OpenText™ Application Security Software System Requirements

System Requirements for OpenText™ Application Security Software

Version 26.4

PDF Generated on November 11, 2026

© Copyright 2026 Open Text

Table of Contents

System Requirements for OpenText™ Application Security Software 4
1 Introduction 4
1.1 Software delivery 4
1.2 Software licenses 4
2 OpenText ScanCentral DAST system requirements 4
2.1 Architectural best practices 4
2.2 Configuration tool CLI 4
2.3 Database requirements 5
2.4 Core components VM 6
2.5 OpenText DAST sensor 6
2.6 Fortify Connect client 7
2.7 Ports and protocols 8
2.8 Browsers 10
2.9 Event-based Web Macro Recorder (standalone) 10
2.10 Software integrations 11
3 OpenText™ ScanCentral SAST system requirements 11
3.1 Controller requirements 11
3.1.1 Controller hardware requirements 11
3.1.2 Controller platforms and architectures 12
3.1.3 Controller application server 12
3.1.4 Interoperability with Application Security 12
3.2 Sensor requirements 13
3.2.1 Sensor hardware requirements 13
3.2.2 Sensor software requirements 13
3.3 Client requirements 14
3.3.1 Client hardware requirements 14
3.3.2 Client software requirements 14
3.3.3 Languages supported for remote translation 15
3.3.4 Build tools supported for remote translation 17
3.4 Supported sensor versions 17
4 OpenText Application Security (Software Security Center) server system requirements 17
4.1 Hardware requirements 18
4.2 Supported platforms and architectures 18
4.3 Supported application server 19
4.4 Database requirements 19
4.5 Kubernetes cluster deployment requirements (optional) 20
4.6 Browsers 21
4.7 Supported authentication systems 21
4.8 BIRT report requirements 22
4.9 Supported service integrations 22
5 OpenText Application Security Tools system requirements 22
5.1 Hardware requirements 23

5.2 Platforms and architectures 23
5.3 Software requirements 23
5.4 Service integrations for Application Security Tools 24
5.5 Secure Code Plugins 24
5.6 Authentication for connecting to Application Security 25
5.7 BIRT reports 25
6 Fortify Project Results (FPR) file compatibility 26
7 Virtual Machine support 26
8 Technologies no longer supported in this release 27
9 Technologies to lose support in the next release 27
10 Acquiring OpenText Application Security Software 27
10.1 Verifying software downloads 30
11 Assistive technologies (Section 508) 31

System Requirements for OpenText™ Application Security Software

Software Version: 26.4.0

Document Release Date: November 2026

Software Release Date: November 2026

1. Introduction

This document describes the environments and products that OpenText supports for this version of Application Security Software, which includes:

- OpenText™ ScanCentral DAST
- OpenText™ ScanCentral SAST
- OpenText™ Application Security Center Server

1.1. Software delivery

OpenText Application Security Software is delivered electronically. See Acquiring Software for more information.

1.2. Software licenses

OpenText Application Security Software products require a license. For OpenText ScanCentral DAST, Fortify SAST, OpenText DAST, and Fortify WebInspect Enterprise, you will receive an email with instructions for how to activate your product.

For all other OpenText Application Security Software products described in this document (including Fortify SAST and Secure Code Plugins), you must download the Fortify license file for your product from the Software Licenses and Downloads (SLD) portal (https://sld.microfocus.com). Use the credentials that Customer Support has provided for access.



Note

Using Fortify License and Infrastructure Manager (LIM) to manage concurrent licenses for Fortify SAST requires LIM version 21.2.0 or later.

2. OpenText ScanCentral DAST system requirements

Before you install OpenText ScanCentral DAST, make sure that your system meets the requirements described in this section. OpenText does not support beta or pre-release versions of operating systems, service packs, or required third-party components.

2.1. Architectural best practices

The OpenText ScanCentral DAST core components and OpenText DAST sensor are available as Docker® images only.

Follow these best practice guidelines when you configure OpenText ScanCentral DAST:

- Run the DAST API, DAST Global Service, DAST Utility Service, WebInspect API Core, and Fortify License and Infrastructure Manager (LIM) Docker containers on the same VM or on separate VMs.
- Do not run the OpenText DAST sensor on the same VM as any of the other DAST components.

For more information about the OpenText ScanCentral DAST components, see What is OpenText ScanCentral DAST? and OpenText ScanCentral DAST with two-factor authentication.

2.2. Configuration tool CLI

This topic describes the software and hardware requirements for the machine on which the ScanCentral DAST configuration tool CLI runs to configure settings for the OpenText ScanCentral DAST components.

Software requirements

The ScanCentral DAST Configuration Tool CLI runs on and works with the software packages listed in the following table.

Package	Versions	Notes
Windows	Windows 11	
	Windows Server 2019	
.NET Platform	.NET SDK Core Runtime 9.0	
Red Hat® Enterprise Linux® (RHEL)	9.x x86_64	

Hardware requirements

OpenText recommends that you use the ScanCentral DAST Configuration Tool CLI on a system that conforms to the supported components listed in the following table.

Component	Requirement	Notes
RAM	2+ GB	Recommended
	1 GB	Minimum

2.3. Database requirements

OpenText ScanCentral DAST supports the databases listed in the following table.

Package	Versions	Notes
Microsoft® SQL Server® (English-language version only)	SQL Server 2022	Recommended No scan database limit; SQL Server must use Mixed Mode.
	SQL Server 2019	No scan database limit; SQL Server must use Mixed Mode.
	Azure SQL Server	Using Azure SQL Server outside the Azure infrastructure might cause poor performance for OpenText ScanCentral DAST. OpenText recommends using Azure SQL Server with OpenText ScanCentral DAST inside the Azure infrastructure only.
	Amazon RDS for SQL Server	
PostgreSQL®	PostgreSQL 15 or later	
	Azure PostgreSQL	
	Amazon RDS for PostgreSQL	

Database recommendations

OpenText recommends that you configure the database server on a separate machine from either Application Security or any other OpenText ScanCentral DAST components.

The ScanCentral DAST SQL database requires case-insensitive collation.

Important

This is opposite the requirement for Application Security databases.

Important recommendation about disk I/O

Disk I/O encompasses the input/output operations on a physical disk. If you are reading data from a file, the processor must wait for the file to be read (the same applies to writing data to a file). OpenText ScanCentral DAST is a high I/O-intensive application, which affects performance. Make sure that your disk subsystem provides low read/write latency. OpenText recommends that you monitor disk I/O as the database grows.

2.4. Core components VM

This topic describes the hardware and software requirements to run the ScanCentral DAST API, Global Service, and Utility Service containers.

Software requirements

The ScanCentral DAST API, Global Service, and Utility Service containers run on and work with the software packages listed in the following table.

Software	Versions	Notes
Docker Enterprise	18.09 or later	
Red Hat Enterprise Linux (RHEL)	9.x x86_64	

Follow Docker recommendations for the Docker engine version to use for these versions of Windows and Red Hat images.

Important

Podman, Rancher, OpenShift, and Nerdctl container management platforms are not currently supported. OpenText ScanCentral DAST execution and performance may differ on these platforms than on the Docker platform. However, OpenText will provide limited support in addressing any defects in the product that are not directly related to these platforms.

Hardware requirements

OpenText recommends that you use the ScanCentral DAST API, Global Service, and Utility Service containers on a system that conforms to the supported components listed in the following table.

Component	Requirement	Notes
RAM	32 GB	
Processor	8 Core	

2.5. OpenText DAST sensor

The OpenText DAST sensor is available in Microsoft Windows® or Linux® Docker® versions.

About the versions

The Linux® version includes the following images:

- dast-scanner: 26.2.ubi.9
- wise: 26.2.ubi.9 - WebInspect script engine (WISE) for JavaScript execution and Web Macro Recorder macro playbacks
- fortify-2fa: 26.2.ubi.9 - 2FA server to synchronize two-factor authentication requests (used only if the scan is configured to playback a two-factor authentication login macro)
- Database for scan data
- scancentral-dast-scannerservice: 26.2.ubi.9

Software requirements

The OpenText DAST sensor has the software requirements listed in the following table.

Package	Versions	Notes
Docker Enterprise	18.09 or later	
Red Hat® Universal Base Image (UBI)	9.x x86_64	The Linux® version supports conducting scans of gRPC APIs.

Follow Docker recommendations for the Docker engine version to use for these versions of Microsoft Windows® and Red Hat® images.

About the database

The OpenText DAST images support both PostgreSQL and SQL Server Express. There is a 10 GB scan database limit for SQL Server Express.

Hardware requirements

OpenText recommends that you install OpenText DAST on Docker on a host that conforms to the supported components listed in the following table and configure the container to use these resources. OpenText does not support beta or pre-release versions of operating systems, service packs, and required third-party components.

Component	Requirement	Notes
Processor	2.5 GHz quad-core or faster	Complex applications might benefit from additional cores.
RAM	16 GB	Complex applications might benefit from additional memory. OpenText recommends 32 GB of memory to scan with single-page application (SPA) support.
Hard disk	40 GB	Using SQL Express and storing scans locally requires additional disk space per scan.

2.6. Fortify Connect client

The Fortify Connect client executable runs on and works with the software packages listed in the following table.

Software	Versions	Notes
ASP.NET Core Runtime	9.0	
OpenSSH Client	7.6 or later	

2.7. Ports and protocols

This section describes the ports and protocols that the ScanCentral DAST components use to make required and optional connections.

ScanCentral DAST API required connections

The following table lists the ports and protocols that the ScanCentral DAST API container uses for required connections.

Endpoint	Port	Protocol	Notes
Application Security ScanCentral DAST Global Service ScanCentral DAST Sensor Service	80	HTTP	If SSL is not configured, the port on the host running the container is forwarded to port 80 on the container. Host port mapping is customizable to the container port.
Application Security ScanCentral DAST Global Service ScanCentral DAST Sensor Service	443	HTTPS	If SSL is configured, the port on the host running the container is forwarded to port 443 on the container. Host port mapping is customizable to container port.
SQL Server, Azure SQL Server, or Amazon RDS for SQL Server	1433	TCP	This is the default SQL Server port.
PostgreSQL, Azure PostgreSQL, or Amazon RDS for PostgreSQL	5432	TCP	This is the default PostgreSQL port.

ScanCentral DAST Global Service required connections

The ScanCentral DAST Global Service does not expose any ports.

The following table lists the ports and protocols that the ScanCentral DAST Global Service container uses for required connections.

Endpoint	Port	Protocol	Notes
SQL Server, Azure SQL Server, or Amazon RDS for SQL Server	1433	TCP	This is the default SQL Server port.
PostgreSQL, Azure PostgreSQL, or Amazon RDS for PostgreSQL	5432	TCP	This is the default PostgreSQL port.

ScanCentral DAST sensor required connections

The ScanCentral DAST sensor does not expose any ports.

The ScanCentral DAST sensor communicates with the ScanCentral DAST API over the port that is exposed on the host running the ScanCentral DAST API container.

ScanCentral DAST Utility Service required connections

The following table lists the ports and protocols that the ScanCentral DAST Utility Service container uses for required connections.

Endpoint	Port	Protocol	Notes
ScanCentral DAST API	5000	HTTP	If SSL is not configured, the port on the host running the container is forwarded to port 5000 on the container. Host port mapping is customizable to the container port.
ScanCentral DAST API	5001	HTTPS	If SSL is configured, the port on the host running the container is forwarded to port 5001 on the container. Host port mapping is customizable to container port.
SQL Server, Azure SQL Server, or Amazon RDS for SQL Server	1433	TCP	This is the default SQL Server port.
PostgreSQL, Azure PostgreSQL, or Amazon RDS for PostgreSQL	5432	TCP	This is the default PostgreSQL port.

Fortify Connect server required connections

The ScanCentral DAST API, Global Service, Utility Service, and Scanner Service (when running in remote mode) access the internal host and internal port that are specified in the "FortifyConnectServerSettings" when configuring your OpenText ScanCentral DAST environment.

Kafka required connections

If Apache® Kafka® is configured, Application Security publishes messages to Kafka using the list of servers in the `stream.kafka.bootstrapServers` specified in the `app.properties` for Application Security. OpenText ScanCentral DAST consumes messages from Kafka using the list of servers listed in the `SSCSettings.KafkaSettings.BootstrapServers` in the settings file used to configure OpenText ScanCentral DAST.

2.8. Browsers

OpenText recommends that you use one of the browsers listed in the following table and a screen resolution of 1400 x 800.

Browser	Version
Google Chrome™	116 or later
Microsoft® Edge	114 or later
Mozilla® Firefox®	116 or later
Apple® Safari	14 or later

2.9. Event-based Web Macro Recorder (standalone)

By default, the Event-based Web Macro Recorder is installed as part of the OpenText DAST toolkit when OpenText DAST is installed on Windows. However, OpenText ScanCentral DAST allows you to download a Microsoft Windows® or Mac® version of the Event-based Web Macro Recorder and install it as a standalone tool.

Hardware requirements

OpenText recommends that you install the standalone Event-based Web Macro Recorder on a system that conforms to the supported components listed in the following table.

Component	Requirement	Notes
Processor	Intel x86	Microsoft Windows®
	Apple® silicon	macOS®
RAM	16 GB	
Hard disk	1 TB	

Windows version software requirements

The Microsoft Windows® version of the Event-based Web Macro Recorder tool runs on and works with the software packages listed in the following table.

Package	Version	Notes
Windows	Windows 11	
	Windows Server 2019	

macOS version software requirements

The macOS® version of the Event-based Web Macro Recorder tool runs on and works with the software packages listed in the following table.

Package	Versions	Notes
Operating System	macOS 14.6	

Running as administrator

The standalone Web Macro Recorder tool requires administrative privileges for proper operation of all features. See the Microsoft Windows® or macOS® documentation for instructions on changing the privilege level to run the Web Macro Recorder tool as an administrator.

2.10. Software integrations

The following table lists products that you can integrate with OpenText ScanCentral DAST.

Product	Versions
Application Security	26.2.0
Kubernetes on Azure	1.19 or later

3. OpenText™ ScanCentral SAST system requirements

This section describes the system requirements for each major component: Controller, sensor, and client.

This section contains the following topics:

- Controller requirements
- Sensor requirements
- Client requirements
- Supported sensor versions

3.1. Controller requirements

This section describes the hardware, platform, and other requirements for the ScanCentral SAST Controller.

This section contains the following topics:

- Controller hardware requirements
- Controller platforms and architectures
- Controller application server
- Interoperability with Application Security

3.1.1. Controller hardware requirements

OpenText recommends that you install the ScanCentral SAST Controller on a high-end 64-bit processor running at 2 GHz with at least 8 GB of RAM.

To estimate the amount of disk space required on the machine that runs the Controller, use one of the following equations:

Intended use	Equation
Remote scan only	*<num_jobs_per_day>* x (*<size_avg_MBS>* + *<size_avg_FPR>* + *<size_avg_SCA_log>*) x *<num_days_data_is_persisted>*
Remote translation and scan	*<num_jobs_per_day>* x (*<size_avg_archived_project_with_dependencies>* + *<size_avg_FPR>* + *<size_avg_SCA_log>*) x *<num_days_data_is_persisted>*

By default, data is persisted for seven days.

3.1.2. Controller platforms and architectures

The ScanCentral SAST Controller supports the platforms and architectures listed in the following table.

Operating system	Versions
Windows	Server 2022 Server 2025
Linux	Red Hat Enterprise Linux 8, 9 SUSE® Linux® Enterprise Server 15

3.1.3. Controller application server

The ScanCentral SAST Controller installation includes the supported Apache® Tomcat™ version 11.0.x that runs on JRE 17 or later.

Important

OpenText recommends using the Tomcat version that is originally shipped. Other versions of Tomcat are not supported.

Note

Use the connector attribute `maxPartCount` to help prevent DoS attacks. By default, it is set to 10. For more information, refer to Tomcat 10.1 documentation.

If you want to keep the latest release of Tomcat 10.1, add the following attribute to the relevant `<Connector>` element in the `server.xml` file:

`maxPartCount=-1`

For example:

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11Nio2Protocol"
               maxThreads="150" SSLEnabled="true" scheme="https" maxPartCount=-1
               secure="true" defaultSSLHostConfigName="<SSLhostconfig>">
```

3.1.4. Interoperability with Application Security

OpenText supports integrating the ScanCentral SAST Controller with an Application Security version that is the same or one version earlier than the Controller version. For example, the 26.2.0 version of the Controller works with the 25.4.0 or 26.2.0

versions of Application Security.

3.2. Sensor requirements

This section describes the hardware and software requirements for ScanCentral SAST sensors.

This section contains the following topics:

- Sensor hardware requirements
- Sensor software requirements

3.2.1. Sensor hardware requirements

ScanCentral SAST sensors are installed on build machines that run OpenText SAST (Fortify Static Code Analyzer). For Fortify SAST hardware requirements, see the *OpenText™ Static Application Security Testing User Guide*.

Sensor disk space requirements

To estimate the amount of disk space required on the machine that runs a ScanCentral SAST sensor, use one of the following equations:

Intended use	Equation
Remote scan only	*<num_of_scans>* x (*<size_avg_MBS>* + *<size_avg_FPR>* + *<size_avg_SCA_log>*) x *<num_days_data_is_persisted>*
Remote translation and scan	*<num_jobs_per_day>* x (*<size_avg_archived_project_with_dependencies>* + *<size_avg_project_with_dependencies>* + *<size_avg_FPR>* + *<size_avg_SCA_log>*) x *<number_days_data_is_persisted>*

By default, data is persisted for seven days.

3.2.2. Sensor software requirements

ScanCentral SAST sensors are installed on build machines that run OpenText SAST (Fortify Static Code Analyzer).

ScanCentral SAST sensors run on the supported platforms and architectures listed in the following table.

Operating system	Platforms	Distributions and versions
Windows	x64	Windows 10, 11 Windows Server 2022, 2025
Linux	x64 ARM	CentOS Linux 7.x (7.6 or later) Red Hat Enterprise Linux 7.x (7.2 or later), 8.x (8.2 or later), 9.x 15 Ubuntu 20.04.1 LTS, 22.04.1 LTS, 24.04 LTS

The following table lists software requirements for remote translation of specific project types.

Technology	Software	Operating systems
.NET (Web projects)	.NET Framework 4.8 or later (MSBuild only) .NET SDK 10.0	Windows
.NET	.NET SDK 10.0	Windows, Linux

3.3. Client requirements

This section describes the hardware and software requirements for the clients as well as languages and build tools supported for remote translation and scan.

This section contains the following topics:

- Client hardware requirements
- Client software requirements
- Languages supported for remote translation
- Build tools supported for remote translation

3.3.1. Client hardware requirements

ScanCentral SAST clients run on the Windows and Linux systems that Fortify SAST supports. You can install the ScanCentral SAST embedded client with the Fortify SAST installation or you can install the ScanCentral SAST standalone client separately. Both are the same client. OpenText recommends that you install ScanCentral SAST standalone clients on a system with at least a two-core processor and 8 GB of RAM.

The project package ScanCentral SAST client produces is roughly the size of the project and any dependencies.

3.3.2. Client software requirements

ScanCentral SAST embedded clients are installed on build machines that run Fortify SAST.

Standalone clients require Java 21 or later.

ScanCentral SAST clients run on the supported platforms and architectures listed in the following table.

Operating system	Platforms	Distributions and versions
Windows	x64	Windows 10, 11 Windows Server 2022, 2025
Linux	x64 ARM	CentOS Linux 7.x (7.6 or later) Red Hat Enterprise Linux 7.x (7.2 or later), 8.x (8.2 or later), 9.x 15 Ubuntu 20.04.1 LTS, 22.04.1 LTS, 24.04 LTS

Packaging of specific project types requires the software listed in the following table.

Language	Software	Operating systems
.NET	.NET Framework 4.8 or later (MSBuild only)	Windows
	.NET SDK 10.0	Windows, Linux
ABAP/BSP	Fortify ABAP Extractor is supported on a system running SAP® release 7.02, SP level 0006.	Windows, Linux
COBOL	Microsoft Visual C++ 2017 Redistributable (x86) **Note** This is not a requirement for legacy COBOL analysis.	Windows

3.3.3. Languages supported for remote translation

ScanCentral SAST clients support generating packages with sources and dependencies for remote translation on sensors for the languages listed in the following table. Clients use the package managers listed below to restore dependencies prior to packaging. For the language-specific software requirements, see Client software requirements.

Language	Package manager
.NET applications in C# and Visual Basic (VB.NET) (.NET Core, .NET Standard, ASP.NET)	NuGet
ABAP®	
Apex	
Classic ASP	
COBOL	
ColdFusion	
Dockerfiles	
Go	
HTML	
Java	
JavaScript, TypeScript	npm, pnpm, Yarn
JSON	
JSP	
Kotlin	
PHP	Composer
PL/SQL, T-SQL	
Python	pip
Ruby	Gem **Note** The client uses Gem only to obtain the gem paths for packaging.
VBScript	
Visual Basic 6.0	

Language	Package manager
XML	

3.3.4. Build tools supported for remote translation

ScanCentral SAST clients support the build tools listed in the following table for remote translation.

Build tool	Versions
dotnet	6.0–10.x
Gradle	5.0–8.13
Apache Maven™ Software	3.8.x, 3.9.x
MSBuild	14.x–17.14

3.4. Supported sensor versions

The ScanCentral SAST client, by default, sends scan requests to the same version of the SAST sensor for remote translation scans. From 25.2.1 onwards, you can specify the SAST sensor version using the command line option `-sastver` or `--sast-version`. For more information, see *Start command* in the *OpenText Fortify ScanCentral SAST Installation, Configuration, and Usage Guide*.

The following table lists the supported SAST sensor versions for remote translation.

ScanCentral client version	Supported SAST (sensor) versions
26.4.0	26.3.x, 26.4.x
26.2.0	26.2.x, 26.3.x
25.4.0	25.4.x, 26.1.x
25.2.1	25.2.x, 25.3.x
25.2.0	25.2.x
24.4.0	24.4.x

4. OpenText Application Security (Software Security Center) server system requirements

This section describes the system requirements for the Application Security server.

This section contains the following topics:

- Hardware requirements
- Supported platforms and architectures
- Supported application server
- Database requirements

- Kubernetes cluster deployment requirements (optional)
- Browsers
- Supported authentication systems
- BIRT report requirements
- Supported service integrations

4.1. Hardware requirements

Application Security requires the hardware specifications listed in the following table.

Server	Component	Minimum required	Minimum recommended
Application server	Java heap size	4 GB	24 GB
Database server	Processor	Eight-core	Eight-core
	RAM	8 GB	64 GB

Database hardware requirements

OpenText recommends an eight-core processor with 64 GB of RAM for the Application Security database. Using less than this recommendation can impact Application Security performance.

Use the following formula to estimate the size (in GB) of the Application Security database disk space:

((<*num_issues*>*30 KB) + <*size_of_artifacts*>) ÷ 1,000,000

where:

- <*num_issues*> represents the total number of issues in the system
- <*size_of_artifacts*> represents the total size in KB of all uploaded artifacts and analysis results



Note

This formula produces only a rough estimate for database disk space allocation. Do not use it to estimate disk space requirements for long-term projects. Disk requirements for Application Security databases increases in proportion to the number of projects, scans, and issues in the system.

Database performance metrics

The following table shows performance metrics (number of issues processed per hour) for Application Security configured with the minimum and the recommended hardware requirements.

Database	**Issues per hour** Minimum configuration	**Issues per hour** Recommended configuration
MySQL	362,514	2,589,385
Oracle® Database	231,392	3,020,950
Microsoft® SQL Server®	725,028	3,625,140

4.2. Supported platforms and architectures

Application Security supports the platforms and architectures listed in the following table.

Operating system	Versions
Microsoft Windows®	Server 2022 Server 2025
Linux®	Red Hat® Enterprise Linux® 8, 9 SUSE® Linux® Enterprise Server 15 **Note** Linux® ARM platform supported as technical preview in 26.4.0.

Note

Although Application Security is not tested on all Linux variants, most distributions are not known to have issues.

4.3. Supported application server

Application Security supports Apache® Tomcat™ version 11.0.x for the following Java™ Development Kit (JDK) versions:

- Oracle JDK 21
- Red Hat OpenJDK 21
- SUSE OpenJDK 21
- Zulu OpenJDK 21 from Azul

OpenText only supports the deployment of a single Application Security instance. That instance must not be behind a layer 7 load balancer of your own implementation. However, OpenText does support an Application Security implementation behind a layer 4 load balancer in a deployment to a Kubernetes cluster.

Important

OpenText does not support the installation of any third-party performance monitoring agents on the Tomcat instance that is hosting Application Security.

4.4. Database requirements

Application Security requires case-sensitive database schema collations.

Important

Application Security is a high I/O-intensive application, which affects performance. Make sure that your disk subsystem provides low read/write latency. OpenText recommends that you monitor disk I/O as the database grows.

All required database drivers are included in the WAR file. Application Security supports the databases listed in the following table.

Database	Versions	Collation / character sets
MySQL	8.4 (Community Edition) Amazon RDS for MySQL (8.4) MySQL Community Server 8.4 GCP Cloud 8.4	latin1_general_cs utf8mb3_bin (if available, preferred over utf8_bin) utf8_bin (only if utf8 is a synonym for utf8mb3 character set)
Oracle	19c (19.3) 26ai	AL32UTF8 for all languages WE8MSWIN1252 for US English
SQL Server	2019 2022 Amazon RDS for SQL Server (2019, 2022) Microsoft® Azure® SQL Database (2019, 2022)	SQL_Latin1_General_CP1_CS_AS

Note

- Application Security does not support the following cloud managed database platforms:
- Azure Database for MySQL
- Oracle in the cloud
- SQL Server on Google Cloud Platform™

OpenText does not support the direct conversion from one database server type to another, such as converting from MySQL to Oracle. To do this, you must use the Server API to move data from your current Application Security instance to a new instance that uses the database server type you want to use going forward. Professional Services can assist you with this process.

4.5. Kubernetes cluster deployment requirements (optional)

To deploy Application Security to a Kubernetes cluster, make sure that the following requirements are met.

Kubernetes cluster requirements

The following are the *minimum* requirements for the default installation:

- Kubernetes versions 1.35, 1.36, 1.37
- Kubernetes persistent volumes with optional support for Pod security context fsGroup option
  Using a non-default container user ID requires fsGroup support.
- Kubernetes LoadBalancer Service type (recommended)
- 28 GB of available RAM and 8 CPUs on a single Kubernetes node
- 4 GiB of storage for persistent volume

Locally-installed tools required

- A kubectl command-line tool
  OpenText recommends that you use the same kubectl command-line tool version as the Kubernetes cluster version or follow the Version Skew Policy on the Kubernetes website.
- Helm command-line tool versions 3.22.1, 4.3.1

To determine which Helm command-line tool version matches your Kubernetes cluster version, see the Helm Version Support Policy on the Helm website.

- (Recommended) A Docker® client and server installation (any version)

Additional requirements

- Kubeconfig file for the Kubernetes cluster
- Docker Hub account with access to Application Security images

> **Note**
> If you need access to the Fortify Docker repository, contact mfi-fortifydocker@opentext.com with your first name, your last name, and your Docker ID. OpenText will then give you access to the Docker organization that contains the Application Security images.

- DNS name for the Application Security web application (address used to access the service)
- Java keystore for setting up HTTPS
  The keystore must contain a CA certificate and a server certificate for the Application Security DNS name with an associated private key.
  - Keystore password
  - Private key password
- Fortify license file

4.6. Browsers

OpenText recommends that you use one of the browsers listed in the following table and a screen resolution of 1400 x 800.

Supported browsers

Browser	Version
Google Chrome™	152 or later
Microsoft® Edge	152 or later
Mozilla® Firefox®	156 or later
Apple® Safari	26 or later

4.7. Supported authentication systems

Application Security supports the following directory services:

- LDAP: LDAP 3 compatible

> **Important**
> Although Application Security supports the use of multiple LDAP servers, it does not support the use of multiple LDAP servers behind a load balancer unless they are exact copies.

- Windows Active Directory service

Single sign-on (SSO)

Application Security supports the following single sign-on solutions:

- HTTP Headers SSO (Oracle SSO, CA SSO)

- SAML 2.0 SSO
- X.509 SSO

4.8. BIRT report requirements

Application Security custom reports support BIRT Report Designer version 4.16.0.

Installing required fonts (Linux only)

To generate BIRT reports on a Linux system from Application Security, you must install the fontconfig library, DejaVu Sans fonts, and DejaVu Serif fonts on the server. If you need to, you can download these fonts from the DejaVu Fonts website.

Installing required libraries (non-GUI Linux only)

To generate reports on a non-GUI Linux system, you must install the GTK and X Window System (X11) libraries.

4.9. Supported service integrations

Application Security supports the service integrations listed in the following table.

Service	Application	Versions
Bug tracking	OpenText™ ALM Quality Center (Deprecated)	12.50
	Azure DevOps **Note** Only basic user password authentication is supported.	Not applicable
	Azure DevOps Server	2020 2022
	Jira Software Server	9.10
	Jira Software Cloud	Not applicable
Static assessments	OpenText™ ScanCentral SAST	26.2.0
Dynamic assessments	OpenText ScanCentral DAST	26.2.0
	Fortify WebInspect Enterprise	Deprecated
Issue Auditing	OpenText™ Fortify Audit Assistant	Not applicable
	OpenText™ Fortify Audit Assistant on Premises	23.2.0 or later

5. OpenText Application Security Tools system requirements

This section describes the system requirements for OpenText Application Security Tools.

This section describes the system requirements for OpenText Application Security Tools and contains the following topics:

- Hardware requirements
- Platforms and architectures
- Software requirements
- Service integrations for Application Security Tools
- Secure Code Plugins
- Authentication for connecting to Application Security
- BIRT reports

5.1. Hardware requirements

OpenText™ Application Security Tools require a system with at least 8 GB of RAM. In addition, the Application Security Toolsused to perform code analysis have the same hardware requirements as Fortify SAST (see *Hardware requirements* in *OpenText™ Static Application Security Testing User Guide*).

5.2. Platforms and architectures

OpenText Application Security Tools support the platforms and architectures listed in the following table.

Operating system	Platforms	Distributions and versions
Microsoft Windows®	x64	10, 11
Linux®	x64 ARM	Red Hat Enterprise Linux 8, 9 SUSE Linux Enterprise Server 15 **Important** Fortify Audit Workbench, Fortify Custom Rules Editor, and Fortify Scan Wizard require GTK version 3.22 or later. Some platform versions include this requirement such as Red Hat Enterprise Linux 7.4 and later.
macOS®	ARM	13, 14, 15

5.3. Software requirements

The Application Security Tools installation includes an embedded OpenJDK/JRE version 25, which the applications and tools require.

To use the Application Security Tools, you must have read and write permissions for the Application Security Tools installation directory.

To run Fortify Audit Workbench, Fortify Custom Rules Editor, or Fortify Scan Wizard remotely from a local server, you must use a remote desktop connection such as Virtual Network Computing (VNC) or Windows Remote Desktop Connection. Do not use X Window System (X11) forwarding to access these applications from a remote server.

5.4. Service integrations for Application Security Tools

The following table lists the supported service integrations for Fortify Audit Workbench and the Secure Code Plugins.

Service	Versions	Supported applications
OpenText Application Quality Management	12.50	Fortify Audit Workbench Fortify Plugin for Eclipse
Azure DevOps Server	2019 2020 2022	Fortify Audit Workbench Fortify Plugin for Eclipse Fortify Extension for Visual Studio
Azure DevOps **Note** Only basic user password authentication is supported.	Not applicable	Fortify Audit Workbench Fortify Plugin for Eclipse
Jira Software Server	8.13 9.10	Fortify Audit Workbench Fortify Plugin for Eclipse
Jira Software Cloud	Not applicable	Fortify Audit Workbench Fortify Plugin for Eclipse
Fortify Software Security Center Bug Tracker	26.4.0	Fortify Audit Workbench Fortify Plugin for Eclipse Fortify Extension for Visual Studio

5.5. Secure Code Plugins

The following table lists the supported integrated development environments (IDE) for the Secure Code Plugins.

Secure Code Plugin	IDE	Versions	Notes
Fortify Plugin for Eclipse	Eclipse	2025-x 2026-x	
Fortify Analysis Plugin for IntelliJ IDEA and Android Studio	IntelliJ IDEA	2025.x 2026.x	IntelliJ IDEA Ultimate and Community Edition are supported.
	Android Studio	2025.x 2026.x	
Fortify Extension for Visual Studio	Visual Studio	2022 2026	Visual Studio Community, Professional, and Enterprise editions for Windows are supported. For supported MSBuild versions, see *OpenText™ Static Application Security Testing User Guide*.

5.6. Authentication for connecting to Application Security

In addition to user name and password authentication, Fortify Audit Workbench and all the Secure Code Plugins can use token-based and SSO authentication with Fortify Software Security Center.

The following table lists the SSO methods that are supported for Fortify SAST applications to connect to Fortify Software Security Center.

Application	SSO method
Fortify Audit Workbench	X.509
Fortify Plugin for Eclipse	X.509
Fortify Extension for Visual Studio	X.509

5.7. BIRT reports

To generate BIRT reports on a Linux system from the Secure Code Plugins or the BIRTReportGenerator utility, you must install the fontconfig library, DejaVu Sans fonts, and DejaVu Serif fonts on the server.

To run the BIRTReportGenerator utility in a Linux Docker container, you must have the X Window System (X11) libraries installed in the image. The X11 libraries provide the graphical user interface API that BIRT requires for data visualization.

Example for Red Hat Enterprise Linux and CentOS:



Example

```
yum -y install xorg-x11-xauth xorg-x11-fonts-* xorg-x11-utils
```

Example for Ubuntu:

Example

```
apt-get install x11-apps
```

6. Fortify Project Results (FPR) file compatibility

OpenText Application Security Software products support opening and uploading FPR files in adjacent releases. OpenText Application Security Software products can open and accept for upload:

- FPR files that have the same version (<*year*>.<*quarter*> portion of the version)
- Older FPR files (within the Product Support Lifecycle policy)
  For example, Fortify Audit Workbench version 26.4.0 can open version 24.2.0 FPR files.
- FPR files that are one version later
  For example, you can upload version 26.4.0 FPR files to Application Security version 25.4.0. Fortify Audit Workbench version 25.4.0 can open version 26.4.0 FPR files.
  OpenText Application Security Software products do not support opening and uploading FPR files generated by later versions of OpenText Application Security Software products when the versions are more than one version apart. For example, uploading a version 26.4.0 FPR to Application Security version 25.2.0 and opening a version 25.2.0 FPR file in the 26.4.0 Fortify Audit Workbench is not supported.

OpenText recommends that you keep your OpenText Application Security Software product versions synchronized so that you are working with FPR file versions that have the same version as your products.

The FPR file version is determined as follows:

- The FPR version is the same version of the analyzer that generated it. For example, an FPR generated by Fortify SAST version 26.4.0 also has version 26.4.0.
- The FPR version is the same version of Application Security or OpenText Application Security Tools that changed or audited the FPR.
- If you merge two FPRs, the resulting FPR has the version of the more recently generated FPR. For example, if you merge a version 24.2.0 FPR with a version 26.4.0 FPR, the resulting FPR has the version 26.4.0.

Caution regarding uploading FPR files to Application Security

Application Security server keeps an FPR file that contains the latest scan results and audit information for each application. Fortify Audit Workbench and the Secure Code Plugins also use this FPR file for collaborative auditing.

Each time you upload an FPR to Application Security, it is merged with the existing FPR. If the FPR has a later version number than the existing FPR, the existing FPR version changes to match the newest FPR.

7. Virtual Machine support

You can run OpenText Application Security Software products on an approved operating system in virtual machine environments. You must provide dedicated CPU and memory resources that meet the minimum hardware requirements. If you find issues that cannot be reproduced on the native environments with the recommended processing, memory, and disk resources, you must work with the provider of the virtual environment to resolve them.

Note

If you run OpenText Application Security Software products in a VM environment, OpenText strongly recommends that you have CPU and memory resources fully committed to the VM to avoid performance degradation.

8. Technologies no longer supported in this release

The following technologies are no longer supported in Application Security Software:

- Service Integrations: Bug Tracking:
  - Azure DevOps Server 2019
- Build Tools:
  - Maven 3.5.x - 3.6.2 (ScanCentral SAST only)
  - xcodebuild 14.3, 14.3.1
- Compilers:
  - swiftc 5.8, 5.8.1
  - Clang 14.0.3
- Databases (Application Security only)
  - SQL Server 2017
- Kubernetes Cluster Deployment (Application Security only):
  - Kubernetes 1.32, 1.33
  - Helm 3.18
- Single Sign-On:
  - SPNEGO/Kerberos SSO
  - Central Authentication Service (CAS) SSO

9. Technologies to lose support in the next release

The technologies listed in this topic are scheduled for deprecation in the next Application Security Software release.



Note

A deprecated technology is no longer recommended for use. Typically, the deprecated item will be removed from the product in a future release. When a technology is deprecated, OpenText recommends that you remove it from your workflow at your earliest convenience.

- Fortify SAST support for all Swift, Xcode, and Objective-C/C++ versions follows the deprecation path Apple Inc. adopts.
- Platforms and architectures (OpenText Application Security Tools):
  - Red Hat Enterprise Linux 7.x

10. Acquiring OpenText Application Security Software

OpenText Application Security Software is available as an electronic download. For instructions on how to download the software from the Software Licenses and Downloads (SLD) portal, click **Contact Us / Self Help** to review the videos and the *Quick Start Guide*.

The following table lists the available packages and describes their contents.

File name	Description

OpenText_Application_Security_Tools_Windows_<version>.zip	OpenText Application Security Tools package for Windows
	This package includes:
	• OpenText Application Security Tools installer, which includes the following components:
	  ○ Fortify Audit Workbench
	  ○ Fortify Custom Rules Editor
	  ○ Fortify Plugin for Eclipse (Eclipse Complete Plugin)
	  ○ Fortify Analysis Plugin for IntelliJ IDEA and Android Studio
	  ○ Fortify Extension for Visual Studio
	  ○ Fortify Scan Wizard
	  ○ ScanCentral SAST client
	• About OpenText Application Security Software Documentation

OpenText_Application_Security_Tools_Windows_<version> Fortify_Tools_<*version*> _Windows.zip.sig	Signature file for the OpenText Application Security Tools Windows package

OpenText_Application_Security_Tools_Linux_<version> Fortify_Tools_<*version*> _Linux.tar.gz	OpenText Application Security Tools package for Linux
	This package includes:
	• OpenText Application Security Tools installer, which includes the following components:
	  ○ Fortify Audit Workbench
	  ○ Fortify Custom Rules Editor
	  ○ Fortify Plugin for Eclipse (Eclipse Complete Plugin)
	  ○ Fortify Analysis Plugin for IntelliJ IDEA and Android Studio
	  ○ Fortify Scan Wizard
	  ○ ScanCentral SAST client
	• About OpenText Application Security Software Documentation

OpenText_Application_Security_Tools_Linux_<version> Fortify_Tools_<*version*> _Linux.tar.gz.sig	Signature file for the OpenText Application Security Tools Linux package

File name	Description

OpenText_Application_Security_Tools_Mac_<version> Fortify_Tools_<*version*> _Mac.tar.gz	OpenText Application Security Tools package for macOS
	This package includes:
	• OpenText Application Security Tools installer, which includes the following components:
	  ○ Fortify Audit Workbench
	  ○ Fortify Custom Rules Editor
	  ○ Fortify Plugin for Eclipse (Eclipse Complete Plugin)
	  ○ Fortify Analysis Plugin for IntelliJ IDEA and Android Studio
	  ○ Fortify Scan Wizard
	  ○ ScanCentral SAST client
	• About OpenText Application Security Software Documentation

OpenText_Application_Security_Tools_Mac_<version> Fortify_Tools_<*version*> _Mac.tar.gz.sig	Signature file for the OpenText Application Security Tools macOS package
Fortify_SSC_Server_<*version*>.zip	Application Security package This package includes: • Application Security WAR file • Application Security seed bundles • About OpenText Application Security Software Documentation
Fortify_SSC_Server_<*version*>.zip.sig	Signature file for the Application Security package
Fortify_ScanCentral_Controller_<*version*>.zip	ScanCentral SAST Controller package This package includes: • ScanCentral SAST Controller • ScanCentral SAST client • About OpenText Application Security Software Documentation
Fortify_ScanCentral_Controller_<*version*>.zip.sig	Signature file for the ScanCentral SAST Controller package

File name	Description

ScanCentral_DAST_<*version*>.zip	OpenText ScanCentral DAST package
	This package includes:
	• DAST.ConfigurationToolCLI.exe
	• SampleSettingsFile.json
	• SampleSettingsFile.yml
	• Linux and Windows versions of the following files to support mTLS authentication:
	  ○ docker-compose-mtls.yml
	  ○ docker-compose-mtls-scannerservice.yml
	  ○ environment-mtls.env
	  ○ environment-scannerservice-mtls.env
	• Linux and Windows versions of the following files to support TLS authentication:
	  ○ docker-compose-tls.yml
	  ○ docker-compose-tls-scannerservice.yml
	  ○ environment-tls.env
	  ○ environment-scannerservice-tls.env
	• ScanCentral DAST - Sensor Service.zip (sensor service and supporting bits)
	• appsettings.json (configures the sensor service)
	• Dynamic_Addons.zip (installers for optional FAST and Scan Scaling components)
	• About OpenText Application Security Software Documentation

ScanCentral_DAST_<*version*>.zip.sig	Signature file for the OpenText ScanCentral DAST package

10.1. Verifying software downloads

This topic describes how to verify the digital signature of the signed file that you downloaded from the Customer Support site. Verification ensures that the downloaded package has not been altered since it was signed and posted to the site. Before proceeding with verification, download the OpenText Application Security Software product files and their associated signature (`*.sig`) files. You are not required to verify the package to use the software, but your organization might require it for security reasons.

Preparing your system for digital signature verification

To prepare your system for electronic media verification:

1. Navigate to the GnuPG site (http://www.gnupg.org).
2. Download and install GnuPG Privacy Guard.
3. Generate a private key, as follows:
    1. Run the following command (on a Windows system, run the command without the `$` prompt):
    `$ gpg --gen-key`
    2. When prompted for key type, select `DSA and Elgamal`.
    3. When prompted for a key size, select `2048`.
    4. When prompted for the length of time the key should be valid, select `key does not expire`.
    5. Answer the user identification questions and provide a passphrase to protect your private key.

4. Download the OpenText GPG public keys (compressed tar file) from https://mysupport.microfocus.com/documents/10180/0/MF_public_keys.tar.gz.
   <<At the time of publishing 24.4.0, this public keys link gives a security error page because the certificate expired 2/6/24. Perhaps this should be changed to https://portal.microfocus.com/s/article/KM000003948?language=en_US, which includes a description of this process.>>
5. Extract the public keys.
6. Import each downloaded key with GnuPG with the following command:
   `gpg --import <path_to_key>/<key_file>`

1. **Example**

   ```
   gpg --verify Fortify_SSC_Server_26.4.0.zip.sig Fortify_SSC_Server_26.4.0.zip
   ```

2. **Example**

   ```
   gpg: Signature made Wed, November 10, 2022 12:05:20 AM PDT using RSA key ID AB42A5CF
   gpg: Good signature from "Micro Focus Group Limited RS A2048 1"
   ```

Note

11. Assistive technologies (Section 508)

In accordance with section 508 of the Rehabilitation Act, Fortify Audit Workbench is engineered to work with the JAWS screen reading software package from Freedom Scientific. JAWS provides text-to-speech support for use by the visually impaired. With JAWS, labels, text boxes, and other textual components can be read aloud, providing greater access to these technologies.

Application Security works well with the ChromeVox screen reader.

© Copyright 2026 Open Text

For more info, visit https://docs.microfocus.com