4.0, 24.4.1, or 24.4.2 sensor unless a specific sensor version is specified with the `sensor_version_for_all_jobs` property (see "Configuring the Controller" on page 29).

Specifying how the Controller maps scan requests to sensor pools

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

- `disabled`— In this mode, a Fortify 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. Alternatively, a 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 Fortify 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 Controller targets the default sensor pool for scan requests. A ScanCentral SAST client cannot request a specific sensor pool in the enforced mode.

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

The following table shows how the Fortify Software Security Center integration with Fortify ScanCentral SAST responds to different input when the `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 sensor pool.

Input	Disabled	Enabled	Enforced
No pool or version specified	Default sensor pool	Default sensor pool	Default sensor pool

Input	Disabled	Enabled	Enforced
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 Controller" on page 29

Encrypting the shared secret on the Controller

Values exist in the ScanCentral Controller configuration file as plain text. You can encrypt the passwords, authentication tokens, and other values for the following properties:

- `client_auth_token`
- `lim_license_pool_password`
- `lim_proxy_password`
- `lim_proxy_user`
- `smtp_auth_pass`
- `ssc_ctrl_account_username`
- `ssc_ctrl_account_password`
- `ssc_scancentral_ctrl_secret`
- `swagger_password`
- `swagger_username`
- `worker_auth_token`

To encrypt a shared secret on the Controller:

1. At the command prompt, type the following:

   ```
   <controller_install_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 the Controller is different from the pwtool key file you use to encrypt secrets on sensors.

   The pwtool generates a new key stored in the file on the path specified in step 1 or reuses an existing file on the specified path.
3. Copy the encrypted secret, and paste it as the value for the property you want to encrypt in the `config.properties` file.

   Tip: OpenText recommends that you assign separate, unique shared secrets for the `client_auth_token`, `smtp_auth_pass`, `ssc_scancentral_ctrl_secret`, and `worker_auth_token` properties.
4. To create additional encrypted shared secrets, repeat steps 1 through 3 for each property value you want to encrypt.
5. Uncomment the following property in the `config.properties` file:

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

See also

"Configuring the Controller" on page 29

Avoiding read timeout errors

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

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

1. On the Controller, open the *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file in a text editor.
2. Increase the value of the `ssc_restapi_connect_timeout` and `ssc_restapi_read_timeout` properties to an acceptable threshold (in milliseconds).
3. Save the changes.

To configure the connection timeout between the Controller and a sensor:

1. On the sensor machine, open the *<sca_install_dir>*`/Core/config/worker.properites` file in a text editor.

2. Uncomment the `restapi_connect_timeout` and `restapi_read_timeout` properties, and then set the value of each to an acceptable threshold (in milliseconds).
3. Save the changes.

To configure the connection timeout between the Controller and a client:

1. On the client machine, open the *<client_install_dir>*`/Core/config/client.properties` file in a text editor.
2. Uncomment the `restapi_connect_timeout` and `restapi_read_timeout` properties, and then set the value of each to an acceptable threshold (in milliseconds).
3. Save the changes.

Configuring licensing with Fortify License and Infrastructure Manager

ScanCentral SAST sensors can run Fortify Static Code Analyzer with Fortify License and Infrastructure Manager (LIM). With a LIM managed concurrent license, multiple sensors can share a single license. When a scan job is completed, canceled, times out, or fails, the license is released.

For information about how to set up the LIM with licenses for Fortify Static Code Analyzer, see *OpenText™ Fortify License and Infrastructure Manager Installation and Usage Guide*. To configure the ScanCentral SAST Controller to use LIM, specify the LIM properties as described in "Configuring the Controller" on page 29.

Starting the Controller

You can start the Fortify ScanCentral SAST Controller manually or set it to start automatically, as a service. For information about how to start the Controller automatically, see "Installing the Controller as a Windows service" on page 22.

To start the Controller manually:

1. To upload your scan results to Fortify Software Security Center, 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:
3. At the command prompt, run one of the following commands:
   - On a Windows system, run `startup.bat`.
   - On a Linux system, run `./startup.sh`.

   If Tomcat is running as a service, rather than running the startup command, you can just start the service.

See also

"Placing the Controller in maintenance mode" on the next page

Placing the Controller in maintenance mode

An abrupt shutdown of the Fortify 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.

1. Log on to Fortify Software Security Center as an administrator and open the ScanCentral view.
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 Controller" on the previous page

"Safely shutting down sensors" on page 56

"Removing the Controller from maintenance mode" below

Removing the Controller from maintenance mode

To remove the Fortify ScanCentral SAST Controller from maintenance mode:

1. Log on to Fortify Software Security Center as an administrator and open the ScanCentral view.
2. In the left pane of the SAST page, select **Controller**.
3. Click **END MAINTENANCE MODE**.

See also

"Placing the Controller in maintenance mode" above

"Stopping the Controller" on the next page

Stopping the Controller

You can stop the Controller immediately using the following procedure. However, OpenText strongly recommends that you first place the Controller in maintenance mode to preserve any scans that are running.

To stop the Fortify ScanCentral SAST Controller:

1. On the machine where the Controller is installed, navigate to the Tomcat `bin` directory:
2. Type one of the following commands:
   - On a Windows system: `shutdown.bat`
   - On a Linux system: `./shutdown.sh`

See also

"Placing the Controller in maintenance mode" on the previous page

"Removing the Controller from maintenance mode" on the previous page

"Safely shutting down sensors" on page 56

Fortify ScanCentral SAST API

The Fortify ScanCentral SAST provides a RESTful API that enables you perform tasks described in the following table. The tasks are grouped by the grouping in the API Documentation (Swagger UI).

Tasks you can perform	Request group
Retrieve the scan requests from the Controller, report job status, and upload artifacts	sensor-controller
Work with scan jobs such as running a new scan or canceling a job	job-controller
Get information from the Controller such as the Fortify Software Security Center URL	info-controller
Check for client or sensor updates	update-controller
Check to see if the Controller is running	core-controller

To use the Fortify ScanCentral SAST API, your application makes an HTTP request and parses the response. The Fortify ScanCentral SAST API uses JSON and XML as its communication format and the standard HTTP methods of GET, POST, and DELETE. URIs have the following structure:

<protocol>://<controller_url>/rest/*<api-version>*/*<endpoint>*

The following is an example cURL:

```
curl -X 'GET' \
  'https://my_ctrl_host:8080/scancentral-ctrl/rest/v4/job/a2f0fe34-f810-
4c76-8e0b-86dfb4f40c9c/status' \
  -H 'accept: */*' \
  -H 'fortify-client: my_secret'
```

Authentication

Authenticate your API request with a Fortify ScanCentral SAST authentication token. Use the value of the `client_auth_token` or the `worker_auth_token` from the `config.properties` file for the Controller depending on the request. Set the same authentication token in the `fortify-client` header that is set for the `client_auth_token`. Similarly, set the same authentication token in the `fortify-worker` header that is set for `worker_auth_token`. The following table lists which authentication token is used for each request group.

Request group	Authentication token	
	client_auth_token	**worker_auth_token**
sensor-controller		x
job-controller	x	
info-controller	x	x
update-controller	x	x
core-controller	x	

Accessing the Fortify ScanCentral SAST API documentation (Swagger UI)

The documentation describes the input, output, and API endpoints. It also provides the ability to test the endpoints before using them in production.

To access this documentation:

1. Configure the credentials for access to the documentation in the Controller `config.properties` file with the two properties: `swagger_username` and `swagger_password`. For more information, see "Configuring the Controller" on page 29.
2. Open a browser and visit *<controller_url>*/rest/swagger-ui/index.html.

Note: OpenAPI documentation in JSON format is available at *<controller_url>*/rest/api-docs.

Chapter 3: About Fortify ScanCentral SAST sensors

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

For MBS scans, ScanCentral SAST supports all languages that Fortify Static Code Analyzer supports. For remote translation and scans of the project packages, ScanCentral SAST supports only the languages that can be used with remote translation. For a list of languages supported for performing remote translation, see the *Fortify Software System Requirements* document.

Tip: As you set up your Fortify 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.

This section contains the following topics:

Installing sensors

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

Installing a sensor using Fortify Static Code Analyzer

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

If you use Windows, you can install the sensor as a Windows service. For instructions, see "Installing a sensor as a service" on the next page.

To install a sensor:

1. Use the instructions provided in the *OpenText™ Fortify Static Code Analyzer User Guide* to install Fortify Static Code Analyzer.

   Make sure you select Fortify ScanCentral SAST client as a component during the Fortify Static Code Analyzer installation.
2. Open the `<sca_install_dir>/Core/config/worker.properties` file in a text editor.
3. Specify a value for the `worker_auth_token` property.

   If you are using a plain text password, use the password set for the `worker_auth_token` property in the Controller `config.properties` file. For information about how to generate an encrypted shared secret, see "Encrypting the shared secret on a sensor" on page 48.
4. Save and close your `worker.properties` file.

See also

"Configuring the Controller" on page 29

"Fortify Static Code Analyzer and ScanCentral SAST version compatibility" on page 58

Installing a sensor as a service

If you use Windows services, you can install the Fortify ScanCentral SAST 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 plain text password, run the following command:

     ```
     setupworkerservice.bat <sca_version> <controller_url> <shared_secret>
     ```
   - To use an encrypted password, run the following command:

     ```
     setupworkerservice.bat <sca_version> <controller_url> "<encrypted_
     shared_secret>" <path_to_pwtool.keys_file>
     ```

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

     where *<sca_version>* is the *<year>.<quarter>* portion of the Fortify Static Code Analyzer version (for example, 24.4).

     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 24.4 https://url.com this^^^^^^^^that
```

For a `worker_auth_token` that contains two caret characters, such as this^^that, run the following command:

```
setupworkerservice.bat 24.4 https://url.com this^^^^^^^^^^^^^^^^that
```

For information about how to encrypt a shared secret, see "Encrypting the shared secret on a sensor" on the next page.

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 53

See also

"Installing sensors" on page 45

Configuring sensors

After you install the Fortify ScanCentral SAST sensors, you can configure sensor settings such as the maximum run time for scans, sensor expiration time, job cleanup timing, and more.

See also

"Configuring proxies for clients and sensors" on page 62

"Avoiding read timeout errors" on page 40

Encrypting the shared secret on a sensor

Values exist in the ScanCentral SAST sensor configuration file as plain text. You can encrypt the `worker_auth_token` property value.

To encrypt a shared secret on a sensor:

1. At the command prompt, run the following command:

   ```
   <sca_install_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 `pwtool.keys` file to *<pwtool_keys_file>* and prints a new encrypted secret to the console.
3. Open the worker.properties file in a text editor and update the values for the following properties:
   a. Copy the encrypted secret and paste it as the value for `worker_auth_token` property.
   b. Add the name of your pwtool keys file:

      `pwtool_keys_file=<pwtool_keys_file>`
4. Save and close the `worker.properties` file.

See also

"Installing sensors" on page 45

Setting the maximum run time for scans

By default, a sensor can run a scan for an indefinite period of time, which prevents it from running other scans. You can limit the amount of time scans can run on sensors for a specific job, for a specific sensor, or globally for all sensors.

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 the maximum run time for a specific job

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

```
scancentral -url <controller_url> start --scan-timeout 1
```

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

```
scancentral -url <controller_url> worker --scan-timeout 2
```

Configuring the maximum run time for all sensors

To configure the maximum run time for all sensors:

1. Open the *<controller_install_dir>*/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties file in a text editor.
2. Set the scan_timeout property value to the maximum number of minutes for scans to run on sensors.
3. Save and close the config.properties file.

Changing sensor expiration time

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

1. Open the *<controller_install_dir>*/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties file in a text editor.
2. Set the worker_expiry_delay property value to the number of hours to elapse after inactivity before sensors expire.
3. Save and close the config.properties file.

Configuring sensors for remote translation of .NET languages

To use your Fortify ScanCentral SAST sensors for remote translation of code written in a .NET language, configure at least one sensor with the software required to support .NET. Sensors on Windows or Linux can accept any package for remote translation built by MSBuild and dotnet if .NET capability is enabled. See the *Fortify Software System Requirements* document for specific .NET version requirements.

After you start a ScanCentral SAST sensor, it automatically detects if a supported version of .NET is installed and displays a message that .NET capability is enabled. This indicates that the sensor can now translate .NET projects.

Important! To avoid Windows errors caused by too long a path during a .NET translation, start ScanCentral SAST sensors from a directory with a short name and path.

See also

"Installing sensors" on page 45

"Starting the sensors" on page 52

Configuring sensors to use the progress command when starting on Java

To use the `progress` command to check the progress of your Fortify Static Code Analyzer scans, you must complete the following sensor configuration:

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 user name.
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 a Windows system, `cacls jmxremote.password /P <username>:R`
   - On a Linux system, `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, Fortify ScanCentral SAST clients start on the specified port using JMX password authentication. Make sure that the port is not already bound.

Caution! 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 Fortify ScanCentral SAST instances, each with its own `worker.properties` file.

Configuring where to generate job files and the worker_ persist.properties file

For containerized deployments, it is useful to determine where files are generated so that you can customize persistence. This enables you to persist the `worker_persist.properties` file, which you

need to maintain sensor pool assignments, without having to keep all 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 job files is *<working_dir>*`/jobs`.

To configure where job files and the `worker_persist.properties` file are generated:

1. On a sensor machine, open the *<sca_install_dir>*`/Core/config/worker.properties` file in a text editor.
2. Specify directories for the following properties:
   - The `props_dir` property specifies where the `worker_persist.properties` file is saved.
   - The `jobs_dir` property specifies the directory where the job files are created.
3. Save and close your `worker.properties` file.
4. Restart the sensor.

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, FPR files, logs, and so on), and Fortify Static Code Analyzer build files related to cleaned Fortify ScanCentral SAST jobs. Although you cannot turn off this feature, you can configure its timing.

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

1. Open the *<sca_install_dir>*`/Core/config/worker.properties` file in a text editor.
2. Configure the following properties based on your scheduling needs.

Property	Description	Default value (hours)
`worker_cleanup_age`	Age (in hours) job files must be before they are removed from the sensor working directory	168 (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.

Configuring sensor properties

In addition to setting sensor properties in the *<sca_install_dir>*`/Core/config/worker.properties` file, you can add them to the SCANCENTRAL_VM_OPTS environment variable. A property value set in the SCANCENTRAL_VM_OPTS environment variable

overrides any equivalent property set in the `worker.properties` file. The following example sets the sensor authorization token and the connection timeout between the Controller and a sensor:

- On a Windows system: `set SCANCENTRAL_VM_OPTS=-Dworker_auth_token=<token> -Drestapi_connect_timeout=10000`
- On a Linux system: `export SCANCENTRAL_VM_OPTS=-Dworker_auth_token=<token> -Drestapi_connect_timeout=10000`

Note: You can also set Java system properties (such as `-Djava.io.tmpdir=<path>`) in the SCANCENTRAL_VM_OPTS environment variable.

Starting the sensors

To start the Fortify ScanCentral SAST sensors:

1. Start the Controller if it is not already running.
2. On each sensor, navigate to *<sca_install_dir>*`/bin`.
3. Start the sensor by typing the following command:

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

If the sensor starts successfully, it displays messages to signal its waiting status on 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" below.

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

See also

"Placing the Controller in maintenance mode" on page 42

"Configuring sensor auto-start" below

Configuring sensor auto-start

The following topics provide general guidance to enable sensor auto-start and might not be appropriate in all environments. OpenText 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

Make sure the ScanCentral SAST 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 user with administrator permissions.

   Sensors are dedicated machines intended only to run Fortify Static Code Analyzer on behalf of Fortify ScanCentral SAST. Do not share them with any other service. To avoid issues associated with insufficient permissions, use a fully privileged administrator 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 `setupworkerservice.bat` with no options to display the usage help.
4. Re-run the batch script with the required options 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. OpenText 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. Make sure that the sensor communicates with the Controller.

See also

"Installing a sensor as a service" on page 46

"Troubleshooting a sensor as a Windows service" on page 94

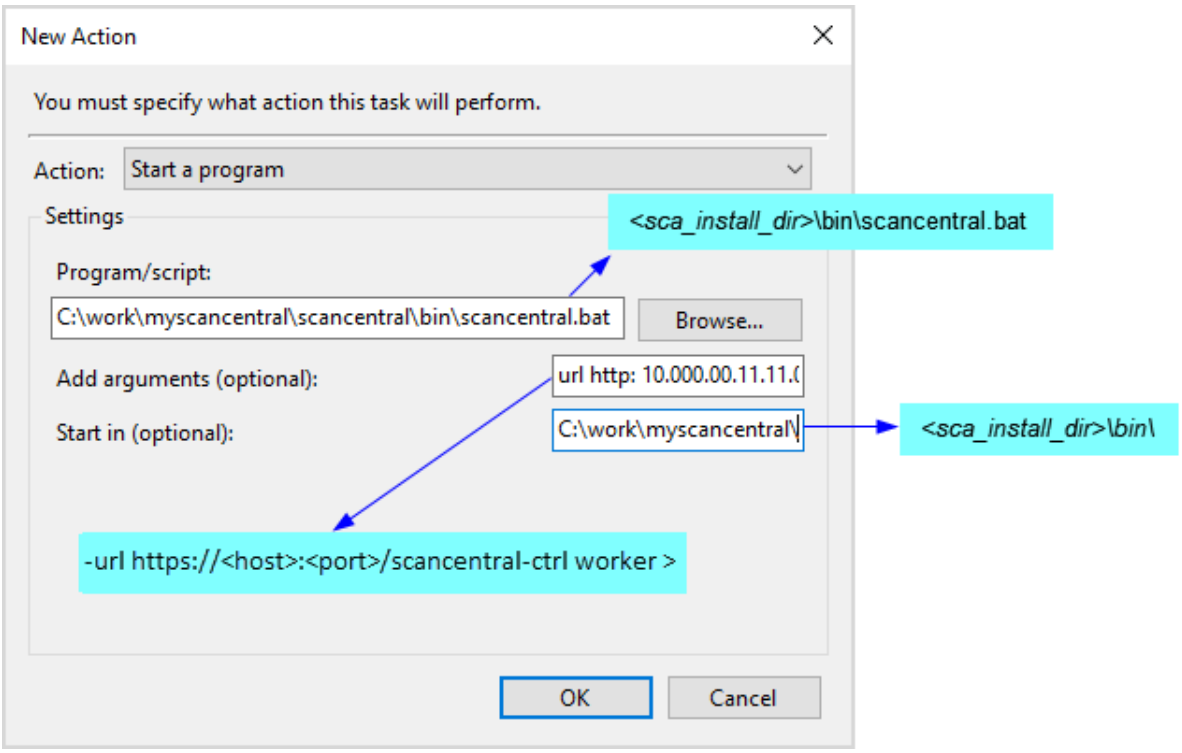
Enabling sensor auto-start on Windows as a scheduled task

To enable Fortify ScanCentral SAST sensor auto-start on Windows as a scheduled task:

1. Log on to the sensor machine as a local user with administrator permissions.

   Sensors are dedicated machines intended only to run Fortify Static Code Analyzer on behalf of Fortify ScanCentral SAST. Do not share them with any other service. To avoid issues associated with insufficient permissions, use a fully privileged administrator account for the auto-start setup.
2. Start the Task Scheduler.
3. In the **Actions** pane, select **Create Task**.
4. On the **General** tab, provide the following information:
   a. In the **Name** box, type a name for the task.
   b. Click **Run whether user is logged on or not**.
5. Click the **Actions** tab, and then click **New**.

   The New Action dialog box opens.



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

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

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

6. Click the **Triggers** tab.
7. Make sure that the **At startup** trigger is enabled, and then click **OK**.
8. Click the **Settings** tab.
9. Make sure the **Stop the task if it runs longer than** check box is cleared, and then click **OK**.
10. 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 when you are logged into the machine.

Enabling sensor auto-start on a Linux system

The following procedure has been tested with Red Hat Enterprise Linux; there might be some variation for other Linux varieties. Review these steps with your system administrator before you

make any changes.

To enable Fortify ScanCentral SAST sensor auto-start on a Linux system:

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 it based on 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 start 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
      ```

For example:

```
tail -f /opt/Fortify/Fortify_SCA_24.4.0/bin/workerout.txt
```

Safely shutting down sensors

This topic describes how to safely shutdown Fortify ScanCentral SAST sensors from Fortify Software Security Center.

Important! If the Controller is in maintenance mode (see "Placing the Controller in maintenance mode" on page 42), you cannot shut down sensors from Fortify Software Security Center.

To shut down active sensors:

1. Sign in to Fortify Software Security Center as an administrator and select the ScanCentral view.
2. In the left pane of the SAST page, 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**.

   If the **SHUT DOWN** button is not enabled, it can mean that:
   - The Controller is in maintenance mode.
   - The sensor is already shut down.
   - 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 complete, the state then changes to **Inactive**.

Chapter 4: About Fortify ScanCentral SAST clients

A Fortify ScanCentral SAST client can generate packages with sources and dependencies, which are uploaded to the Controller for remote translation and scan on sensors. You can use this functionality independent of Fortify Static Code Analyzer.

A Fortify ScanCentral SAST client can also run on a build machine where Fortify Static Code Analyzer translates code and generates 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 options, are uploaded to the Controller for analysis by sensors.

This section contains the following topics:

Embedded clients and standalone clients

A client can be either an *embedded* client, which is part of the Fortify Static Code Analyzer distribution or a *standalone* client, which is independent of Fortify Static Code Analyzer. The interface for issuing Fortify ScanCentral SAST commands is installed on your client. You use this interface to set the options for the scan and communicate your intentions to the Controller.

Within a Fortify Static Code Analyzer installation, the files used to create Fortify ScanCentral SAST sensors and embedded clients are the same. The only difference is how you invoke the functionality from the command line. To use Fortify ScanCentral SAST as a sensor, you run Fortify ScanCentral SAST using the `worker` command. To use Fortify ScanCentral SAST as an embedded client to start a scan, 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. You can use an embedded client for either local translation and remote scan or remote translation and scan.

A standalone client does not require the installation of Fortify Static Code Analyzer. You can use it to create a package of the code with its dependencies to send to the Controller for remote translation and scan.

Fortify Static Code Analyzer and ScanCentral SAST version compatibility

The Fortify Static Code Analyzer version on a Fortify ScanCentral SAST client must be compatible with the Fortify Static Code Analyzer version installed on the sensors. The version number format is `year.quarter.patch.buildnumber` (for example 24.4.0.0068). By default, the year and quarter portions of the Fortify Static Code Analyzer version numbers on both the client and sensor must match. For example, version 24.4.0 works with version 24.4.1. For other options of supported version compatibility, see the ScanCentral SAST Controller configuration property `sensor_ version_for_ all_jobs` in "Configuring the Controller" on page 29.

To determine the Fortify Static Code Analyzer version, run the command `sourceanalyzer -version`.

Installing 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 machine you plan to use as Fortify ScanCentral SAST clients. If you use a language supported for remote translation, you can install standalone clients, independent of Fortify Static Code Analyzer. For a list of languages that ScanCentral SAST supports for remote translation, see the *Fortify Software System Requirements* document.

In this guide, *<client_install_dir>* refers to the ScanCentral SAST client installation directory.

See also

"Fortify Static Code Analyzer and ScanCentral SAST version compatibility" above

"Installing an embedded client" below

"Installing a standalone client" on the next page

Installing an embedded client

Use an embedded client (client included with Fortify Static Code Analyzer) to perform a local translation before submitting the remote scan 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 *OpenText™ Fortify Static Code Analyzer User Guide* to install Fortify Static Code Analyzer on your build machine.

Make sure you select Fortify ScanCentral SAST client as a component during the Fortify Static Code Analyzer installation.

3. Navigate to the *<sca_install_dir>*/Core/config directory, and then open the client.properties in a text editor.
4. Set the same value for the client_auth_token property that you set for the client_auth_token property on the Controller (in the *<controller_install_dir>*/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties file).
   For information about how to generate an encrypted shared secret, see "Encrypting the shared secret on a client" on page 61.
5. Save and close the client.properties file.

See also

"Installing a standalone client" below

Installing a standalone client

To submit scan requests for remote translation and remote scan to your Fortify ScanCentral SAST sensors, you can use standalone clients. A standalone client is independent of a Fortify Static Code Analyzer installation.

To install a standalone client:

1. Copy the Fortify ScanCentral SAST client files to your machine by doing one of the following:
   - Install from a ScanCentral SAST client ZIP file:
     i. Extract the contents of the Fortify_ScanCentral_Client_*<version>*_x64.zip file to any directory on your machine.

     Important! Make sure that the installation path contains no spaces.

     ii. On Linux systems, give the scancentral, pwtool, and packagescanner files execute permission.
     iii. Add *<client_install_dir>*/bin to your PATH environment variable.
     The *<client_install_dir>* is the directory where you extracted the ScanCentral SAST client ZIP.
     iv. Set the SCANCENTAL_JAVA_HOME environment variable to point to a Java version that Fortify ScanCentral SAST client supports, and make sure that you add the Java executable to the PATH environment variable.
   - Install the ScanCentral SAST client as a component of a Fortify Applications and Tools installation.
     For instructions, see the *OpenText™ Fortify Static Code Analyzer Applications and Tools Guide*.
2. Open the *<client_install_dir>*/Core/config/client.properties file in a text editor.

3. Set the same value for the `client_auth_token` property that you set for the `client_auth_token` property on the Controller (in the *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file). For information about how to generate an encrypted shared secret, see "Encrypting the shared secret on a client" on the next page.
4. Save and close the `client.properties` file.

See also

"Placing multiple standalone clients on the controller" below

"Upgrading a client" on page 67

Placing multiple standalone clients on the controller

You can place multiple standalone clients of different supported versions on the Controller. To do this, place any number of client ZIP files for all supported versions into the *<controller_install_dir>*`/tomcat/client` directory. You can use any ZIP file names. At startup, the Controller parses the available clients.

To install a patch for a client or sensor version installed on the Controller, place the patch ZIP file into the *<controller_install_dir>*`/tomcat/client` directory. If automatic updates is enabled, the clients of that version are automatically updated with the patch. For information about how to enable automatic updates of your clients and sensors, see "Enabling automatic updates of clients and sensors" on page 68.

Configuring clients

After you install the Fortify ScanCentral SAST client, you can encrypt shared secrets and configure proxy settings.

To configure the ScanCentral SAST client:

1. On the client machine, open the *<client_install_dir>*`/Core/config/client.properties` file in a text editor.
2. Configure the properties described in the following table.

Property	Description
`client_auth_token`	Specifies 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 a client" on the next page.

Property	Description
pwtool_keys_file	Specifies the path to a file with pwtool keys. If encrypted passwords are used, this must specify a file with the pwtool keys used to encrypt the passwords. For more information, see "Encrypting the shared secret on a client" below.
restapi_connect_timeout	Specifies the Controller connection timeout (in milliseconds). You can use this, and the restapi_read_timeout property to resolve timeout errors between the Controller and the client.
restapi_read_timeout	Specifies the Controller read timeout (in milliseconds). You can use this, and the restapi_connect_timeout property to resolve timeout errors between the Controller and the client.
use_system_gradle	If set to true, ScanCentral SAST uses the Gradle included in the PATH environment variable. By default, ScanCentral SAST uses the Gradle wrapper included in the project being analyzed.
debricked_cli_dir	(Fortify on Demand users only) Specifies a custom location for the Debricked CLI installation.

For a description of the proxy-related properties for clients, see "Configuring proxies for clients and sensors" on the next page.

3. Save and close the client.properties file.

See also

"Avoiding read timeout errors" on page 40

Encrypting the shared secret on a client

Passwords exist in the ScanCentral SAST client configuration file as plain text. You can encrypt the client_auth_token property value.

To encrypt a shared secret on a client:

1. At the command prompt, run one of the following commands:
   - For an embedded client installed with 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. Open the `client.properties` file in a text editor and update the values for the following properties:

   a. Copy the new encrypted secret, and paste it as the value for the `client_auth_token` property.

   b. Add the name of your pwtool keys file:

   `pwtool_keys_file=<pwtool_keys_file>`

4. Save and close the `client.properties` file.

See also

"Installing clients" on page 58

Configuring proxies for clients and sensors

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

To configure proxies for clients and sensors:

1. Navigate to the *<client_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 proxy 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 the proxy user name.
ssc_proxy_password	If authentication is required, type the password for the proxy user.

2. Save and close the `client.properties` and `worker.properties` files.

3. To enable proxy authentication when the Controller is running under HTTPS, add the `-Djdk.http.auth.tunneling.disabledSchemes` Java property to the SCANCENTRAL_VM_OPTS environment variable by typing one of the following commands:

   On a Windows system: `set SCANCENTRAL_VM_OPTS=-Djdk.http.auth.tunneling.disabledSchemes=`

   On a Linux system: `export SCANCENTRAL_VM_OPTS=-Djdk.http.auth.tunneling.disabledSchemes=`

Configuring client properties

In addition to setting client properties in the *<client_install_dir>*`/Core/config/client.properties` file, you can add them to the SCANCENTRAL_VM_OPTS environment variable. A property value set in the SCANCENTRAL_VM_OPTS environment variable overrides any equivalent property set in the `client.properties` file. The following example sets the client authorization token and the connection timeout between the Controller and a sensor:

- On a Windows system: `set SCANCENTRAL_VM_OPTS=-Dclient_auth_token=`*<token>* `-Drestapi_connect_timeout=10000`
- On a Linux system: `export SCANCENTRAL_VM_OPTS=-Dclient_auth_token=`*<token>* `-Drestapi_connect_timeout=10000`

Note: You can also set Java system properties (such as `-Djava.io.tmpdir=`*<path>*) in the SCANCENTRAL_VM_OPTS environment variable.

Chapter 5: Upgrading Fortify ScanCentral SAST components

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

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

This section contains the following topics:

Supporting multiple Fortify Static Code Analyzer versions

To support heterogeneous environments and facilitate phased Fortify Static Code Analyzer upgrades, the Fortify ScanCentral SAST 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. By default, jobs from each client are then routed to the sensor that has the same Fortify Static Code Analyzer version installed. You can change this behavior and specify a specific sensor version for all jobs (see "Configuring the Controller" on page 29).

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

Important! If you clone VMs, then after cloning, you *must* remove the `worker_persist.properties` file from the directory specified for the `props_dir` property (see "Configuring where to generate job files and the worker_persist.properties file" on page 50).

Use sensor machines dedicated to Fortify ScanCentral SAST and run sensors under a dedicated user name. Run only one sensor instance per machine.

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

Make sure a security system or other tool does not block the following channels of communication:

- Controller to Fortify Software Security Center port (for uploads of scan results)
- Fortify Software Security Center to the ScanCentral SAST Controller port (for Fortify ScanCentral SAST administration console functionality)
- Clients to the Controller port
- Sensors to the Controller port
- Clients to the Fortify Software Security Center port (required only if Fortify Software Security Center is in lockdown mode, or if you use the `-sscurl` option)

Upgrading the Controller

To upgrade your Fortify ScanCentral SAST Controller:

1. (Recommended) Allow all jobs to finish.

   Place the Controller in maintenance mode so that sensors complete all currently running scans.
2. Shut down the Controller.
3. Back up the existing Controller directories.
4. Install the new Controller in a different location from the existing Controller directories.

   If you plan to install the Controller as a Windows or Linux service, make sure that you install the Controller in a directory where the local service (Windows) or the user or group using the service (Linux) has access.
5. If your existing `config.properties` file has been modified, you must manually apply any changes you made to the new `config.properties` file.

   You cannot simply copy the existing `config.properties` file.
6. If (and only if) you are upgrading your Controller from version 23.1.x to version 24.4.0, run the migration script as follows:
   a. Open a command prompt and navigate to the new 24.4.0 Controller installation directory.
   b. At the command prompt, enter `cd db-migrate`.
   c. Identify the `cloudCtrlDb` and Controller directories for the existing Fortify ScanCentral SAST version. In the following example, the existing Controller is installed on a Windows system in the `C:\scancentral23.1.0` directory:

   ```
   C:\scancentral23.1.0\tomcat\cloudCtrlDb
   C:\scancentral23.1.0\tomcat\webapps\scancentral-ctrl
   ```

d. Run the following command.

This command example includes the example directories shown in the preceding step.

```
migrate C:\scancentral23.1.0\tomcat\cloudCtrlDb
C:\scancentral23.1.0\tomcat\webapps\scancentral-ctrl
```

The migration script generates the `cloudCtrlDb` directory in the current working directory.

7. Navigate to the `jobFiles` and `cloudCtrlDb` directories of the existing Controller, and then copy them to the corresponding directories for the new Controller.

Important! If you migrated the database (step 6), make sure that you copy the migrated database (`cloudCtrlDb` directory) to the new Controller installation directory.

The process owner must have write permission for the database file in the `cloudCtrlDb` directory. If you run the ScanCentral SAST Controller as a Windows service, make sure that the LOCAL SERVICE account has write permission to the database file.

To change these directories, edit the `job_file_dir` and `db_dir` properties in the `config.properties` file (see "Configuring the Controller" on page 29).

8. Start the new Controller.

The database is automatically migrated.

9. (Optional) Remove the Controller directories for the previous version.

See also

"Installing the Controller" on page 21

"Upgrading Fortify ScanCentral SAST components" on page 64

"Upgrading sensors" below

"Enabling automatic updates of clients and sensors" on page 68

Upgrading sensors

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

To upgrade your Fortify 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 plan only to use remote translation and analysis workflows. Local translation requires a local Fortify Static Code Analyzer installation. You can also find the Fortify 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 sensors and clients. For details, see "Enabling automatic 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. Install or upgrade Fortify Static Code Analyzer using the instructions provided in the *OpenText™ Fortify Static Code Analyzer User Guide*.
3. Check the `<sca_install_dir>/Core/config` directory to make sure that the `worker.properties` file resides there.
4. Add the following property to the `worker.properties` file:

   ```
   worker_auth_token=<value_set_in_controller_configuration>
   ```

5. Specify either a plain 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 48.
6. Save the `worker.properties` file.
7. Start the sensors.

See also

"Enabling automatic updates of clients and sensors" on the next page

"Starting the sensors" on page 52

"Configuring sensors to use the progress command when starting on Java" on page 50

"Upgrading the Controller" on page 65

Upgrading a client

Important! OpenText recommends that your standalone Fortify 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), do one of the following:

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

To upgrade an embedded client, which 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 *OpenText™ Fortify Static Code Analyzer User Guide*.
4. Accept all overwrite requests.

   On a Linux system, you might 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

"Installing a standalone client" on page 59

"Installing an embedded client" on page 58

Enabling automatic updates of clients and sensors

You can have all Fortify ScanCentral SAST clients and sensors check with the Controller after a manual update and following each startup to determine whether updates are available (meaning 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:

- You can update standalone clients to a major or a patch version (for example from 24.2.0 to 24.4.0, or from 24.4.0 to 24.4.1).
- If automatic updates are enabled and a major update of standalone clients fails, the clients do not start any jobs until they are updated.
- If automatic updates are enabled and a patch update of standalone clients fails, the clients continue to work and a warning is displayed.
- You can only update embedded clients and sensors to a patch version (for example, from 24.2.0 to 24.2.1 or 24.2.2, but not to 24.4.0). Automatic updates for major versions is not available for embedded clients and sensors.
- If automatic updates are enabled and a patch update of an embedded client fails, the clients and sensors continue to work and a warning is displayed.

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

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

To enable automatic updates of your clients and sensors:

1. Open the *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file in a text editor.
2. To enable automatic updates, set `client_auto_update` to `true`.
3. Save and close the file.

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

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

See also

"Upgrading Fortify ScanCentral SAST components" on page 64

"Upgrading the Controller" on page 65

Chapter 6: Submitting scan requests

You can request a scan that performs remote translation and scan or one that performs a remote scan for a project that is already translated to your Fortify ScanCentral SAST sensors. This chapter describes how to submit scan requests (including special considerations for some project languages), how to upload your scan results to Fortify Software Security Center in your scan request, and how to prepare a Fortify ScanCentral SAST package to be scanned without sending it to a Controller.

This section contains the following topics:

Submitting local translation and remote scan requests

You can submit a project that Fortify Static Code Analyzer has already translated to your Fortify ScanCentral SAST sensors for remote scanning. To submit a scan request to perform only the scan phase, use the `start` command with either the `--build_id` (`-b`) or the `-mbs` option to identify the local translation or an existing mobile build session file together with the `-scan` option. The following is an example of a scan request to submit a remote scan:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -b <build_id> -scan
```

You can include supported Fortify Static Code Analyzer scan options in the scan request (see "Options accepted for -sargs (--scan-args)" on page 108). You must include only the Fortify Static Code

Analyzer analysis options after the `-scan` option. If the parameter for the option you specify includes a space, you must enclose it in quotes. For example:

```
-scan -build-label "Application 5.4 June 8, 2025"
```

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

By default, jobs submitted and scan results (FPR files) cannot be larger than 1 GB. Before you start large scans, review "Optimizing scan performance" on page 83.

See also

"Submitting remote translation and scan requests" below

"Global options" on page 98

"Start command" on page 99

Submitting remote translation and scan requests

If you use a supported language, you can submit your project to your Fortify ScanCentral SAST sensors for a complete remote analysis (both translation and scan phases). To submit a scan request that performs both the translation and scan phases, use the `start` command.

Fortify ScanCentral SAST automatically detects the build tool you are using 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` (Windows) or to `dotnet` (Linux) and sets `-bf` to the `xxx.sln` file. If Fortify ScanCentral SAST 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 was selected based on the multiple file types found.

The following table provides example scan request commands for different tasks. The examples assume that the command is run from the project's working directory. The build tool option `--build-tool` (`-bt`) shown in these example commands is not required.

Task	Example command
Start a job to scan a .NET application	`scancentral -sscurl <ssc_url> -ssctoken <token> start`
Start a job to scan a dotnet project on Windows	`scancentral -sscurl <ssc_url> -ssctoken <token> start -bt dotnet -bf mySolution.sln`

Task	Example command

Start a job to scan a Maven project that includes the test scope	`scancentral -sscurl <ssc_url> -ssctoken <token> start -bt mvn --include-test`
	or
	`scancentral -sscurl <ssc_url> -ssctoken <token> start -t`

Start a job to scan a Maven project with a non-default build file	`scancentral -sscurl <ssc_url> -ssctoken <token> start -bt mvn -bf c:\myproj\myproj-pom.xml`
Start a job to scan a JavaScript/TypeScript project	`scancentral -sscurl <ssc_url> -ssctoken <token> start`
Start a job to scan a PHP version 8.2 project	`scancentral -sscurl <ssc_url> -ssctoken <token> start -hv 8.2`
Start a job to scan an ABAP project	`scancentral -sscurl <ssc_url> -ssctoken <token> start`
Start a job to scan a Go project with a build tag	`scancentral -sscurl <ssc_url> -ssctoken <token> start -targs "-gotags release"`
Start a job to scan a Ruby project	`scancentral -sscurl <ssc_url> -ssctoken <token> start`
Start a job to scan a Gradle project	`scancentral -sscurl <ssc_url> -ssctoken <token> 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 -sscurl <ssc_url> -ssctoken <token> start -email username@domain.com -upload -application "MyProject" -version "1.0"`

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

See also

"Submitting local translation and remote scan requests" on page 70

"Global options" on page 98

"Start command" on page 99

"Uploading results to Fortify Software Security Center" 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:

- The UUID for the sensor pool
- The `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 and open the Fortify ScanCentral SAST page.
2. In the left pane of the SAST page, select **Sensor Pools**.
3. In the **Sensor Pools** table, copy the value shown in the **UUID** column for the sensor pool you want to target for a scan request.

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

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

- At the command prompt on the client host, run the following command:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -pool <uuid>
```

Scanning Java projects

Set the `SCANCENTRAL_JAVA_HOME` environment variable to a version of Java that Fortify ScanCentral SAST supports. For the supported Java versions, see the *Fortify Software System Requirements* document.

Note: This is only required if the JAVA_HOME environment variable is set to Java version that Fortify ScanCentral SAST does not support.

Scanning .NET projects

Fortify ScanCentral SAST MSBuild integration is available on Windows only. Fortify ScanCentral SAST dotnet integration is available on Windows and Linux.

To translate and scan .NET projects, the client machine must have the software required to build and package .NET projects installed:

- MSBuild or dotnet (see supported versions of MSBuild in the *Fortify Software System Requirements* document)

- NuGet (optional)
- .NET Framework, .NET Core, or .NET Standard as required for the project configuration

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

Some projects also require that you start NuGet to restore some dependencies. If any dependencies are unresolved, the build fails and the scan results might be incomplete. For these types of projects, you must install NuGet manually on the machine and make sure it is included in the PATH environment variable. If NuGet is found, Fortify ScanCentral SAST runs it automatically.

The following are command-line examples to translate and scan a .NET project:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start --build-tool msbuild
--build-file <sln_file_or_path_to_sln_file>
```

```
scancentral -sscurl <ssc_url> -ssctoken <token> start --build-tool dotnet
```

The following command uses MSBuild integration on a Windows client and dotnet integration on a Linux client because no build tool option is specified:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start --build-file <sln_
file_or_path_to_sln_file>
```

Note: To use the dotnet integration on a Windows client, you must include `-bt dotnet`.

If no build tool is specified, ScanCentral SAST client tries to automatically detect the build tool for `*.sln`, `*.csproj`, `*.vbproj`, and `dirs.proj`.

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

Excluding .NET Projects from analysis

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

For example, the solution `MySolution.sln` includes two projects: ProjectA and ProjectB. The *<build_config>* file, created in Visual Studio excludes ProjectB from the builds. To exclude ProjectB from Fortify ScanCentral SAST analysis, run the following from the directory where the solution file resides:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start --build-tool msbuild
--build-file MySolution.sln --build-command "/t:Rebuild
```

```
/p:Configuration=<build_config>"
```

See also

"Configuring sensors for remote translation of .NET languages" on page 49

Scanning JavaScript and TypeScript code

By default, any NPM dependencies (node_modules directory) that exists in your project is included in the project package only for translation. This improves the analysis results by including type resolution information from the JavaScript and TypeScript code. However, Fortify Static Code Analyzer excludes the files in node_modules from the analysis and no vulnerabilities are reported for these NPM dependencies in the scan results.

To exclude the node_modules directory from your project package, use the `-exclude` option in the `start` or `package` command. For example:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -exclude "node_
modules"
```

To include the NPM dependencies in the scan results, see the *OpenText™ Fortify Static Code Analyzer User Guide* for information about translating JavaScript and TypeScript code.

Scanning Python projects

Fortify ScanCentral SAST clients can work with Python projects in three ways:

- Submit a scan request in a prepared virtual environment (see "Submitting a scan request in a virtual environment" on the next page).
- Use an existing virtual environment, without activating that virtual environment (see "Submitting a scan request in an unactivated virtual environment" on the next page). In this case, Fortify ScanCentral SAST activates the virtual environment.
- Start the job outside of a virtual environment (see "Submitting a scan request outside of a virtual environment" on page 77).

The following table provides examples of different ways to submit scan requests for Python code.

Task	Example command
Start a job to scan a Python 3 project	`scancentral -sscurl <ssc_url> -ssctoken <token> start --python-requirements <requirements_file_path>`
Start a job to scan a Python 2 project	`scancentral -sscurl <ssc_url> -ssctoken`

Task	Example command
	`<token> start –-python-version 2 --python-requirements <requirements_file_path>`
Start a job to scan a Python project under an active virtual environment with dependencies already installed	`scancentral -sscurl <ssc_url> -ssctoken <token> start`
Start a job to scan a Python project under an active virtual environment without project dependencies installed	`scancentral -sscurl <ssc_url> -ssctoken <token> start --python-requirements <requirements_file_path>`
Start a job to scan a Python project using an existing Python virtual environment and install project dependencies	`scancentral -sscurl <ssc_url> -ssctoken <token> start --python-virtual-env <venv_location>`

Submitting a scan request in a virtual environment

If you work in a virtual environment, all your project dependencies are already installed. You do not need to invoke the pip package manager before you start the job. Fortify ScanCentral SAST can detect the Python version automatically.

To start the scan job in a virtual environment:

1. At the command prompt, activate the virtual environment.
2. Start a job to scan the Python project as shown in the following example:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start
```

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

```
scancentral -sscurl <ssc_url> -ssctoken <token> start --python-requirements <requirements_file_path>
```

Submitting a scan request in an unactivated virtual environment

To start the scan job in a virtual environment (with all dependencies installed) without activating that virtual environment:

- At the command prompt, start the Python project scan as shown in the following examples:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start --python-virtual-
env <venv_location>
```

or

```
scancentral -sscurl <ssc_url> -ssctoken <token> start --python-virtual-
env <venv_location> --python-requirements <requirements_file_path>
```

Fortify ScanCentral SAST goes to the virtual environment, determines the Python version used, packages all required libraries, and then submits the scan job to the Controller.

Submitting a scan request outside of a virtual environment

To start the scan job if there is no virtual environment on the client, you must have Python installed on the client. If the Python version is not specified in the command, then Fortify ScanCentral SAST uses first working version from PATH environment variable. Fortify ScanCentral SAST locates the Python installation. In this case, Fortify ScanCentral SAST creates a temporary virtual environment, installs all dependencies from the requirements file, and then submits the job to the Controller.

To start the scan job outside of a virtual environment:

- At the command prompt, start the scan job as shown in the following example:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start --python-version 3
```

Scanning Go projects

To enable Fortify ScanCentral SAST clients to package Go projects for remote translation and scan, the following requirements must be met:

- The Go compiler must be installed on the client to resolve project dependencies.
- The Go compiler executable location must be available in the PATH variable.
- Configure the Go environment variables. 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 directory under the project root has all the required dependencies. The sensor rewrites the `GOFLAGS` system variable with `GOFLAGS=-mod=vendor` when it runs a Fortify Static Code Analyzer

translation.

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

Scanning PHP projects

If your PHP project uses the Composer dependency manager and you want to include dependencies in the analysis, then do the following on the client machine:

- Install PHP and Composer
- Configure the `php.ini` to run Composer for your project

This enables Fortify ScanCentral SAST client to invoke Composer to restore the dependencies before packaging the project for analysis. If the above requirements are not met, then Fortify ScanCentral SAST packages the project without restoring the dependencies.

Scanning COBOL projects

Fortify ScanCentral SAST clients can package COBOL projects for remote translation and scan. For detailed information about the requirements and options available for COBOL analysis, see the *OpenText™ Fortify Static Code Analyzer User Guide*.

You must have a sensor with the Windows operating system. Fortify ScanCentral SAST automatically assigns COBOL scans to a Windows sensor. If no Windows sensor is available, then the scan job is created but cannot be started.

Make sure the copybook files are in a separate directory from the COBOL source code files. OpenText recommends that you place your COBOL source code files in a directory called sources and your copybook files in a directory called copybooks. Create these directories at the same level.

Note: To analyze a COBOL project on Linux and to use Legacy COBOL translation, you must perform a local Fortify Static Code Analyzer translation:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -b <build_id>
```

The following example command submits a scan request for a COBOL project where the copybooks files are in the local `copybooks` directory:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -targs "-copydirs
copybooks -dialect COBOL390"
```

The following example command submits a scan request for a COBOL project that contains source code files with a non-standard file extension `mfcbl`:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -targs "-copydirs
MyCopydir1;MyCopydir2 -Dcom.fortify.sca.fileextensions.mfcbl=COBOL"
```

The following example command submits a scan request for a COBOL project that contains source code files without file extensions:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -targs "-copydirs
MyCopyDir -noextension-type COBOL"
```

Scanning SQL projects

On Windows (and Linux for .NET projects only), Fortify Static Code Analyzer assumes that files with the `.sql` extension are T-SQL rather than PL/SQL. To perform remote translation of a SQL project, you might need to specify what type of SQL your project uses.

To scan the project, run one of the following commands:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -targs "-sql-language
PL/SQL"
```

or

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -targs "-sql-language
TSQL"
```

Uploading results to Fortify Software Security Center

To submit a scan request and upload the scan results to an application version in Fortify Software Security Center, you must have an authentication token of type ScanCentralCtrlToken. You can create an authentication token with the fortifyclient utility or in Fortify Software Security Center. You can reuse the token for future requests. The fortifyclient utility is provided with Fortify Software Security Center and the Fortify Applications and Tools installation. For more information about creating authentication tokens with the fortifyclient utility or in Fortify Software Security Center, see the *OpenText™ Fortify Software Security Center User Guide*.

There are two options for providing upload permission, which depend on the permissions you want to give to your Fortify Software Security Center users:

- The user assigned a role that has **Run ScanCentral SAST scans**, **View ScanCentral SAST**, **View application versions**, and **Upload analysis results** permissions generates the token.
- The user assigned a role that has the **Run ScanCentral SAST scans** and **View ScanCentral SAST** permissions (and does not have the **Upload analysis results** permission) generates the token and

the Controller is configured with a ScanCentral SAST Controller service account.

Use this option to upload the scan results to Fortify Software Security Center using the Controller service account.

To configure a ScanCentral SAST Controller service account:

a. In Fortify Software Security Center, create a ScanCentral SAST Controller service account that has the **ScanCentral SAST Controller** role.

   For instructions on how to create Fortify Software Security Center user accounts, see the *OpenText™ Fortify Software Security Center User Guide*.

b. Open the `<controller_install_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file in a text editor.

c. Specify the credentials for the ScanCentral SAST Controller service account in the `ssc_ctrl_account_username` and `ssc_ctrl_account_password` properties.

d. Save and close the `config.properties` file.

e. To apply the change, restart the Controller.

Note: The **Run ScanCentral SAST scans** permission and the **ScanCentral SAST Controller** role are available in Fortify Software Security Center version 24.4.0 and later. To use an earlier version of Fortify Software Security Center, you must do one of the following:

- Ensure that the account of the user that generates the token has a role that includes the **Upload analysis results** and **View ScanCentral SAST** permissions.
- Configure the Controller (steps b-e in the previous procedure) with a ScanCentral SAST Controller service account created in Fortify Software Security Center that has a role that includes the **View ScanCentral SAST**, **View application versions**, and **Upload analysis results** permissions.

Examples of scan requests that upload scan results

The following example scan requests perform a remote translation and scan and upload the scan results:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -upload -versionid
<app_version_id>
```

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -upload -application
<app_name> -version <app_version>
```

The following example scan request performs a local translation and remote scan and uploads the scan results:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -upload -versionid
<app_version_id> -b <build_id> -scan
```

See also

"Retrying failed uploads to Fortify Software Security Center" on the next page

"Global options" on page 98

"Start command" on page 99

"Submitting remote translation and scan requests" on page 71

"Submitting local translation and remote scan requests" on page 70

Specifying a scan results (FPR) file name

You can specify the name of the scan results (FPR) file you upload to Fortify Software Security Center using the `-fprssc` option with the `start` command. The file name must not exceed 128 characters in length and *must not* contain the following characters:

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

The following example scan request performs a remote translation and scan and specifies a name for the FPR file to upload:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -upload -versionid
<app_version_id> -fprssc <my_fpr>.fpr
```

The following example scan request performs a remote scan and specifies a name for the FPR file for upload:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -upload -versionid
<app_version_id> -fprssc <my_fpr>.fpr -b <build_id> -scan
```

See also

"Global options" on page 98

"Start command" on page 99

Preventing replacement of duplicate scan requests

A duplicate scan request occurs if you have more than one scan requests that upload scan results to the same application version in Fortify Software Security Center. If the Controller is configured to replace duplicate scan jobs by enabling the `replace_duplicate_scans` property, you can prevent the replacement for specific scan requests with the `--disallow-replacement` (`-dr`) option in a scan request.

Consider the following scenario:

1. Submit a scan for upload to AppA 1.0, scan job 1 is added to the queue.
2. Submit a scan for upload to AppA 1.0, scan job 1 is canceled and scan job 2 is added.
3. Submit a scan for upload to AppA 1.0 with the `-dr` option, scan job 2 is canceled and scan job 3 is added to the queue.
4. Submit a scan for upload to AppA 1.0 with or without the `-dr` option, scan job 3 remains in the queue and scan job 4 is added to the queue.

The following example scan request performs a remote translation and scan, uploads the results to the application version AppA, 1.0 on Fortify Software Security Center, and overrides a duplicate replacement to ensure the scan job is not removed from the queue by future scan requests uploaded to the same application version:

```
scancentral -sscurl <ssc_url> -ssctoken <token> start -upload -application
AppA -version 1.0 --disallow-replacement
```

See also

"Configuring the Controller" on page 29

Retrying failed uploads to Fortify Software Security Center

If a job configured to upload scan results to Fortify Software Security Center fails, the Fortify ScanCentral SAST Controller retries to upload (up to five attempts by default) and, if the next attempt fails, waits two minutes before it tries again.

If the Controller fails to upload an FPR file to Fortify Software Security Center, you can use the upload command as follows to resend the FPR:

```
scancentral -sscurl <ssc_url> -ssctoken <token> upload -token <job_token>
```

where *<job_token>* is the token for original job that failed to upload the FPR.

See also

"Configuring upload to Fortify Software Security Center retry attempts" on the next page

"Uploading results to Fortify Software Security Center" on page 79

Configuring upload to Fortify Software Security Center retry attempts

To configure the number of times the Controller can retry to upload scan results, and the amount of time the Controller waits after a failed upload before it tries again:

1. Open the `<controller_install_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file in a text editor.
2. To set the maximum number of upload retry attempts, locate the `ssc_upload_retry_count` property, and replace the default value of `5` with any integer value from `1` to `10`.

   Note: If the specified value is outside of the valid range or is invalid, Fortify ScanCentral SAST applies the default value.

3. To set the interval between upload retry attempts, locate the `ssc_upload_retry_interval` property, and replace the default value of `120` (seconds) with any integer value from `60` (1 minute) to `900` (15 minutes).

   Note: If the specified value is outside of the valid range or is invalid, Fortify ScanCentral SAST applies the default value.

4. Save and close the `config.properties` file.

See also

"Uploading results to Fortify Software Security Center" on page 79

"Retrying failed uploads to Fortify Software Security Center" on the previous page

Optimizing scan performance

If you plan to regularly scan large applications, OpenText 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. Set the Fortify Static Code Analyzer scan parameters for optimal performance by adjusting the memory settings to align with your hardware.

   For information about how to tune Fortify Static Code Analyzer, see the *OpenText™ Fortify Static Code Analyzer User Guide*.
2. Run a scan.
3. Note the size of the resulting FPR file and scan log.

4. To ensure that the Fortify ScanCentral SAST Controller and Fortify Software Security Center can accept FPR or log files larger than 1 GB, increase the maximum upload size threshold by doing the following:
   a. Open the `<controller_install_dir>/tomcat/webapps/scancentral-ctrl/classes/config.properties` file.
   b. Set the Controller threshold to the maximum size in megabytes as follows:

      ```
      max_upload_size=<max_size_in_megabytes>
      ```

      The default value is 1024.
5. Make sure that Fortify Static Code Analyzer is configured to process large FPR files.

   For more information, see the *OpenText™ Fortify Static Code Analyzer User Guide*.

See also

"Configuring the Controller" on page 29

Generating a ScanCentral SAST package

Use the package command to create a ZIP archive of the specified project. The ZIP archive project package includes the following information:

- Libs—Folder that contains the project dependencies (Gradle, Maven, MSBuild, Java, and .NET projects)
- Src—Folder that contains the source files
- metadata—Specification file that the sensor uses to generate Fortify Static Code Analyzer commands

The following table provides examples of different commands to generate a project package with Fortify ScanCentral SAST client. The examples assume that the command is run from the project's working directory. In these examples, ScanCentral SAST client creates a package with the name `fortifypackage.zip` unless the `-o` option is used to specify a package name.

Note: ScanCentral SAST client can automatically detect the build tool you are using based on the project files being scanned so use of the `--build-tool` (`-bt`) option is usually not required.

Task	Example command
Create a package from a dotnet project on Linux.	`scancentral package`
Create a package from an MSBuild project.	

Task	Example command
Create a package from a dotnet project on Windows.	`scancentral package -bt dotnet`
Create a package from a Gradle project.	`scancentral package`
Create a package from a Maven project with a custom pom.xml file.	`scancentral package -bf myCustomPom.xml`
Create a package from an ABAP project.	`scancentral package`
Create a package from an Apex project.	`scancentral package`
Create a package from a Classic ASP project.	`scancentral package`
Create a package from a COBOL project.	`scancentral package -targs "-copydirs copybooks" -targs "-dialect COBOL390"`
Create a package from a ColdFusion (CFML) project.	`scancentral package`
Create a package from a Java project.	`scancentral package`
Create a package with the name MyPackage.zip from a Java project.	`scancentral package -o MyPackage.zip`
(For use with OpenText™ Fortify on Demand only) Create a package from a Java project and include additional files required for open source software composition analysis.	`scancentral package -oss`
Create a package from a JavaScript/TypeScript project.	`scancentral package`
Generate a package from an Android project in Kotlin that uses the Android plugin.	`scancentral package -bt gradle`

Task	Example command
Create a package from a Go project.	`scancentral package`
Create a package for only IaC/Dockerfiles. **Note:** If Dockerfiles are included in a Gradle, Maven, or MSBuild project, then the Docker files are automatically included in the package.	`scancentral package`
Create a package from a PHP project.	`scancentral package`
Create a package from a Python 2 project.	`scancentral package -yv 2 -pyr <requirements_file_path>`
Create a package from a Python project under an active virtual environment with dependencies already installed.	`scancentral package`
Create a package from a Python project under an active virtual environment without project dependencies installed.	`scancentral package -pyr <requirements_file_path>`
Create a package from a Python project using an existing Python virtual environment and install project dependencies.	`scancentral package -pyv <venv_location>`
Create a package from a Ruby project.	`scancentral package`
Create a package from a SQL project.	`scancentral package -targs "-sql-language TSQL"`
	`scancentral package -targs "-sql-language PL/SQL"`
Create a package from a Visual Basic project.	`scancentral package`

See also

"Package command" on page 104

"Using the PackageScanner tool" below

Open source software composition analysis (Fortify on Demand only)

Fortify on Demand customers can use the `--open-source-scan` (`-oss`) option with the `package` command to include additional files required for open source software composition analysis by OpenText™ Debricked. By default, the Fortify ScanCentral SAST client uses the Debricked CLI to automatically generate the Debricked lock files required for open source composition analysis. Using the Debricked CLI, gives you the most up-to-date Debricked artifact generation. Fortify ScanCentral SAST client installs the Debricked CLI if it is not yet installed and checks for a newer version online.

The ScanCentral SAST client installs the Debricked CLI in one of the following locations:

- Default location: *<client_install_dir>*`/Core/lib/debricked/`
- Custom location specified by the `debricked_cli_dir` property in the *<client_install_dir>*`/Core/config/client.properties` file

If you want to use the Debricked CLI without the automatic installation, you can manually place the Debricked CLI in either location. See the Debricked CLI documentation for instructions on how to download the latest releases. To avoid automatic updates of the Debricked CLI, include the `--skip-debricked-update` (`-sdu`) option in your ScanCentral SAST client package command.

Using the PackageScanner tool

If you have Fortify Static Code Analyzer locally installed, you can run an analysis of a project package without sending it to the Controller. The PackageScanner tool takes a project package created by the ScanCentral SAST client `package` command, generates Fortify Static Code Analyzer commands, and then scans it using a locally installed Fortify Static Code Analyzer. You can find the packagescanner tool in the *<sca_install_dir>*`/bin/` directory.

Note: You can set Java system properties for the PackageScanner tool to use by adding them to the SCANCENTRAL_VM_OPTS environment variable. For example, to specify a temp directory that has a short path in Windows, type:

```
set SCANCENTRAL_VM_OPTS=-Djava.io.tmpdir=C:\mytemp
```

The following table describes the PackageScanner tool command-line options.

Packagescanner option	Description
`--fpr <file>.fpr`	(Required) Specifies the FPR file to which scan results are written.
`--package <package_name>.zip`	(Required) Specifies the path to the project package file generated by the Fortify ScanCentral SAST client with the `package` command.
`-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, PackageScanner automatically generates one.
`--sca-path <path>`	(Optional if started from Fortify Static Code Analyzer) Specifies the path to the Fortify Static Code Analyzer executable. If the ScanCentral SAST client is part of the Fortify Static Code Analyzer installation (embedded), the path is determined automatically.

`-targs,`
`--translation-arguments`
`<translation_options>`
	(Optional) Specifies 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 parameter.

`-sargs,` `--scan-arguments <scan_options>`	(Optional) Specifies 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-translation-log <log_file_path>`	(Optional) Specifies a log file for translation commands. By default, PackageScanner creates the log file in a temporary directory, which is removed after the program execution.
`--sca-scan-log <path>`	(Optional) Specifies a log file for scan commands.

Packagescanner option	Description
	By default, PackageScanner creates the log file in a temporary directory, which is removed after the program execution.
`--working-dir <dir>`	(Optional) Specifies a directory where the project package is unpacked and PackageScanner creates the Fortify Static Code Analyzer project root directory. By default, PackageScanner creates this directory in a temporary location and removes it after the program execution (unless the `-debug` option is specified).
`-debug`	(Optional) Enables debug logging for Fortify ScanCentral SAST clients and sensors.
`-v`, `--version`	(Optional) Displays the PackageScanner tool version.

The following are example PackageScanner commands:

```
packagescanner --package package.zip --fpr results.fpr
packagescanner --package package.zip --fpr results.fpr --translation-
arguments "-debug -verbose" --scan-arguments "-debug -verbose"
packagescanner --package package.zip --fpr results.fpr --sca-translation-
log trans.log --sca-scan-log scan.log
packagescanner --package package.zip --fpr results.fpr --sca-path
C:\fortify\bin\sourceanalyzer.exe
packagescanner --package package.zip --fpr results.fpr --working-dir
C:\packageScannerTemp
```

See also

"Generating a ScanCentral SAST package" on page 84

Chapter 7: Managing scan requests and scan results

This section describes how to view the status of your scan requests, retrieve the scan results, and cancel scan requests from the Fortify ScanCentral SAST client command line.

You can also manage scan requests and obtain scan results from Fortify Software Security Center. For more information, see *OpenText™ Fortify Software Security Center User Guide*.

This section contains the following topics:

Viewing the scan request status

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

```
scancentral –url <controller_url> status -token <job_token>
```

You can also view the scan request status from Fortify Software Security Center. For instructions, see the *OpenText™ Fortify Software Security Center User Guide*.

The following table lists the possible values for Fortify ScanCentral SAST scan request and upload status, which are available in the console, the scan logs, and the Fortify Software Security Center user interface. The `SSC upload status` is provided only for scan requests that include uploading the scan results (FPR file) to Fortify Software Security Center.

Status type	Status	Description
Job status	PENDING	The Controller accepted the scan job.
	QUEUED	Scan job was assigned to a sensor.
	CANCELED	Scan was canceled.
	RUNNING	Scan is currently running.
	FAILED	Scan failed due to a Fortify Static Code Analyzer error.
	FAULTED	Scan failed due to a sensor error.
	TIMEOUT	Scan was canceled due to timeout.
	COMPLETED	Scan completed successfully.
SSC upload status	PENDING	Request to upload the scan results (FPR) is pending.
	QUEUED	A scan results (FPR) upload is awaiting upload to Fortify Software Security Center.
	CANCELED	A scan results (FPR) upload to Fortify Software Security Center was canceled or failed.
	FAILED	A scan results (FPR) upload to Fortify Software Security Center failed.
	COMPLETED	A scan results (FPR) file was uploaded to Fortify Software Security Center successfully.

See also

"Status command" on page 108

Retrieving scan results from the Controller

To retrieve scan results, run the following command:

```
scancentral -url <controller_url> retrieve -token <job_token> -f
<results>.fpr -log <my_log>.log
```

See also

"Retrieve command" on page 109

Canceling scan requests

To cancel a scan request, run the following command:

```
scancentral –url <controller_url> cancel -token <job_token>
```

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

See also

"Cancel command" on page 111

Chapter 8: Troubleshooting

The following topics provide information on how to troubleshoot problems you might encounter working with Fortify ScanCentral SAST and how to gather information for Customer Support.

This section contains the following topics:

Locating log files

The following table describes where to find the log files for different components.

Component	Operating system	Log file location
Controller	Windows Linux	*<controller_install_dir>*/tomcat/logs/scancentralCtrl.log **Note:** For information about how to configure the logging level for the Controller, see "Configuring the log level on the Controller" on page 95.
Sensor Client	Windows	C:\Users*<username>*\AppData\Local\Fortify\scancentral-*<version>*\log
	Linux	*<userhome>*/.fortify/scancentral-*<version>*/log

Troubleshooting the Controller

After upgrading the binaries on the local server for the Controller, you can access the Controller using the address *<protocol>*://*<controller_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.

Open the *<client_install_dir>*`/Core/Config/client.properties` file to make sure that the value set for the `client_auth_token` property matches the value for the same property in the `config.properties` file found in your Controller installation directory.

Troubleshooting a sensor as a Windows service

To troubleshoot issues encountered during the configuration of sensor auto-start as a Windows service, review the logs listed in the following table.

Log type	Default log file location[1]
Primary Fortify ScanCentral SAST sensor log	`C:\Windows\System32\config\systemprofile\AppData\Local\Fortify\scancentral-`*<version>*`\log\scancentral.log`
Sensor temporary directories 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 **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.	`C:\Users\Public\Fortify\SC\workerout.log` `C:\Users\Public\Fortify\SC\workererr.log`
Commons-daemon log	`C:\Users\Public\Fortify\SC\`*<year_month_day>*`.log`

[1] The log file location might be different if you changed the account under which the service is run or you have set the WORKDIR environment variable.

If you experience an issue starting a ScanCentral SAST sensor that you installed as a Windows service and the log files do not include enough information to resolve the issue, you can run the service as a console application to get more information. Run the following commands from an administrator command prompt:

```
cd <sca_install_dir>\bin\scancentral-worker-service
prunsrv.exe //TS//FortifyScancentralWorkerService
```

This enables you to see any service startup errors that might help you to troubleshoot the issue.

Preserving the Fortify Static Code Analyzer project root directory

By default, the ScanCentral SAST sensor creates a temporary working directory to unpack the project package and store temporary files for the scan including the Fortify Static Code Analyzer project root directory. This working directory is automatically deleted after the scan unless the `-debug` option is provided in the scan request. You can also configure an option to prevent the Fortify Static Code Analyzer project root directory from being deleted. To preserve the Fortify Static Code Analyzer project root directory:

1. Browse the *<sca_install_dir>*`/Core/config` directory and open the `worker.properties` file in a text editor.
2. Look for the `delete_sca_build_dir` property and set it to `false`.
3. Save the changes.

After the scan is complete, you can find the Fortify Static Code Analyzer project root directory in the job directory, which is in one of the following locations:

- The `jobs` directory in the sensor's working directory
- In the directory configured with the `jobs_dir` property in the `worker.properties` file

See also

"Configuring where to generate job files and the worker_persist.properties file" on page 50

Configuring the log level on the Controller

Fortify ScanCentral SAST logs typically provide enough information to follow the flow of operations under normal conditions. If things are not working as expected, the logging might not provide enough information to determine the actual root cause of the issue. If the ScanCentral SAST Controller log information is insufficient, you can increase the amount of information by changing the log level. The following instructions describe how to configure the log level on the Controller. For instructions on how to change the log level on sensors and clients, see "Enabling debugging on clients and sensors" on the next page.

To configure the log level on the Controller:

1. Open the `<controller_install_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/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 shown in the following example:

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

4. To apply the change, restart the Controller.

For more information about log levels and defining custom log levels, see the Apache Logging Services website.

See also

"Enabling debugging on clients and sensors" below

"Locating log files" on page 93

Enabling debugging on clients and sensors

Fortify ScanCentral SAST logs typically provide enough information to follow the flow of operations under normal conditions. If things are not working as expected, the logging might not provide enough information to determine the actual root cause of the issue. If the client or sensor log information is insufficient, you can increase the log level by adding the `-debug` command-line option to the ScanCentral SAST command. Make sure that you specify the `-debug` option *before* the command action.

Examples:

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

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

See also

"Configuring the log level on the Controller" on the previous page

"Locating log files" on page 93

Creating a log archive for Customer Support

If you are experiencing any issues with Fortify ScanCentral SAST, you can use the `-diag` option for the `start` command to generate a ZIP file that includes debug log files from clients, sensors, and Fortify Static Code Analyzer. You can share this ZIP when you contact Customer Support.

The following is an example command to generate the archive:

```
scancentral -url <controller_url> start --diagnosis <debug_data.zip>
```

The generated ZIP file contains the following:

- Client debug log entries for the specific scan invocation only
- Sensor debug log entries for the specific job
- The Fortify Support log from Fortify Static Code Analyzer
- MSBuild or dotnet build log

  Included only when scanning .NET projects.
- Metadata file from the project package

  Included when using remote translation and scan.

Appendix A: Fortify ScanCentral SAST command-line options

This appendix describes the command-line options that you can use with Fortify ScanCentral SAST client.

This section contains the following topics:

Global options

This topic describes the global command-line options that you can use with Fortify ScanCentral SAST client. You must specify these options before any of the commands described in the following sections.

Global option	Description
`-h,` `--help <command>`	Displays help for the selected command. To see all command help, type `-h all`.
`-v,` `--version`	Displays the Fortify ScanCentral SAST version.

Global option	Description
`-sscurl <web_address>`	Specifies the web address of a Fortify Software Security Center server that is integrated with the Controller. You must include the `-ssctoken` option with this option for authentication.
`-ssctoken <token>`	Specifies a Fortify Software Security Center authentication token of type ScanCentralCtrlToken. For information about how to acquire authentication tokens, see the *OpenText™ Fortify Software Security Center User Guide*. You must include the `-sscurl` option with this option to specify the Fortify Software Security Center server.
`-url <web_address>`	Specifies a Fortify ScanCentral SAST Controller web address. If you are uploading scan results to Fortify Software Security Center, then the Controller must be integrated with an Fortify Software Security Center instance. **Note:** Do not include the `-sscurl` and `-ssctoken` option pair with this option.
`-debug`	Enables debug logging on Fortify ScanCentral SAST clients and sensors. For information on how to configure the logging level on the Controller, see "Configuring the log level on the Controller" on page 95.

Start command

Use the `start` command to perform a remote scan, or to perform a remote translation and scan.

Start option	Description
Options for all scan requests	
`-upload`, `--upload-to-ssc`	Uploads the scan results to Fortify Software Security Center after completion. For more information about uploading scan results, see "Uploading results to Fortify Software Security Center" on page 79.

Start option	Description
`-application <name>`	Specifies the Fortify Software Security Center application name. The *<name>* value is case-sensitive.
`-version`, `--application-version <name>`	Specifies the Fortify Software Security Center application version name. The *<name>* value is case-sensitive.
`-versionid`, `--application-version-id <id>`	Specifies the Fortify Software Security Center application version ID.
`-uptoken`, `--ssc-upload-token <token>`	Specifies the Fortify Software Security Center authentication token of type ScanCentralCtrlToken, which is only required if you are uploading scan results and specify the Controller with the global `-url` option. **Note:** If the `pool_mapping_mode` property is set to `disabled`, you can also use a token of type AnalysisUploadToken. For information about how to acquire authentication tokens, see the *OpenText™ Fortify Software Security Center User Guide*.
`-fprssc`, `--fpr-filename-on-ssc <file>`	Specifies the name to use for the FPR file uploaded to Fortify Software Security Center. For more information about this option, see "Specifying a scan results (FPR) file name" on page 81.
`-dr`, `--disallow-replacement`	Prevents a scan job from being replaced because it is a duplicate (targeted to the same application version as an existing queued scan job). For more information about this option, see "Preventing replacement of duplicate scan requests" on page 82.
`-block`	Waits for the job to complete, and then downloads the scan results.
`-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.

Start option	Description
`-diag`, `--diagnosis <zip_file>`	Generates a ZIP file that includes debug log information from client, sensor, and Fortify Static Code Analyzer that Customer Support requires to analyze any problems you might encounter. For more information about this option, see "Creating a log archive for Customer Support" on page 97.
`-email <address>`	Specifies the email address for job status notifications.
`-filter <file>`	Specifies a filter file to use during a scan (repeatable).
`-log`, `--log-file <file>`	Specifies a file name for the local log file after the scan is complete.
`-o`, `--overwrite`	Overwrites the existing FPR or log with new data.
`-projtl`, `--project-template <file>`	Specifies an issue template file to include.
`-pi`, `--poll-interval <n>`	Specifies how often (in seconds) to poll the processing status. The valid range for *<n>* is from `10` to `60`.
`-pool`, `--submit-to-pool <uuid>`	Specifies a specific sensor pool for the scan request.
`-sto`, `--scan-timeout <n>`	Specifies the maximum amount of time (in minutes) a sensor can work on an assigned job (and prevent the sensor from doing other work). Use of this option has a higher priority than the `scan_timeout` property setting in the `config.properties` file.
`-rules <file/dir>`	Specifies a custom rules file or directory to use during the scan (repeatable).
`-sp`, `--save-package <file>`	Specifies the project package file to save after submitting the scan request. The *<file>* must have a `*.zip` extension. This project package contains the following information: • Libs—Folder that contains the project dependencies (Gradle, Maven, MSBuild, Java, and .NET projects) • Src—Folder that contains the source files

Start option	Description
	• metadata—Specification file that the sensor uses to generate Fortify Static Code Analyzer commands
Options for local translation and remote scan requests	
`-b`, `--build-id <id>`	Specifies the build ID of a previously translated project to upload to the Controller for analysis.
`-mbs <file>`	Specifies a mobile build session file for a previously translated project to upload to the Controller for analysis.
`-projroot`, `--project-root <dir>`	Specifies the project directory for the mobile build session export.
`-scan`	Sets the point beyond which all options are for Fortify Static Code Analyzer.
Options for remote translation and scan requests	
`-p`, `--package <file>`	Specifies the project package file to upload to the Controller (see "Package command" on page 104).
`-bt`, `--build-tool <name>`	Specifies the build tool used for the project. The valid values for *<name>* are `dotnet`, `gradle`, `msbuild` (Windows only), `mvn`, or `none`. The following example specifies a maven project with build parameters: `-bt mvn -bc "package --setting custom.xml"` If not specified, Fortify ScanCentral SAST automatically detects the build tool based on the project files being scanned.
`-bc`, `--build-command <commands>`	(For use with Maven, Gradle, dotnet, and MSBuild) Specifies custom build parameters for preparing and building a project. The following example build command starts a Gradle build before packaging the project: `-Prelease=true clean customTask build` If you use the `-bc` option and the build fails, ScanCentral SAST stops working on the build.

Start option	Description
	(Gradle only) If you *do not* use `-bc`, the default command, default tasks, and target are invoked. If the build fails, ScanCentral SAST displays a warning, but continues to work and then displays a message to indicate that the build procedure failed and your results might be incomplete.
`-bf`, `--build-file <file>`	Specifies the build file if you are not using a default name such as `build.gradle` or `pom.xml`.
`-q`, `--quiet`	Prevents the printing to stdout from the build execution.
`-skipBuild`	Disables the project preparation build step before packaging. If you use this option, any `-bc` option specified is ignored. Only apply this option for Gradle and Maven builds.
`-t`, `--include-test`	(Java projects only) Includes the test source set (Gradle) or test scope (Maven) to scan.
`-exclude <file>`	Specifies a file or directory (with absolute or relative path, or Ant-style path pattern) to exclude from a project package (repeatable).
`-hv`, `--php-version <version>`	Specifies the PHP version. If not specified, ScanCentral SAST automatically detects the installed PHP version.
`-pyr`, `--python-requirements <file>`	Specifies the Python project requirements file to install and collect dependencies.
`-pyv`, `--python-virtual-env <dir>`	Specifies the Python virtual environment location.
`-yv`, `--python-version <version>`	Specifies the Python version. The valid values are `2` and `3`. This option is ignored if ScanCentral SAST client is started under a Python virtual environment or if `--python-virtual-env` is specified.
`-targs`, `--translation-args <translation_option>`	Specifies a Fortify Static Code Analyzer translation option (repeatable). For multiple translation options, use multiple `-targs` options. If the translation option has a path parameter that includes a space, enclose the path in single quotes. For a list

Start option	Description
	of the Fortify Static Code Analyzer options you can use with the `-targs` option, see "Options accepted for -targs (--translation-args)" on page 107. If you use the `-targs` option with the `--package` option, Fortify ScanCentral SAST ignores it and displays an error message.
`-sargs`, `--scan-args <scan_option>`	Specifies a Fortify Static Code Analyzer scan options (repeatable). For multiple scan options, use multiple `-sargs` options. If the scan option has a path parameter that includes a space, enclose the path in single quotes. For a list of the Fortify Static Code Analyzer options you can use with the `-sargs` option, see "Options accepted for -sargs (--scan-args)" on page 108.

Package command

Use the `package` command to create a ZIP archive (project package) of your project. You can either:

- Upload this project package to the Controller with the ScanCentral SAST `start` command
- Run an analysis with a locally installed Fortify Static Code Analyzer using the PackageScanner tool
- Upload this project package to Fortify on Demand for analysis

Caution! To avoid a packaging failure for projects with file paths that contain an umlaut, you must first add the `com.fortify.sca.CmdlineOptionsFileEncoding` property to the *<sca_install_dir>*`/Core/config/fortify-sca.properties` file and specify a value for it that is not ASCII encoding.

Package option	Description
`-bt`, `--build-tool <name>`	Specifies the name of the build tool used for the project. The valid values for *<name>* are `dotnet`, `gradle`, `msbuild` (Windows only), `mvn`, and `none`. If not specified, Fortify ScanCentral SAST automatically detects the build tool based on the project files being scanned.

Package option	Description
`-bc`, `--build-command <commands>`	(For use with Maven, Gradle, dotnet, and MSBuild) Specifies custom build parameters for preparing and building the project. The following example build command starts a Gradle build before packaging: `-Prelease=true clean customTask build` If you use the `-bc` option, and the build fails, ScanCentral SAST 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 SAST displays a warning, but continues to work and then displays a message to indicate that the build procedure failed and you might get incomplete results.
`-bf`, `--build-file <file>`	Specifies the build file if you are not using a default name such as `build.gradle` or `pom.xml`.
`-q`, `--quiet`	Prevents the printing of stdout from the build execution.
`-skipBuild`	Disables the project preparation build step before packaging. If you use this option, any `-bc` option specified is ignored. Only apply this option for Gradle and Maven builds.
`-t`, `--include-test`	(Java projects only) Includes the test source set (Gradle) or test scope (Maven) to scan.
`-exclude <file>`	Specifies a file or directory (with absolute or relative path, or Ant-style path pattern) to exclude from a project package (repeatable).
`-hv`, `--php-version <version>`	Specifies the PHP version. If not specified, ScanCentral SAST automatically detects the installed PHP version.
`-oss`, `--open-source-scan`	(For use with Fortify on Demand only) Specifies to generate and collect additional files for open source software composition analysis. For details, see the *OpenText™ Fortify on Demand User Guide*.

Package option	Description
`-sdu`, `--skip-debricked-update`	(For use with Fortify on Demand only) Specifies not to check for an updated version of the Debricked CLI. If this option is specified and no Debricked CLI is currently installed, then ScanCentral SAST generates and collects the additional files for open source software composition analysis without the Debricked CLI. You must also specify the `-oss` option to use this feature.
`-pyr`, `--python-requirements` *`<file>`*	Specifies the Python project requirements file to install and collect dependencies.
`-pyv`, `--python-virtual-env` *`<dir>`*	Specifies the Python virtual environment location.
`-yv`, `--python-version` *`<version>`*	Specifies the Python version to automatically find the installed Python. The valid values are `2` and `3`. This option is ignored if ScanCentral SAST client is started under a Python virtual environment or if `-python-virtual-env` is specified.
`-targs`, `--translation-args` *`<option>`*	Specifies a Fortify Static Code Analyzer translation option (repeatable) For multiple translation options, use multiple `-targs` options. If the translation option has a path parameter that includes a space, enclose the path in 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 the next page.
`-o`, `--output` *`<file>`*	Specifies the output file name. The file extension must be `*.zip`. If not specified, ScanCentral SAST writes the project package to a ZIP archive with the name `fortifypackage.zip`.

See also

"Generating a ScanCentral SAST package" on page 84

"Using the PackageScanner tool" on page 87

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

This topic lists the Fortify Static Code Analyzer translation options you can use with the Fortify ScanCentral SAST `-targs` option. You can use these options with the ScanCentral SAST `start` and `package` commands. For descriptions of the Fortify Static Code Analyzer translation options listed in this topic, see the *OpenText™ Fortify Static Code Analyzer User Guide*.

-autoheap

-abap-includes

-appserver

-appserver-home

-appserver-version

-build-label

-build-project

-build-version

-checker-directives

-copydirs

-cp, -classpath

-debug

-debug-mem

-debug-verbose

-dialect

-disable-language

-django-disable-autodiscover

-django-template-dirs

-enable-language

-encoding

-exclude

-extdirs

-gotags

-gopath

-goproxy

-jdk, -source

-jvm-default

-noextension-type

-php-source-root

-php-version

-python-no-auto-root-calculation

-python-path

-python-version

-quiet

-ruby-path

-rubygem-path

-show-unresolved-symbols

-source-base-dir

-sourcepath

-sql-language

-v, -version

-verbose

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

This topic lists the Fortify Static Code Analyzer scan options you can use with the Fortify ScanCentral SAST `-sargs` option. You can use these options with the Fortify ScanCentral SAST `start` command. For descriptions of the Fortify Static Code Analyzer scan options listed in this topic, see the *OpenText™ Fortify Static Code Analyzer User Guide*.

-analyzers	-disable-source-bundling	-no-default-rules
-autoheap	-disable-metatable	-no-default-sink-rules
-bin, -binary-name	-enable-analyzer	-no-default-source-rules
-build-label	-filter	-p , -scan-precision
-build-project	-fvdl-no-descriptions	-project-template
-build-version	-fvdl-no-enginedata	-quick
-debug	-fvdl-no-progdata	-quiet
-debug-mem	-fvdl-no-snippets	-rules
-debug-verbose	-legacy-jsp-dataflow	-sc, -scan-policy
-disable-analyzer	-machine-output	-v, -version
-disable-default-rule-type	-no-default-issue-rules	-verbose

Status command

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

Status option	Description
`-ctrl`	Checks whether the Controller is running.
`-token`, `--job-token` *<token>*	Specifies the job token for a remote scan job.
`-bl`, `--block-until` *<action>*	Specifies to have the process (scan or merge) wait until the Fortify Software Security Center FPR upload and processing are complete, and then download the merged FPR file from Fortify Software Security Center.

Status option	Description
	The following values are valid for *<action>*: • `scan`—Direct the scan process to continue to run until the scan is complete and available on the Controller. • `sscproc`—Wait for Fortify Software Security Center processing to complete. If the scan results file (FPR) is not uploaded to Fortify Software Security Center, an error occurs.
`-bto`, `--block-timeout` *<n>*	Specifies how long (in minutes) to block processing. The valid range for *<n>* is from `0` to `10080` minutes. Specify `0` for no timeout.
`-pi`, `--poll-interval` *<n>*	Specifies how frequently (in seconds) to poll the processing status. The valid range for *<n>* is from `10` to `60`.

See also

"Viewing the scan request status" on page 90

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 or later, some sensor configuration is required to use the `progress` command . For instructions, see "Configuring sensors to use the progress command when starting on Java" on page 50.

Retrieve command

Use the `retrieve` command to download the scan results, log files, and job file for a remote scan job from the Fortify ScanCentral SAST Controller.

Retrieve option	Description
`-token`, `--job-token` *<token>*	Specifies the job token for a remote scan job.

Retrieve option	Description
`-f`, `--output-file <file>.fpr`	Specifies a file name for the local scan results (FPR) file.
`-j`, `--job-file <file>.zip`	Specifies a file name for the local job file that was submitted to Fortify ScanCentral SAST for analysis. The job file for remote translation contains the project package (sources, dependencies, and metadata). The job file for local translation contains the mobile build session (MBS) file.
`-log`, `--log-file <file>`	Specifies a file name for the local Fortify Static Code Analyzer log file.
`-slog`, `--sensor-log-file <file>`	Specifies the file name for the local sensor log output.
`-o`, `--overwrite`	Overwrites an existing scan results (FPR), log, or job file with new data.
`-block`	Specifies to wait for the job to complete and then download the scan results.
`-bto`, `--block-timeout <n>`	Specifies how long (in minutes) to block processing. The valid range for *<n>* is from `0` to `10080` minutes. Specify `0` for no timeout. The default value is `0`.
`-pi`, `--poll-interval <n>`	Specifies how frequently (in seconds) to poll the processing status. The valid range for *<n>* is `10` to `60` seconds.

See also

"Retrieving scan results from the Controller" on page 92

Upload command

Use the `upload` command to resend an FPR file to Fortify Software Security Center after a previous upload attempt failed.

Upload option	Description
`-token`,	Specifies the job token for the remote scan job to resend an

Upload option	Description
`--job-token <token>`	FPR file to Fortify Software Security Center.

See also

"Retrying failed uploads to Fortify Software Security Center" on page 82

Cancel command

Use the `cancel` command to cancel a pending or running remote scan job.

Cancel option	Description
`-token`, `--job-token <token>`	Specifies the job token for the remote scan job you want to cancel.

See also

"Canceling scan requests" on page 92

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 them to the next major version.

Examples:

```
scancentral -url <controller_url> update
```

or

```
scancentral -sscurl <ssc_url> -ssctoken <token> update
```

Worker command

Use the `worker` command to assign a sensor pool or set a timeout.

Worker option	Description
`-pool`, `--assign-to-pool` *<uuid>*	Specifies the sensor pool to which the sensor is 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` *<n>*	Specifies the maximum amount of time (in minutes) a sensor can work on an assigned job (and prevent the sensor from doing other work). Use of this worker option has a higher priority than the `scan_timeout` property setting in the `config.properties` file.

Send Documentation Feedback

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

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

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

Feedback on Installation, Configuration, and Usage Guide (Fortify ScanCentral SAST 24.4.0)

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@opentext.com.

We appreciate your feedback!