opentext™

OpenText™ Fortify ScanCentral SAST

Software Version: 23.2.0

Installation, Configuration, and Usage Guide

Document Release Date: Revision 1: January 4, 2024
Software Release Date: December 2023

Legal Notices

Open Text Corporation

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

Copyright Notice

Trademark Notices

Documentation Updates

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

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

This document was produced on December 21, 2023. To check for recent updates or to verify that you are using the most recent edition of a document, go to:

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

Contents

Preface

Contacting Fortify Customer Support

Visit the Support website to:

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

https://www.microfocus.com/support

For More Information

For more information about Fortify software products:

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

About the Documentation Set

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

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

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

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

Fortify Product Feature Videos

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

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

Change Log

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

Software Release / Document Version	Changes
23.2.0 / Revision 1: January 4, 2024	Updated: • Changed the requirements for when to run the migration script when upgrading the ScanCentral SAST Controller. (see "Upgrading the Controller" on page 61)
23.2.0	Added: • "Optional Kubernetes and Docker Deployment" on page 16 • "Installing the Controller as a Service on Linux" on page 21 • "Managing the Controller Service on Linux" on page 22 • "Fortify ScanCentral SAST API" on page 39 • "Working with COBOL Projects" on page 73 and added supported COBOL-related options (see "Options Accepted for -targs (--translation-args)" on page 103) • "Retrying Failed Uploads to Fortify Software Security Center" on page 76 • "Preserving the Fortify Static Code Analyzer Project Root Directory" on page 88 • "Creating a Log Archive for Customer Support" on page 90 Updated: • Updates for analyzing .NET projects (see "Configuring Sensors to Offload Translation for .NET Languages" on page 46 and "Working with .NET Projects" on page 68) • Added descriptions of the scan status values (see "Viewing the Scan Request Status" on page 83) • Added supported `-scan-policy` option (see "Options Accepted for -sargs (--scan-args)" on page 104)

Software Release / Document Version	Changes
	• Added supported COBOL options (see "Options Accepted for -targs (--translation-args)" on page 103) Removed: • The `arguments` command is deprecated and was removed from the document. Use the `-targs` or `-sargs` option with the `start` or `package` commands instead.
23.1.0 / Revision 1: June 2023	Updated: • Step 1 was revised to 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 42). • In "Status Command Options" on page 92, the `--block-for` option was changed to `--block-until` and removed from the list of start command options.
23.1.0	Added: • "Securing Tomcat Server" on page 15 • "Configuring Where to Generate Job Files and the worker_persist.properties File" on page 47 Updated: • The `client_zip_location`, `ssc_restapi_connect_timeout`, and `ssc_restapi_read_timeout` properties were added in "Configuring the Controller" on page 27. The `lim_proxy_server`, `remote_ip_proxy_header`, and `ssc_trusted_proxies_remote_ip` properties were removed. • "Avoiding Read Timeout Errors" on page 37 was updated to describe the procedures for configuring timeout between the Controller and sensors, between the Controller and clients, and between the Controller and Fortify Software Security Center. • Information about auto-detection of the build tool was added to "Offloading Both Translation and Scanning" on page 67. • The `--block-timeout`) and `--poll-interval` options were

Software Release / Document Version	Changes
	added to the `retrieve` and `status` command options (see "Retrieve Command Options" on page 99 and "Status Command Options" on page 92) • The list of accepted Fortify Static Code Analyzer options was added (see "Options Accepted for -targs (--translation-args)" on page 103 and "Options Accepted for -sargs (--scan-args)" on page 104) Removed: • Configuring the Logging Level for Sensors
22.2.0	Added: • "Configuring Proxies for Clients and Sensors" on page 59 • "Specifying the Scan Results (FPR) File Name" on page 75 • "Configuring Java Memory for the Service" on page 20 • "Working with Java 8 Projects" on page 74 Updated: • Modified the procedure for upgrading the Controller (see "Upgrading the Controller" on page 61) • The `-application-version` option was removed from "Start Command Options" on page 93. • The `-fprssc`, (`--fpr-filename-on-ssc`), `-versionid` (`--application-version-id`) options were added to "Start Command Options" on page 93. • A cautionary note related to file paths that include an umlaut was added to "Package Command Options" on page 100. • (Fortify on Demand only) The `-oss` option was added to "Package Command Options" on page 100. Removed: • The `allow_insecure_clients_with_empty_token` property was removed from the list of Controller properties.
22.1.0 / Revision 2 - August 25, 2022	Updated: • Changes were made to the `Start` command (see "Start Command

Software Release / Document Version	Changes
	Options" on page 93)
22.1.0	Updated: • Added information about installing a standalone client and instructions on how to create multiple standalone clients of different supported versions in the Controller (see "Installing Sensors" on page 42) • Added the `update` command (see "Update Command" on page 103)

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) provided for this purpose. In addition to freeing up build machines, this process makes it easy to add more resources to the scan machines as needed, without having to interrupt your build process. It's simple-to-use interface enables integration of static analysis with the build process and provides the ability to dynamically scale the sensors needed to perform the work required of the CI/CD pipeline with respect to running scans.

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

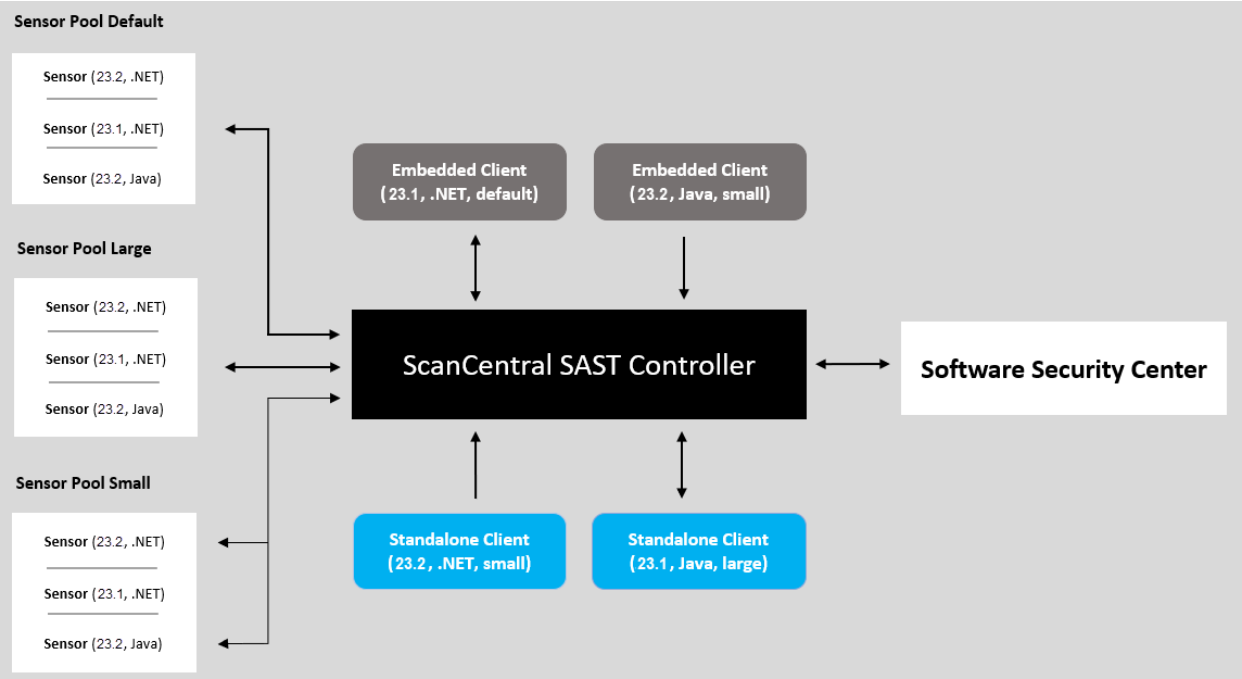
- Offload Translation and Scanning—You can offload the complete analysis to the sensors. Your application must be written in a language supported for remote translation. For information about the languages supported, see the *Fortify Software System Requirements* document. If your code is written using a language other than one supported for offloading translation, then you must offload the scanning only.
- Offload Scanning Only—You can perform the translation phase (less processor- and time-intensive than the scan phase) on a local or build machine. After the translation is complete, ScanCentral SAST moves the Fortify Static Code Analyzer mobile build session (MBS) to the sensors for scanning.

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. A OpenText™ Fortify Software Security Center server is optional.

- **ScanCentral SAST Controller**—A standalone web application that receives the Fortify Static Code Analyzer mobile build sessions (MBS) and scan instructions from ScanCentral SAST clients (or project packages with translation and scan instructions), routes the information to sensors, and (optionally) uploads scan results (FPR files) to Fortify Software Security Center. For more detail, see "About the Fortify ScanCentral SAST Controller" on page 19.
- **ScanCentral SAST client**—A build machine on which Fortify Static Code Analyzer translates code and generates Fortify Static Code Analyzer mobile build sessions (MBS). The translated source code, along with optional and required data, such as custom rules and Fortify Static Code Analyzer command-line options, are uploaded to the ScanCentral Controller. Clients can also generate packages for remote translation, independent of Fortify Static Code Analyzer. For more detail, see "About Fortify ScanCentral SAST Clients" on page 55.
- **ScanCentral SAST sensors**—Distributed network of computers set up to receive scan requests and analyze code using Fortify Static Code Analyzer. A sensor accepts either a mobile build session (MBS) file and performs a scan, or it accepts a project package that contains sources and dependencies, which it translates and scans. For more detail, see "About Fortify ScanCentral SAST Sensors" on page 42.

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

To successfully deploy Fortify ScanCentral SAST, complete the following tasks in the order listed here:

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

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

Securing 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, Fortify 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. Fortify also recommends that you use only strong cipher suites with Tomcat. Finally, take any additional steps necessary to secure Tomcat Server in your operating environment.

Using Secure Cipher Suites

Fortify 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 9 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 GitHub at https://github.com/fortify/helm3-charts.

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.
What's New in Fortify Software <version> Fortify_Whats_New_<*version*>.pdf	This document describes the new features in Fortify Software products.

Fortify Software Security Center

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

Document / File Name	Description
OpenText™ Fortify Software Security Center User Guide SSC_Guide_<*version*>.pdf	This document provides Fortify Software Security Center users with detailed information about how to deploy and use Fortify Software Security Center. It provides all of the information you need to acquire, install, configure, and use Fortify Software Security Center. It is intended for use by system and instance administrators,

Document / File Name	Description
	database administrators (DBAs), enterprise security leads, development team managers, and developers. Fortify Software Security Center provides security team leads with a high-level overview of the history and current status of a project.

Fortify Static Code Analyzer

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

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

Chapter 2: 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 the 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 update your Controller, see "Upgrading Fortify ScanCentral SAST Components" on page 60 and "Upgrading the Controller" on page 61.

Important!

- Before you install the Controller, you must first download and configure a Java Runtime Environment (JRE). For information about supported JRE versions, see the *Fortify Software System Requirements* document. For information about how to download and configure a JRE, see the documentation for the supported JRE version.
- If you plan 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 to 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 administrative 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 `Tomcat9`.

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 `tomcat9w.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 9.0 service for the Controller:

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

```
service.bat remove
```

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

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

```
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 19.
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 URL

In this guide, *<controller_url>* refers to a correctly formatted Fortify ScanCentral SAST URL. The correct format for the Controller URL 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 CLI. 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 located in the United States, type `US`.
Confirm your entries:	Type `yes` to confirm your entries.
Enter key password for <tomcat><Return if same as keystore password>:	Password for your Tomcat server key or press **Enter** to use the same password you established for your keystore. Fortify recommends that you create a new key password.
Re-enter new password:	Re-type your key password.

3. To export the certificate from the Tomcat keystore, open a command prompt and type 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. Navigate to *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes`, and then open the `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 56 and, "Installing Sensors" on page 42, respectively.
9. Add your self-signed certificate to the Java keystore on all entities that communicate with the Controller (includes all clients, sensors, and Fortify Software Security Center installations) as follows:
   a. For 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 located in the United States, type `US`.
Confirm your entries:	Type `yes` to confirm your entries.
Enter key password for `<tomcat><Return if same as keystore password>`:	Type a password for your Tomcat server key, or press **Return** to use the same password you established for your keystore. Fortify recommends that you create a new password.
Re-enter new password:	Re-type your key password.

3. Generate a Certificate Signing Request (CSR).

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

   ```
   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. Navigate to *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes`, and then open the `config.properties` 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 be used, 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 URL.

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

To configure the Controller:

1. Navigate to *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes`.

2. Open the `config.properties` file in a text editor, and then configure the properties listed in the following table.

Property	Description
`accept_job_when_no_sensor_available`	Determines whether to accept or reject scan requests submitted by clients if no compatible sensors (or compatible versions) are available. The default value is `true`. In the following examples, the property is set to `false`: • If a version 22.2 client submits a scan request, and only version 23.2 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.
`cleanup_period`	Specifies the frequency (in minutes) with which expired jobs and sensors are cleaned up. The default is `60`.
`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 36.
`client_auto_update`	If set to `true`, enables the Controller to automatically update all outdated sensors and clients. For details, see "Enabling Automatic Updates of Clients and Sensors" on page 64.
`client_zip_location`	Specifies the location of the directory that contains Fortify ScanCentral SAST client ZIP files. To enable remote upgrades of one or multiple client versions, place them in this directory. (You can use any ZIP file names.) `client_zip_location=${catalina.base}/client`
`db_dir`	Fortify ScanCentral SAST database home directory. The default value is `${catalina.base}/cloudCtrlDb`.

`email_allow_list`	Specifies the list of email domains that the Controller can use to send notifications.
	Examples of valid values:
	`*@yourcompanyname.com`
	`*@*yourcompanyname.com`

Property	Description

	`a*@yourcompanyname.com`
	`name@yourcompanyname.com`
	To specify multiple values, you can use commas (,), colons (:), or semicolons (;) as delimiters.

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

`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 `false`.
`from_email`	Specifies the email address of the sender.
`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, jobs are deleted from the Controller after 168 hours (or 7 days).
`job_file_dir`	Specifies the job storage directory. The default value is: `${catalina.base}/jobFiles`.

Property	Description
`lim_license_pool`	Specifies the name of the OpenText™ Fortify License and Infrastructure Manager 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 36.
`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 username 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 page 36.
`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 36.
`lim_server_url`	Specifies the URL for the LIM server Web Site.
`max_upload_size`	Specifies the maximum size (in megabytes) of files that can be 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 "About the pool_mapping_mode Property" on page 34.
`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. Otherwise, you can comment it out.
`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

Property	Description
	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 a specific sensor (see "Setting the Maximum Run Time for Scans" on page 45 and "Start Command Options" on page 93).
`smtp_host`	Specifies the SMTP server host name.
`smtp_port`	Specifies the SMTP server port number.
`smtp_ssl`	If set to `true`, the Controller uses SSL for connections to the SMTP server. Otherwise, it does not use SSL (default).
`smtp_ssl_check_trust`	If set to `false`, the SMTP server certificate is always trusted. Otherwise, the certificate trust is based on the certification path (the default).
`smtp_ssl_check_server_identity`	If set to `false`, the SMTP server identity is not checked. Otherwise, the Controller checks server identity, as specified by RFC 2595 (the default).

`smtp_auth_user`
/
`smtp_auth_pass`
	If your SMTP server requires authentication, uncomment both the `smtp_auth_user` and `smtp_auth_pass` properties and set their values. Otherwise, leave both properties commented out.
	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 36.

`ssc_lockdown_mode`	If set to `true`, ScanCentral SAST clients are forced to 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.

Property	Description
	In SSC lockdown mode, you: • Cannot use the client command `-url` option, but must use the `-ssc_url` option with the `-ssc_token` option instead • Must specify the application name and version, or the application version ID, and the `-upload` option when starting the scan • Cannot use the `-pool` option, because the job is automatically assigned to the pool configured for the specified application version
`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 `ssc_remote_ip_trusted_proxies_range` 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.
`ssc_restapi_read_timeout`	Specifies the Fortify Software Security Center connection 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.
`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.

Property	Description
	(Optional) Use an encrypted shared secret. For instructions on how to encrypt a shared secret, see "Encrypting the Shared Secret on the Controller" on page 36. **Note:** The `ssc_cloudctrl_secret` property is supported for backward compatibility with Fortify CloudScan.
`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 76.
`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 76.

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

`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 36.
`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 36.
`this_url`	Specifies the URL for the Controller; used in emails to refer to this server for manual job result downloads. Example: `https://<controller_host>:8443/scancentral-ctrl`
`use_starttls`	If set to `true`, uses the STARTTLS protocol command (Opportunistic

Property	Description
	SSL/TLS) to inform the SMTP server that the email client wants to upgrade from an insecure connection to a secure connection using SSL/TLS. The default is `false`.
`worker_auth_token`	Specifies a string that contains no spaces or backslashes used to secure the Controller for use by authorized sensors only. If you prefer not to use plain text, you can use an encrypted shared secret as the value for this property. For instructions on how to encrypt this value, see "Encrypting the Shared Secret on the Controller" on page 36.
`worker_expiry_delay`	Specifies the number of hours after a sensor stops communicating that it becomes a candidate for cleanup. The default is 168 hours (or 7 days).
`worker_inactive_delay`	Specifies the number of minutes after a sensor becomes inactive that all of its unfinished jobs are marked as faulted. Assign a value that is much larger than `worker_stale_delay`. Note that this property uses different time units than does `worker_stale_delay`.
`worker_stale_delay`	Specifies the number of seconds after a sensor stops communicating that it becomes stale. Assign a value that is larger than the `worker_sleep_interval` and `worker_jobwatcher_interval` defined for any sensor.

3. Save and close your `config.properties` file.
4. Start the Controller.

   For instructions, see "Starting the Controller" on page 37.

See Also

"Installing the Controller" on page 19

"Stopping the Controller" on page 39

"Placing the Controller in Maintenance Mode" on page 38

"Configuring Job Cleanup Timing on Sensors" on page 48

About the pool_mapping_mode Property

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

- `DISABLED`— In this mode, a 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. Or 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 default sensor pool is targeted for scan requests. A client cannot request a specific sensor pool in the ENFORCED mode.

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

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

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

Input	DISABLED	ENABLED	ENFORCED
version specified			

See Also

"Configuring the Controller" on page 27

Encrypting the Shared Secret on the Controller

Passwords exist in the ScanCentral Controller configuration file as plain text. You can encrypt the passwords and other values. You can use encrypted keys as values for the following properties:

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

To encrypt a shared secret on the Controller:

1. At the command prompt, run the following command: *<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 sensors is different from the pwtool key file you use to encrypt secrets on the Controller.

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

   `client_auth_token`, `lim_license_pool_password`, `lim_proxy_password`, `lim_proxy_user`, `smtp_auth_pass`, `ssc_scancentral_ctrl_secret`, `swagger_password`, `swagger_username`, `worker_auth_token`

   Tip: Fortify 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. Create additional encrypted shared secrets (steps 1 and 2) and, in the `config.properties` file, paste these as values for the two properties to which you did not already assign an encrypted secret in step 3.
5. Uncomment the following 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 27

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, navigate to the `<controller_install_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory and open the `config.properties` file in a text editor.
2. Increase the value of the `restapi_connect_timeout` and `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, navigate to the `<sca_install_dir>/Core/config` directory and open the `worker.properties` file in a text editor.
2. Uncomment the `restapi_connect_timeout` and `restapi_read_timeout` properties, and 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, navigate to the `<client_install_dir>/Core/config` directory and open the `client.properties` file in a text editor.
2. Uncomment the `restapi_connect_timeout` and `restapi_read_timeout` properties, and set the value of each to an acceptable threshold (in milliseconds).
3. Save the changes.

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 20.

To start the Controller manually:

1. If you plan 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" below

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

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

1. Log on to Fortify Software Security Center as an administrator and open the Fortify ScanCentral SAST page.
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 54

"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 Fortify ScanCentral SAST page.
2. In the left pane of the SAST page, select **Controller**.
3. Click **END MAINTENANCE MODE**.

See Also

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

"Stopping the Controller" below

Stopping the Controller

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

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" above

"Safely Shutting Down Sensors" on page 54

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	sensor-controller

Tasks you can perform	Request Group
artifacts	
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		√
job-controller	√	
info-controller	√	√

Request Group	Authentication token	
	client_auth_token	worker_auth_token
update-controller	√	√
core-controller	√	

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 27.
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 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 MBS scans, ScanCentral SAST supports all languages that Fortify Static Code Analyzer supports. For remote translation and scans of the prepared packages, ScanCentral SAST supports only the languages that can be used with remote translation. For information about the languages supported for performing remote translation, see "Installing Clients" on page 56.

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, Fortify recommends that you install sensors in a separate subnet. Use the sensors only as scan boxes. Fortify ScanCentral SAST supports only one sensor per machine.

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 62.

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. Navigate to the `<sca_install_dir>/Core/config` directory, and open the `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 the next page.
4. Save and close your `worker.properties` file.

See Also

"Fortify Static Code Analyzer and ScanCentral SAST Version Compatibility" on page 56

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 Fortify Static Code Analyzer version (major.minor).

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

   saved_caret_count = carets_used_on_command_line / 8

Examples:

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

```
setupworkerservice.bat 23.2 http://url.com this^^^^^^^^that
```

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

```
setupworkerservice.bat 23.2 http://url.com this^^^^^^^^^^^^^^^^that
```

- For information about how to encrypt a shared secret, see "Encrypting the Shared Secret on a Sensor" below.

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 49

See Also

"Installing Sensors" on page 42

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 59

"Avoiding Read Timeout Errors" on page 37

Encrypting the Shared Secret on a Sensor

Passwords 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**.

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

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

See Also

"Installing Sensors" on page 42

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:

```
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. Navigate to the *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory, and open the `config.properties` file in a text editor.
2. Set the `scan_timeout` property 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. Navigate to the *<controller_install_dir>*/tomcat/webapps/scancentral-ctrl/WEB-INF/classes directory, and open the 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 to Offload Translation for .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 as long as .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 .NET translation, Fortify strongly recommends that you start ScanCentral SAST sensors from a directory with a short name and path.

See Also

"Installing Sensors" on page 42

"Starting the Sensors" on page 48

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, navigate to the *<sca_install_dir>*`/Core/config` directory, and then open the `worker.properties` file in a text editor.
2. Add the following properties to the file, and specify the directories for each:
   - The `props_dir` property 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. Navigate to the *<sca_install_dir>*`/Core/config` directory, and then open the `worker.properties` file in a text editor.
2. Configure the following properties based on your scheduling needs.

Property	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.

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. Run 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" on the next page.

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 38

"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. Fortify strongly recommends that you review the instructions with your system administrator and make any changes required for your environment.

Enabling Sensor Auto-Start on Windows as a Service

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 administrative permissions.

   Sensors are dedicated machines that are meant 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 administrative account for the auto-start setup.
2. Open a command prompt and navigate to the `<sca_install_dir>\bin\scancentral-worker-service` directory.
3. Run the `setupworkerservice.bat` script with no options to see 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. Fortify recommends that you change the startup type setting to **Manual** until you verify that the sensor runs successfully. After verification, change the startup type setting to **Automatic (Delayed Start)** in Windows Services.
8. Make sure that the sensor communicates with the Controller.

Troubleshooting

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

Log Type	Log File Location
Primary Fortify	`C:\Windows\System32\config\systemprofile\AppData\Local`

Log Type	Log File Location
ScanCentral SAST sensor log	`\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`

See Also

"Installing a Sensor as a Service" on page 43

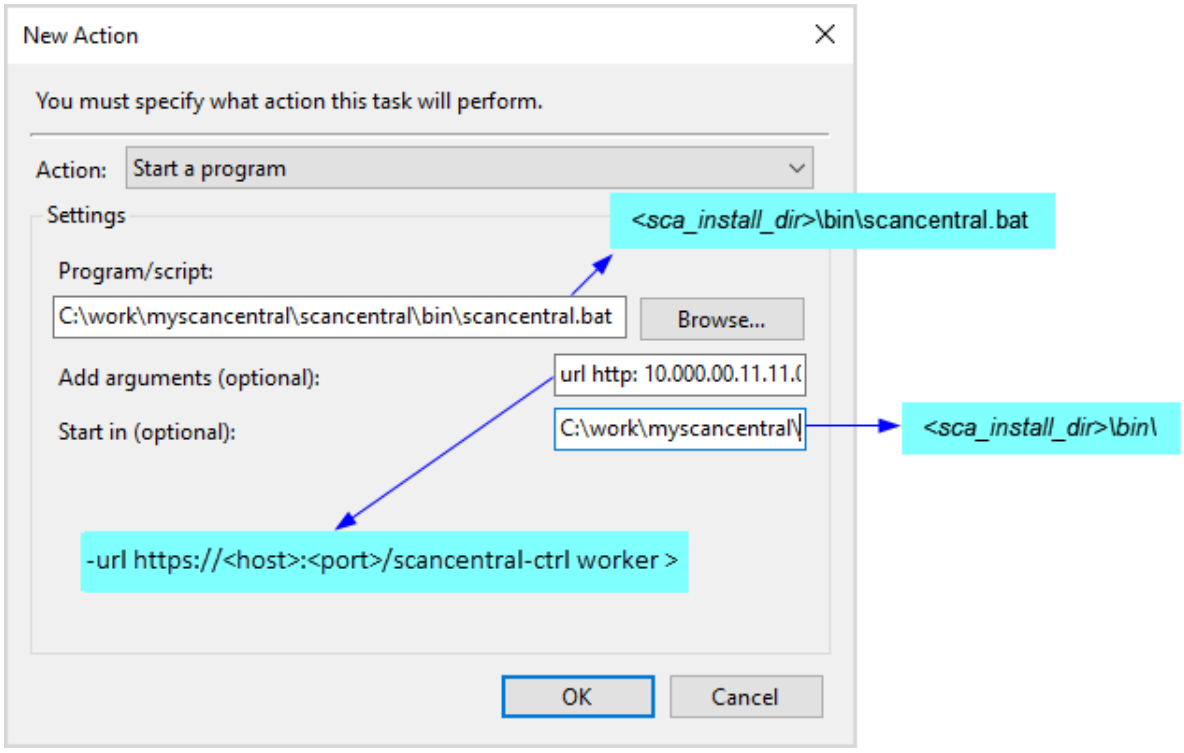
Enabling Sensor Auto-Start on Windows as a Scheduled Task

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 administrative permissions.

   Sensors are dedicated machines that are meant 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 administrative 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**.

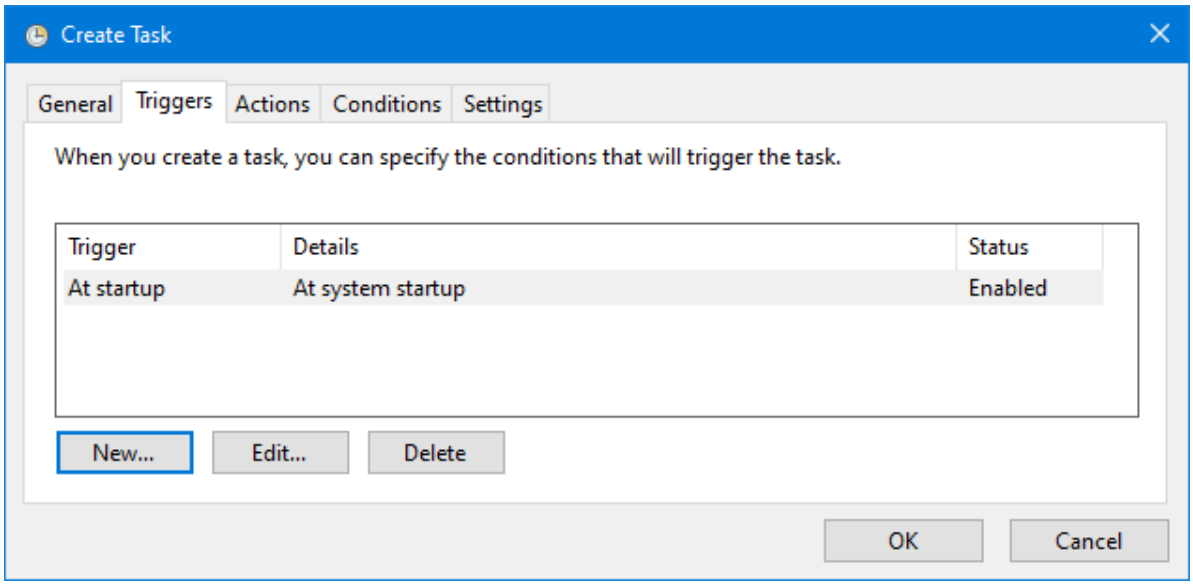
5. Click the **Actions** tab, and then click **New**.
   The New Action dialog box opens.



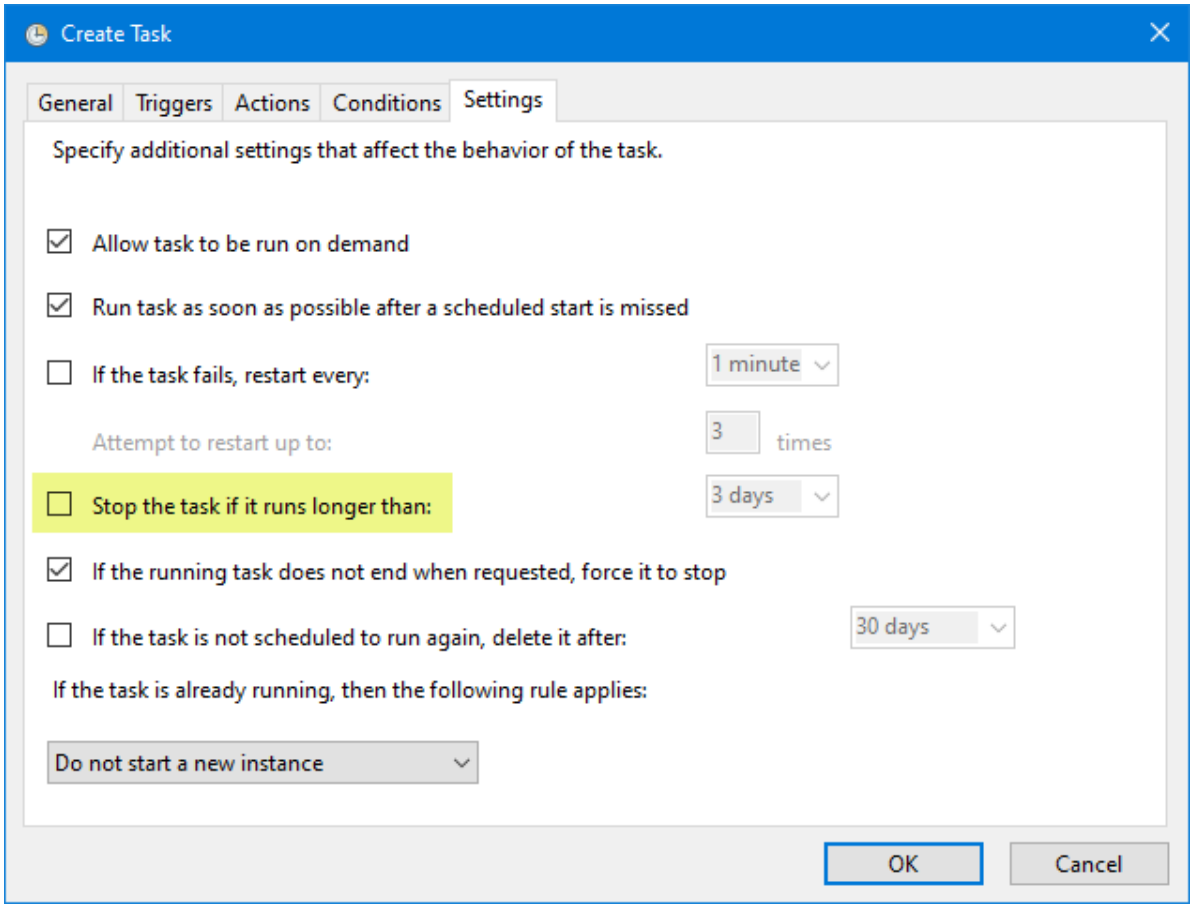
   a. In the **Action** list, select **Start a program**.
   b. In the **Program/script** box, type the directory path to your `scancentral.bat` file (for example, *<sca_install_dir>*`\bin\scancentral.bat`).
   c. In the **Add arguments (optional)** box, type the following:

      `-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. Click **Save**.

11. Restart the machine.

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

You can also start and stop the scheduled task manually from the Task Scheduler interface when 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 boot up even if sensor startup fails or hangs.
   - The double-dash (`--`) is important to separate the options for sudo from the options for your service.

   b. Make the change to the startup file.

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

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

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

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

   c. Verify that the output shows that the sensor is not started under root.

d. To verify the processes under the user, type:

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

e. Verify that the output displays the sensor process.

f. To verify the existence and contents of the script output file, type:

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

For example:

```
tail -f /opt/Fortify/Fortify_SCA_23.2.0/bin/workerout.txt
```

Safely Shutting Down Sensors

This topic describes how to move Fortify ScanCentral SAST sensors to **Shutdown** or **Shutdown scheduled** mode from Fortify Software Security Center.

Important! When the Controller is in maintenance mode (see "Placing the Controller in Maintenance Mode" on page 38), you cannot shut down sensors from Fortify Software Security Center. Also, to shut down sensors from Fortify Software Security Center, the sensors must be version 21.2.0 or later.

To shut down active sensors:

1. Log on to Fortify ScanCentral SAST as an administrator and open the Fortify ScanCentral SAST page.
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 sensor version is earlier than 21.2.0.
   - The sensor was already shut down.
   - The Controller is in maintenance mode.
   - The sensor is inactive or disabled.

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

Chapter 4: About Fortify ScanCentral SAST Clients

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

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

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.

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 their 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 initiate a scan, you invoke it using the `start` command. Sensor functionality depends on Fortify Static Code Analyzer. So, you can have a standalone client, but not a standalone sensor.

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

A standalone client, which does not require Fortify Static Code Analyzer to be installed, can create a package of the code with its dependencies to send to the Controller for translation and scanning.

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 `major.minor.patch.buildnumber` (for example 23.2.0.0068). The major and minor portions of the Fortify Static Code Analyzer version numbers on both the client and sensor must match. For example, version 23.2.0 works with version 23.2.x.

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 of the machines you plan to use as Fortify ScanCentral SAST clients. If you use a language that supports offloading the translation phase of analysis to your sensors, you can install standalone clients, independent of Fortify Static Code Analyzer. For a list of languages supported for offloading the translation phase, 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 a Standalone Client" below

"Installing an Embedded Client Using Fortify Static Code Analyzer" on page 58

Installing a Standalone Client

If you plan to offload both the translation and scanning phases of analysis to your Fortify ScanCentral SAST sensors, you can use standalone clients. These are independent of a Fortify Static Code Analyzer installation.

To install a standalone client:

1. 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.

2. Add the *<client_install_dir>*/bin to your PATH environment variable.
   The *<client_install_dir>* is the directory where you extracted the Fortify ScanCentral SAST client ZIP in step 1.
3. On the machine to which you extracted the Fortify_ScanCentral_Client_*<version>*_x64.zip file, install JRE 11 or later.
4. Set the JAVA_HOME environment variable to point to JRE 11 or later, and make sure that you add the Java executable to the PATH environment variable.

   Important! If you have a Java 8 project that fails to build because Fortify ScanCentral SAST requires Java 11 or later to run, set the SCANCENTRAL_JAVA_HOME environment variable to point a supported version of Java.
5. Navigate to the *<client_install_dir>*/Core/config directory, and then open the client.properties in a text editor.
6. 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.
7. Save and close the client.properties file.

See Also

"Placing Multiple Standalone Clients on the Controller" below

"Installing an Embedded Client Using Fortify Static Code Analyzer" on the next page

"Upgrading a Client" on page 63

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 any and all supported versions into the *<controller_install_dir>*/tomcat/client directory. The ZIP file names themselves are unimportant. 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 64.

Installing an Embedded Client Using Fortify Static Code Analyzer

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

To install an embedded client:

1. Log on to a build machine using credentials for an account that is *not* an administrator or root account.
2. Use the instructions provided in the *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" below.
5. Save and close the `client.properties` file.

See Also

"Installing a Standalone Client" on page 56

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. Copy the new encrypted secret, and paste it as the value for the `client_auth_token` property in the `client.properties` file.
4. Add the following to the `client.properties` file:

`pwtool_keys_file=<pwtool_keys_file>`

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

See Also

"Installing Clients" on page 56

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:

1. Go 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 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. To enable proxy authentication when the Controller is running under HTTPS, go to the *<client_install_dir>*`/bin` directory, and then add the following property to the scancentral executable file:

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

   Example:

```
$JAVA_CMD -Djdk.http.auth.tunneling.disabledSchemes= -
Dscancentral.installRoot="${FORTIFY_HOME}" -Dlog4j.dir="${SCANCENTRAL_
LOG}" $SCANCENTRAL_JAVA_PROPS -jar "${FORTIFY_
HOME}/Core/lib/scancentral-launcher-23.2.0.0.jar" "$@"
```

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.

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

This section contains the following topics:

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. Jobs from each client are then routed to the sensor that has the same Fortify Static Code Analyzer version installed.

If you have an existing Fortify Static Code Analyzer installation (that includes the scancentral 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 that was specified for the `props_dir` property

(see "Configuring Where to Generate Job Files and the worker_persist.properties File" on page 47).

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 that the following channels of communication are not blocked by a security system or other tool:

- Controller to Fortify Software Security Center port (for scan uploads)
- Fortify Software Security Center to the ScanCentral Controller port (for Fortify ScanCentral SAST administration console functionality)
- Clients to the 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 the `-ssc_url` option is used)

Upgrading the Controller

The following procedure describes how to upgrade the Fortify ScanCentral SAST Controller.

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

To upgrade your 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. 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.
4. 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.
5. If (and only if) you are upgrading your Controller from a 23.1.x or earlier version to version 23.2.1, run the migration script as follows.

a. Extract the contents of the `Fortify_ScanCentral_Controller_<version>_x64.zip` file.

b. Open a command prompt, and navigate to the `db-migrate` directory.

c. Identify the `cloudCtrlDb` and Controller directories for the older (existing) Fortify ScanCentral SAST version. In the following example, the existing Controller was installed on a Windows system in the `C:\scancentral22.1.0` directory:

```
C:\scancentral22.1.0\tomcat\cloudCtrlDb
C:\scancentral22.1.0\tomcat\webapps\scancentral-ctrl
```

d. Run the following command.

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

```
migrate C:\scancentral22.1.0\tomcat\cloudCtrlDb
C:\scancentral22.1.0\tomcat\webapps\scancentral-ctrl
```

The `cloudCtrlDb` directory is generated in the current working directory.

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

To change these directories, edit the `job_file_dir` and `db_dir` properties in the `config.properties` file (see "Configuring the Controller" on page 27).

7. Start the new Controller.

The database is automatically migrated.

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

See Also

"Installing the Controller" on page 19

"Upgrading Fortify ScanCentral SAST Components" on page 60

"Upgrading Sensors" below

"Enabling Automatic Updates of Clients and Sensors" on page 64

Upgrading Sensors

Important! If Fortify Static Code Analyzer is installed in a location that requires that you have administrative permissions to modify it (for example in `Program Files`), then to update a sensor you must start it with administrative 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 based on 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 44.
6. Save the `worker.properties` file.
7. Start the sensors.

See Also

"Enabling Automatic Updates of Clients and Sensors" on the next page

"Installing Sensors" on page 42

"Installing Clients" on page 56

"Upgrading Fortify ScanCentral SAST Components" on page 60

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

"Upgrading the Controller" on page 61

Upgrading a Client

Important! Fortify 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 56

"Installing an Embedded Client Using Fortify Static Code Analyzer" 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 patch or major version (for example from 23.1.0 to 23.2.0, or from 23.1.0 to 23.1.1).
- 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 23.1.0 to 23.1.1 or 23.1.2, but not to 23.2.0). 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.

About Scan Assignment

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

Important! 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. Navigate to the *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory and open the `config.properties` file in a text editor.
2. Locate the `client_auto_update` property.
3. To enable automatic updates, set `client_auto_update` to `true`.

   To turn off automatic updates, set the value to `false` (the default).
4. Save and close the file.

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

Important! If Fortify Static Code Analyzer is installed in a location that requires that you have administrative permissions to modify it (for example in `Program Files`), then to update a sensor, you must start it with administrative 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 60

"Upgrading the Controller" on page 61

Chapter 6: Submitting Scan Requests

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

This section contains the following topics:

Offloading Scanning Only

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

```
scancentral -url <controller_url> start -b <build_id> -scan
```

You can pass any supported Fortify Static Code Analyzer scan tuning options at the command prompt after the `-scan` option. If you use options such as `-build-label`, `-build-application`, or `-build-version`, make sure that you escape the quotes that enclose the parameter. For example:

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

If the submission succeeds, you 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.

Jobs submitted and scan results (FPR files) can be no larger than 1 GB. Before you start large scans, review "Optimizing Scan Performance" on page 77. For information about the options to use for large scans, see the *OpenText™ Fortify Static Code Analyzer User Guide*.

Offloading Both Translation and Scanning

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

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` and sets `-bf` to the `xxx.sln` file.

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

The following table provides example scan request commands for different tasks. In these examples, Fortify ScanCentral SAST is integrated with Fortify Software Security Center, email is configured for Fortify ScanCentral SAST, and Fortify Software Security Center, the Controller, and sensors are up and running.

Note: The build tool option (`-bt`) in these example commands is not required.

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

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

Start a job to scan a Maven project with a non-default build file	`scancentral -url <controller_url> start -bt mvn -bf c:\myproj\myproj-pom.xml`
Start a job to scan a JavaScript/TypeScript project	`scancentral -url <controller_url> start -bt none`
Start a job to scan a PHP version 7.1 project	`scancentral -url <controller_url> start -bt none -hv 7.1`

Task	Command
Start a job to scan an ABAP project	`scancentral -url <controller_url> start`
Start a job to scan a Ruby project	`scancentral -url <controller_url> start`
Start a job to scan a Gradle project	`scancentral -url <controller_url> start -bt gradle`
Start a job to scan a Gradle project, get email notifications from the Controller, and upload the results to Fortify Software Security Center	`scancentral -url <controller_url> start -email username@domain.com -upload -application "MyProject" -version "1.0" -uptoken <scancentralctrl_token>`

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 andopen 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 sensors that are unassigned and enabled 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:

```
scancentral –url <controller_url> start -pool <uuid>
```

Working with .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, Fortify 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 -url <controller_url> start --build-tool msbuild --build-file
<sln_file_or_path_to_sln_file>
```

```
scancentral -url <controller_url> 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 -url <controller_url> 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 -url <controller_url> start --build-tool msbuild --build-file
MySolution.sln --build-command "/t:Rebuild /p:Configuration=<build_config>"
```

Working with Go Projects

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

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

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

  `export GOPROXY=...` (Linux)

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

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

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

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

Working with Python Projects

Fortify ScanCentral SAST clients can work with Python projects in any of the following 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 72).

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

Task	Command
Start a job to scan a Python 3 project	`scancentral -url <controller_url> start --python-version 3 --python-requirements <requirements_file_path>`
Start a job to scan a Python project under an active virtual environment with dependencies already installed	`scancentral -url <controller_url> start`
Start a job to scan a Python project under an active virtual environment without project dependencies installed	`scancentral -url <controller_url> 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 -url <controller_url> start --python-virtual-env <venv_location> --python-requirements <requirements_file_path>`

Submitting a Scan Request in a Virtual Environment

If you work in a virtual environment, all of your project dependencies are already installed. You do not need to invoke the pip package manager before you start 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 -url <controller_url> start
```

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

```
scancentral -url <controller_url> 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 -url <controller_url> start --python-virtual-env <venv_
location>
```

or

```
scancentral -url <controller_url> 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 when there is no virtual environment on the client, you must have Python installed on the client. You must also specify the Python version, and the Python requirements file. 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 –url <controller_url> start --python-requirements
<requirements_file_path> --python-version <version>
```

Working with Salesforce Apex Projects

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

To scan the project using Fortify ScanCentral SAST, run the following command:

```
scancentral -url <controller_url> start -targs "-apex"
```

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

Working with 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 -url <controller_url> start -targs "-sql-language PL/SQL"
```

or

```
scancentral -url <controller_url> start -targs "-sql-language TSQL"
```

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

Working with COBOL Projects

Fortify ScanCentral SAST clients can package COBOL projects for remote translation and scanning. 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. Fortify 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 -url <controller_url> 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 -url <controller_url> 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 -url <controller_url> 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 -url <controller_url> start -targs "-copydirs MyCopyDir -
noextension-type COBOL"
```

Working with Java 8 Projects

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

Submitting Scan Requests and Uploading Results to Fortify Software Security Center

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

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

To submit a job and upload the scan results to an application version in Fortify Software Security Center:

1. Generate an authentication token to use with Fortify ScanCentral SAST by typing the following command:

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

   The following is a sample of the command output where *<token>* is the resulting authentication token value:

   ```
   Authorization Token
   <token>
   ```

2. To list the application versions to which a user account has access, open a command prompt, and then type the following command:

   ```
   fortifyclient listApplicationVersions -url <ssc_url> -authtoken <token>
   ```

The following is a sample of the command output:

```
ID      Application Name        Version
10002   Bill Payment Processor  1.1
10000   Logistics       1.3
10001   Logistics       2.5
10004   RWI     1.0
10003   Web application 1.0
```

3. Submit your job and upload your scan results to a Fortify Software Security Center application version, by typing the following command:

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

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

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

See Also

"Retrying Failed Uploads to Fortify Software Security Center" on the next page

"Start Command Options" on page 93

Specifying the 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 following example offloads the scan only and specifies a name for the FPR file for upload:

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

The following an example offloads the 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> -uptoken <token> -fprssc <my_fpr>.fpr
```

See Also

"Start Command Options" on page 93

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 -url <controller_url> upload -token <job_token>
```

where *<job_token>* corresponds to the original job that failed to upload the FPR.

See Also

"Configuring Upload to Fortify Software Security Center Retry Attempts" below

"Submitting Scan Requests and Uploading Results to Fortify Software Security Center" on page 74

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. Navigate to the *<controller_install_dir>*/tomcat/webapps/scancentral-ctrl/WEB-INF/classes directory and open the 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

"Submitting Scan Requests and Uploading Results to Fortify Software Security Center" on page 74

"Retrying Failed Uploads to Fortify Software Security Center" above

Optimizing Scan Performance

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

To optimize your scan:

1. 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. Navigate to the `<controller_install_dir>/tomcat/webapps/scancentral-ctrl` directory and open the `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 your Fortify Static Code Analyzer hardware and application configuration is set to process large FPR files. For more information, see the *OpenText™ Fortify Static Code Analyzer User Guide*.

See Also

"Configuring the Controller" on page 27

Generating a Fortify ScanCentral SAST Package

The following table provides examples of different ways to generate a package with Fortify ScanCentral SAST client.

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 .NET application.	`scancentral package -bf mySolution.sln -o myPackage.zip`

Task	Example Command
Create a package from an MSBuild project.	
Create a package from a Gradle project.	`scancentral package -o myPackage.zip`
Create a package from a Maven project with a custom pom.xml file.	`scancentral package -bf myCustomPom.xml -o myPackage.zip`
Create a package from an ABAP project.	`scancentral package -o myPackage.zip`
Create a package from an Apex project.	`scancentral package -targs "-apex" -o myPackage.zip`
Create a package from a Classic ASP project.	`scancentral package -o myPackage.zip`
Create a package from a COBOL project.	`scancentral package -targs "-copydirs copybooks" -targs "-dialect COBOL390" -o myPackage.zip`
Create a package from a ColdFusion (CFML) project.	`scancentral package -o myPackage.zip`
Create a package from a Java project.	`scancentral package -o myPackage.zip`
Create a package from a JavaScript/TypeScript project.	`scancentral package -o myPackage.zip`
Create a package from a JavaScript/TypeScript project and include the node_modules. **Caution!** Including node_modules can increase the package size as well as the scan time.	`scancentral package -snm -o myPackage.zip`
Generate a package from an Android project in Kotlin that uses the Android plugin.	`scancentral package -bt gradle -o myPackage.zip`

Task	Example Command
Create a package from a Go project.	`scancentral package -o myPackage.zip`
Create a package for only IaC/Dockerfiles. **Note:** If Dockerfiles are included in a Gradle, Maven, or MSBuild project, then the Docker files will be automatically included in the package.	`scancentral package -bt none -o myPackage.zip`
Create a package from a PHP 7.1 project.	`scancentral package -hv 7.1 -o myPackage.zip`
Create a package from a Python 2 project.	`scancentral package -yv 2 -pyr <requirements_file_path> -o myPackage.zip`
Create a package from a Python project under an active virtual environment with dependencies already installed.	`scancentral package -o myPackage.zip`
Create a package from a Python project under an active virtual environment without project dependencies installed.	`scancentral package -pyr <requirements_file_path> -o myPackage.zip`
Create a package from a Python project using an existing Python virtual environment and install project dependencies.	`scancentral package -pyv <venv_location> -pyr <requirements_file_path> -o myPackage.zip`
Create a package from a Ruby project.	`scancentral package -o myPackage.zip`

Create a package from a SQL project.	`scancentral package -targs "-sql-language TSQL" -o myPackage.zip`
	
	`scancentral package -targs "-sql-language PL/SQL" -o myPackage.zip`

Create a package from a Visual Basic 6.0 project.	`scancentral package -o myPackage.zip`

See Also

"Package Command Options" on page 100

"Using the PackageScanner Tool" below

Using the PackageScanner Tool

If you have Fortify Static Code Analyzer installed locally, you can run an analysis of a package locally, without sending it to the Controller. The PackageScanner tool takes a package created using the Fortify ScanCentral SAST package command, generates Fortify Static Code Analyzer commands, and then scans it using a locally-installed Fortify Static Code Analyzer. The packagescanner tool is located in the *<sca_install_dir>*/bin directory. The following table describes the PackageScanner tool command-line options.

Packagescanner Option	Description
-b, --build-id *<id>*	(Optional) Specifies the build ID. Fortify Static Code Analyzer uses the build ID to track which files are compiled and combined as part of a build, and later, to scan those files. If you do not specify a build ID, Fortify ScanCentral SAST automatically generates one.
--debug	(Optional) Enables debug logging for Fortify ScanCentral SAST clients and sensors.
--fpr *<file>*.fpr	(Required) Specifies the FPR file to which analysis results are written.
--package *<package_file>*.zip	(Required) Specifies the path to the package file generated by Fortify ScanCentral SAST with the package command.
-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-path *<sourceanalyzer_exe_path>*	(Optional if started from Fortify Static Code Analyzer) Specifies the path to the Fortify Static Code Analyzer executable. If Fortify ScanCentral SAST is part of the Fortify Static Code Analyzer

Packagescanner Option	Description
	installation, the path is determined automatically.
`--sca-scan-log <log_file_path>`	(Optional) Specifies a log file for scan commands. By default, the log file is created in a temporary directory, which is removed after program execution.
`--sca-translation-log <log_file_path>`	(Optional) Specifies a log file for translation commands. By default, the log file is created in a temporary directory, which is removed after program execution.
`-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.
`-v`, `--version`	(Optional) Displays the PackageScanner tool version.
`--working-dir <dir>`	(Optional) Specifies a directory where the 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 program execution (unless the `-debug` option is specified).

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 Fortify ScanCentral SAST Package" on page 77

Chapter 7: Managing Scan Requests and Scan Results

This section describes how to view the status of your scan requests, retrieve the scan results, cancel scan requests, work with Fortify ScanCentral SAST from Fortify Software Security Center, and other related tasks.

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 scan request status from the Fortify Software Security Center user interface. 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.
	FAULTED	Scan failed as the result of an unexpected 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	Scan results (FPR) upload is awaiting upload to Fortify Software Security Center.
	CANCELED	Scan results (FPR) upload to Fortify Software Security Center was canceled or failed.
	FAILED	Scan results (FPR) upload to Fortify Software Security Center failed.
	COMPLETED	Scan results (FPR) file was uploaded to Fortify Software Security Center successfully.

For information about how this status information is represented in Fortify Software Security Center, see the *OpenText™ Fortify Software Security Center User Guide*.

See Also

"Status Command Options" on page 92

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 Options" on page 99

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 Options" on page 100

Working with Fortify ScanCentral SAST from Fortify Software Security Center

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

- The Fortify Software Security Center user interface includes a ScanCentral SAST view where you can see the status of recent scan requests.

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

- The Controller can upload scan results directly to Fortify Software Security Center application versions.
- You can create and manage Fortify ScanCentral SAST sensor pools from Fortify Software Security Center.

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

See Also

"Configuring the Connection to Fortify Software Security Center" below

Configuring the Connection to Fortify Software Security Center

You can monitor Fortify ScanCentral SAST and display its results in Fortify Software Security Center. You can also create and manage ScanCentral SAST sensor pools. For instructions on how to configure ScanCentral SAST with Fortify Software Security Center, see the *OpenText™ Fortify Software Security Center User Guide*.

See Also

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

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 the next page.
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 http://servername:8080/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.

Go to the *<controller_install_dir>*`/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory and check the `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.

Preserving the Fortify Static Code Analyzer Project Root Directory

By default, the ScanCentral SAST sensor creates a temporary working directory to unpack the 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. Navigate to 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 47

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. Navigate to `<controller_install_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes`.
2. Open the `log4j2.xml` file in a text editor.
3. 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">`
4. 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">
   ```

5. 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 87

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 (start, retrieve, and so on).

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 87

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
- Metadata file from the remote translation file

Appendix A: Fortify ScanCentral SAST Command-Line Options

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

This section contains the following topics:

Global Options

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

Global Option	Description
`-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 88.
`-h`, `--help <command>`	Displays help for the selected command. To see all command help, type `-h all`.
`-ssctoken <token>`	Specifies a Fortify Software Security Center authentication

Global Option	Description
	token of type ScanCentralCtrlToken. See the *OpenText™ Fortify Software Security Center User Guide* for descriptions of other tokens that you can use to connect with Fortify Software Security Center.
`-sscurl <url>`	Specifies a Fortify Software Security Center server URL.
`-url <url>`	Specifies a Fortify ScanCentral SAST Controller URL.
`-version`	Displays the Fortify ScanCentral SAST version.

Status Command Options

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

Status Option	Description
`-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. 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. If `0` is specified, no timeout is set.
`-ctrl`	Checks whether the Controller is running.
`-pi`, `--poll-interval <n>`	Specifies how frequently (in seconds) to poll the processing status. The valid range for *<n>* is from `10` to `60`.

Status Option	Description
`-token`, `--job-token <token>`	Specifies the job token to query.

See Also

"Viewing the Scan Request Status" on page 83

Start Command Options

You can use the options listed in this section with the start command to perform a remote scan, or to perform a remote translation and scan.

Start Option	Description
`-application <name>`	Specifies the Fortify Software Security Center application name.
`-b`, `--build-id <id>`	Specifies the build ID of the session to export.
`-bc`, `--build-command <commands>`	(For use with Maven, Gradle dotnet, and MSBuild) Specifies custom build parameters for preparing and building a project. For example, to invoke a Gradle build before packaging: `-Prelease=true clean customTask build` If you use the `-bc` option, and the build fails, ScanCentral 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 your results might be incomplete.
`-bf`, `--build-file <file>`	Specifies the build file, unless it has a default name such as `build.gradle` or `pom.xml`. You cannot use this option with the `-scan` option.
`-block`	Waits for the job to complete, and then downloads the scan results.

Start Option	Description
`-bt`, `--build-tool <name>`	(Optional) Specifies the build tool name used for the project. The valid values for *<name>* are `dotnet`, `gradle`, `msbuild`, `mvn`, or `none`. Example: `-bt mvn -bc "package --setting custom.xml"` You cannot use this option with the `-scan` option. The `-bt` option is *not* required. Fortify ScanCentral SAST can automatically detect the build tool based on the project files being scanned.
`-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 details, see "Creating a Log Archive for Customer Support" on page 90.
`-email <address>`	Specifies the email address for job status notifications.
`-exclude <file>`	Specifies a file or directory (with absolute or relative path, or Ant-style path pattern) to exclude from a package (repeatable).
`-f`, `--output-file <file>`	Specifies the name for the local FPR file output. Use with the `-block` option to specify the name for the local FPR file output after a scan is completed.
`-filter <file>`	Specifies a filter file to use during a scan (repeatable).
`-fprssc`, `--fpr-filename-on-ssc <file>`	Specifies the name to use for the FPR files uploaded to Fortify Software Security Center. The file name must not exceed 128 characters in length and *must not* contain the following characters: • colon (:) • backslash (\) • forward slash (/) • asterisk (*)

Start Option	Description
	• question mark (?) • vertical bar or pipe (|) • less than (<) • greater than (>) • double quote (")
`-hv`, `--php-version <version>`	Specifies the PHP version.
`-log`, `--log-file <file>`	Specifies a file name for the local log file after the scan is complete.
`-mbs <file>`	Specifies a mobile build session file to upload.
`-o`, `--overwrite`	Overwrites the existing FPR or log with new data.
`-p`, `--package <file>`	Specifies the project package file to upload.
`-pi`, `--poll-interval <n>`	Specifies how frequently (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.
`-projroot`, `--project-root <dir>`	Specifies the project directory for the mobile build session export.
`-projtl`, `--project-template <file>`	Specifies an issue template file to include.
`-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.
`-q`, `--quiet`	Prevents the printing to stdout from the build execution.
`-rules <file/dir>`	Specifies a custom rules file or directory to use during the scan (repeatable).

Start Option	Description
`-sargs`, `--scan-args <scan_option>`	Specifies a Fortify Static Code Analyzer scan options (repeatable). Takes a single string option. 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 104. **Note:** You cannot use the `-sargs` option with the `-scan` option. It is for use in remote translation and scan only.
`-scan`	Sets the point beyond which all options are for Fortify Static Code Analyzer. You cannot use this option with the `--build-tool` or `--package` option.
`-skipBuild`	Disables the project preparation build step before packaging. If you use `-skipBuild` option, any `-bc` option specified is ignored. **Note:** You can apply this option to Gradle and Maven build tools, but not to MSBuild.
`-snm`, `--scan-node-modules`	Specifies node_modules dependencies in the package. If you set `--scan-node-modules`, all third-party library scan results are added to the resulting FPR. **Tip:** Because including node_modules dependencies in a package does not improve type resolution or dataflow, and can result in an excessive number of false positives, Fortify recommends that you exclude them from scans. By default, node_modules dependencies are not included in a package unless you apply the `--scan-node-modules` option from the command line.
`-sp`, `--save-package <file>`	Specifies the package file to save after uploading. The file extension must be `*.zip`.

Start Option	Description
`-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.
`-t`, `--include-test`	Includes test source set (Gradle) or test scope (Maven) to scan (for Java projects only).

`-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 of the Fortify Static Code Analyzer options you can use with the `-targs` option, see "Options Accepted for -targs (--translation-args)" on page 103.
	
	If you use the `-targs` option with the `--package` option, Fortify ScanCentral SAST ignores it and displays an error message.
	
	Note: You cannot use the `-targs` option with the `-scan` option. The `-targs` option is for use in remote translation and scan only.

`-upload`, `--upload-to-ssc`	Uploads the FPR to Fortify Software Security Center after completion.

Start Option	Description
`-uptoken`, `--ssc-upload-token <token>`	Specifies the Fortify Software Security Center file upload authentication token. If the `pool_mapping_mode` property is set to `DISABLED` on the Controller, you can use a Fortify Software Security Center of type AnalysisUploadToken. However, if the `pool_mapping_mode` is set to ENABLED, a token of type ScanCentralCtrlToken is required. For information about how to acquire authentication tokens, see the *OpenText™ Fortify Software Security Center User Guide*.
`-version`, `--application-version <name>`	Specifies the Fortify Software Security Center application version name.
`-versionid`, `--application-version-id <id>`	Specifies the Fortify Software Security Center application version ID.
`-yv`, `--python-version <version>`	Specifies the Python version to automatically find the installed Python. Allowed values: 2 or 3. This option is ignored if the Fortify ScanCentral SAST client is started under a Python virtual environment or if `-python-virtual-env` is specified.

Upload Command Options

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

Upload Option	Description
`-token`, `--job-token <token>`	Specifies the job token to use to resend an FPR file to Fortify Software Security Center.

Retrieve Command Options

Use the `retrieve` command to download the result of a remote scan job from the Fortify ScanCentral SAST Controller.

Retrieve Option	Description
`-block`	Waits for the job to complete and then downloads 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. If `0` is specified, no timeout is set. The default value is `0`.
`-f`, `--output-file <file>`	Specifies a file name for the local scan results (FPR) file after the scan is complete.
`-log`, `--log-file <file>`	Specifies a file name for the local log file after the scan is complete.
`-o`, `--overwrite`	Overwrites an existing FPR or log file with new data.
`-pi`, `--poll-interval <n>`	Specifies how frequently (in seconds) to poll the processing status. The valid range for *<n>* is `10` to `60` seconds.
`-slog`, `--sensor-log-file <file>`	Specifies the file name for local sensor log output.
`-token`, `--job-token <token>`	Specifies the job token to query.

See Also

"Retrieving Scan Results from the Controller" on page 84

Cancel Command Options

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

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

See Also

"Canceling Scan Requests" on page 85

Worker Command Options

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

Worker Option	Description
`-hello`	Sensor reporting for duty.
`-pool`, `--assign-to-pool <uuid>`	Specifies the sensor pool to which the sensor is to be assigned after it connects to the Controller. If the sensor is already assigned to a pool, this option overrides that assignment. If an error occurs In sensor pool assignment, the sensor shuts down.
`-sto`, `--scan-timeout <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.

Package Command Options

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

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

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

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. For example, to invoke a Gradle build before packaging: `-Prelease=true clean customTask build` If you use the `-bc` option, and the build fails, ScanCentral 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`.
`-bt`, `--build-tool` *<name>*	Specifies the name of the build tool used for the project. You cannot use this option with the `-scan` option. The valid values for *<name>* are `dotnet`, `gradle`, `msbuild`, `mvn`, and `none`.
`-exclude <file>`	Specifies a file or directory (with absolute or relative path, or Ant-style path pattern) to exclude from a package (repeatable).
`-hv`, `--php-version` *<version>*	Specifies the PHP version.
`-o`, `--output` *<file>*	Specifies the output file name. The file extension must be `*.zip`.
`-oss`, `--open-source-scan`	(For use with Fortify on Demand only) Specifies to generate and collect additional files for open source scanning. For details, see the *OpenText™ Fortify on Demand User Guide*.
`-pyr`, `--python-requirements` *<file>*	Specifies the Python project requirements file to install and collect dependencies.

Package Option	Description
`-pyv`, `--python-virtual-env <dir>`	Specifies the Python virtual environment location.
`-q`, `--quiet`	Prevents the printing of stdout from the build execution.
`-skipBuild`	Disables the project preparation build step before packaging.
`-snm`, `--scan-node-modules`	Specifies to include node_modules dependencies in the package. If you set this option, all third-party library scan results are added to the resulting FPR. **Tip:** Because including node_modules dependencies in a package does not improve type resolution or dataflow results, and because they degrade translation and scan speed, Fortify recommends that you exclude them from scans. By default, node_modules are not included in a package unless you apply the `--scan-node-modules` option from the command line.
`-t`, `--include-test`	(Java projects only) Includes the test source set (Gradle) or test scope (Maven) to scan
`-targs`, `--translation-args <option>`	Specifies a Fortify Static Code Analyzer translation option (repeatable) Takes a single string option. 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.
`-yv`, `--python-version <version>`	Specifies the Python version to automatically find the installed Python. Allowed values: 2 or 3. This flag is ignored if the Fortify ScanCentral SAST client is started under a Python virtual environment or if `-python-virtual-env` is specified.

See Also

"Generating a Fortify ScanCentral SAST Package" on page 77

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 46.

Update Command

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

Examples:

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

or

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

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

The following table 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 a description of the Fortify Static Code Analyzer translation options, see the *OpenText™ Fortify Static Code Analyzer User Guide*.

-autoheap	-dialect	-php-version
-abap-includes	-disable-language	-project-root
-apex	-django-disable-autodiscover	-python-no-auto-root-calculation
-appserver	-django-template-dirs	-python-path
-appserver-home	-enable-language	-python-version
-appserver-version	-encoding	-quiet
-build-label	-exclude	

-build-project	-extdirs	-ruby-path
-build-version	-gopath	-rubygem-path
-checker-directives	-goproxy	-show-unresolved-symbols
-copydirs	-goroot	-source-base-dir
-cp	-jdk, -source	-sourcepath
-debug	-jvm-default	-sql-language
-debug-mem	-noextension-type	-v, -version
-debug-verbose	-php-source-root	-verbose

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

The following table 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 a description of the Fortify Static Code Analyzer translation options, see the *OpenText™ Fortify Static Code Analyzer User Guide*.

-analyzers	-disable-default-rule-type	-project-root
-autoheap	-enable-analyzer	-project-template
-build-label	-filter	-quick
-build-project	-legacy-jsp-dataflow	-quiet
-build-version	-no-default-issue-rules	-rules
-debug	-no-default-rules	-sc, scan-policy
-debug-mem	-no-default-sink-rules	-v, -version
-debug-verbose	-no-default-source-rules	-verbose
-disable-analyzer	-p , -scan-precision	

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 23.2.0)

Just add your feedback to the email and click send.

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

We appreciate your feedback!